



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

FACULTAD DE INGENIERÍA

**Sistema de efectos digitales de audio  
controlado por un Smartphone en tiempo  
real**

**TESIS**

Que para obtener el título de  
**Ingeniero Eléctrico Electrónico**

**PRESENTAN**

Francisco Javier López Delhumeau  
Rodrigo Romero Vergara

**DIRECTOR DE TESIS**

M. I. Larry Hipólito Escobar Salguero



Ciudad Universitaria, Cd. Mx., 2022



---

Si el hombre es formado por las circunstancias, entonces es necesario formar las circunstancias humanamente.

---

K. Marx y F. Engels  
*La sagrada Familia.*

La poesía no está solamente en los textos, está en aquellos que la escriben, resplandece a través de su simple presencia aquí, cerca de nosotros.

---

Yves Bonnefoy  
*La luz impronunciable.*



# Agradecimientos

## **Rodrigo Romero Vergara**

Quiero agradecer profundamente la oportunidad que me ha dado la vida de poder ejercer una profesión tan maravillosa como es la ingeniería.

A mi tía Glorinela, mi tío Rafa, a mi abuelita Raquel y mi abuelita Tere, y a toda mi familia que forjó mi carácter y me permitió crecer.

Gracias a los profesores que tuve la fortuna de conocer en la facultad, que compartieron no solo sus conocimientos sino sus experiencias y valores a lo largo de mis años en la carrera. Al maestro Larry, director de ésta Tesis, que sin su guía y paciencia no hubiera sido posible realizar.

A mis amistades que hice durante la carrera, Frank, Manuel, Jordy, Emilio, Axel, Beca, Áaron, con los que compartí muchas memorias y recibí su apoyo innumerables veces . A Rubén, Eli, Cinthya y Laura quienes han creído en mí por muchos años, que me han guiado en los momentos más oscuros y, finalmente, quiero agradecer especialmente a mi compañero de Tesis, Paco, que no solo me ayudo en lo académico, sino en lo personal a ser la persona que soy.

Éste trabajo va con especial agradecimiento a mis padres, Daniel y Angélica, que se esforzaron día a día por verme crecer, que me inculcaron los valores necesarios para ser una persona de bien y el amor que todo hijo necesita. Gracias papá, que en paz descanse, y gracias mamá.

## **Francisco Javier López Delhumeau**

Primeramente agradezco a mis padres, Ale y Frank, porque sin su apoyo incondicional, sin sus valores y sin su amor esto no sería posible. Quiero que sepan que esto también es un logro suyo. A mi hermano Alejandro, por ser un gran ejemplo y por darme calma cuando las cosas pintaban difíciles. A Isaac, Regina y Santi, por ser mis pequeños motores.

A mis abuelas Cris y Rufi, por hacerme de buena madera, y a mi abuelo Lalo porque con tus enseñanzas y cariño, pude cumplir lo que te prometí. A toda mi familia que es el pilar de lo que soy.

A nuestro tutor y director de tesis Larry, por sus consejos, su disponibilidad y su infinita paciencia. A nuestros sinodales y profesores, que nos compartieron su tiempo y experiencia para desarrollarnos tanto académica como personalmente, y por darme esta motivación de siempre querer aprender más.

---

A Roy, por su apoyo como compañero de tesis y como amigo, por su nobleza y sinceridad en todo momento. A Sofi por todo el amor, la compañía y la sabiduría que me permitió compartir con ella durante este trayecto. A mis amigos de la facultad, Frank, Jordy, Manuel, Fany y Emilio, por las risas y el aprendizaje que compartimos durante estos años de estudiantes.

A Camus y Dante que me acompañaron durante las noches de desvelo. A mis amigos Ariel, Pablo, Isaí, Kat, Saúl y en especial a Fer. Gracias por escucharme, leerme y corregirme cuando fue necesario. A todos aquellos que me apoyaron en este viaje en el que por momentos dudé pero cada uno fue una razón para decidir continuar, les agradezco de corazón.

# Resumen

En esta tesis se presenta una breve investigación acerca de los efectos especiales más utilizados en la industria de la música, cine, videojuegos y realidad virtual, para posteriormente ser diseñados, simulados e implementados en el DSP TM320F28377S de Texas Instruments dónde los algoritmos de cada efecto y la configuración de los periféricos fueron programados en una combinación de lenguaje ensamblador y lenguaje C.

A su vez, se realizó y documentó el análisis, diseño e implementación de los módulos analógicos que nos permiten obtener y proporcionar señales de audio por medio de una entrada y salida de tipo estéreo, más la incorporación de un módulo bluetooth HC05 que se comunica con una aplicación instalada en un smartphone. La aplicación en el smartphone ayuda al usuario a elegir el efecto de audio y dar información del paneo de la señal que se procesa en el DSP en tiempo real. Fue diseñada y creada por medio de la plataforma App Inventor del MIT cuya licencia es de código abierto.

En la última sección de este trabajo, se presentan los resultados obtenidos y el análisis de los mismos, haciendo una breve comparación con los resultados simulados y comprobando el cumplimiento de las características teóricas necesarias para cada efecto y el funcionamiento de todo el sistema en tiempo real. Asimismo, se anexan las ligas correspondientes para descargar el código implementado en esta tesis.





# Abstract

In this Thesis we set forth brief research of the most used special effects in music, cinema, video games, and the virtual reality industry, in order to then be designed, simulated, and implemented on the Texas Instruments DSP TM320F28377S. The algorithms of some effects and the configuration of the peripheral were programmed using a combination of ASM and C languages.

Furthermore, we composed and documented the analysis, design, and implementation for each analogic module, allowing us to obtain and deliver the audio signal using an Estero input and output. Also, we added the HC05 Bluetooth module, which will enable us to communicate with our App previously downloaded on a Smartphone. This App was designed and developed using the open-source “MIT App Inventor” platform.

In the last section of this document, we present the analysis of the resulting data, adding a brief comparison with the simulation outcomes, demonstrating the fulfillment of reaching the theoretic properties needed for each of the effects and the accomplishment of the system running in real-time. Additionally, we attach the links for downloading the source code used in this investigation.



# Índice general

Agradecimientos	III
Resumen	V
Abstract	VII
Índice de Figuras	XIII
Índice de Tablas	XVII
<b>1. Introducción</b>	<b>1</b>
1.1. Objetivo . . . . .	2
1.2. Definición del problema . . . . .	2
1.3. Metodología . . . . .	3
1.4. Descripción de Contenido . . . . .	4
<b>2. Teoría del procesamiento de audio y efectos digitales</b>	<b>7</b>
2.1. Definición y clasificación de los efectos de audio . . . . .	8
2.1.1. Efectos por retardo . . . . .	11
2.1.2. Efecto de reverberación (efecto espacial) . . . . .	14
2.1.3. Efectos por distorsión (efectos no lineales) . . . . .	17
2.1.4. Efectos por filtrado. . . . .	21
2.1.5. Paneo de sonido estéreo. . . . .	24
2.2. Resumen . . . . .	27
<b>3. Acondicionamiento, sensores e interfaz del sistema</b>	<b>29</b>
3.1. Acondicionamiento de una señal . . . . .	30
3.1.1. Transductores . . . . .	30
3.1.2. Etapa de amplificación . . . . .	30
3.1.3. Etapa de offset . . . . .	35
3.2. Filtros digitales . . . . .	37
3.2.1. Filtros FIR . . . . .	38
3.2.2. Filtros IIR . . . . .	39
3.3. Módulo Bluetooth . . . . .	40
3.3.1. Comandos AT . . . . .	41
3.4. Sensores de teléfonos inteligentes: Acelerómetro, giroscopio y magnetómetro .	44

3.4.1.	Acelerómetro . . . . .	44
3.4.2.	Giroscopio . . . . .	45
3.4.3.	Magnetómetro . . . . .	46
3.5.	Aplicación Móvil . . . . .	47
3.5.1.	Android . . . . .	49
3.5.2.	MIT App Inventor . . . . .	50
3.6.	Resumen . . . . .	53
<b>4.</b>	<b>Periféricos del DSP: ADC, DAC y protocolos de comunicación</b>	<b>55</b>
4.1.	DSP: TM320F28377S . . . . .	56
4.2.	Subsistema analógico . . . . .	57
4.2.1.	Modulación por codificación de pulsos . . . . .	57
4.2.2.	Muestreo y retención (Sample & Hold o S/H) . . . . .	61
4.2.3.	Convertidor Digital-Analógico (DAC) . . . . .	62
4.2.4.	Convertidor Analógico Digital (ADC) . . . . .	64
4.2.5.	Tipos de ADC . . . . .	64
4.2.6.	ADC en el DSP TMS320F28377s . . . . .	68
4.2.7.	DAC en el DSP TMS320F28377s . . . . .	72
4.3.	Comunicación serial . . . . .	73
4.3.1.	Características de una comunicación serial . . . . .	73
4.3.2.	Trama de datos . . . . .	74
4.3.3.	Protocolo de comunicación SCI . . . . .	74
4.3.4.	SCI en el DSP TMS320F28377s . . . . .	76
4.4.	Resumen . . . . .	78
<b>5.</b>	<b>Diseño e implementación del sistema</b>	<b>79</b>
5.1.	Requerimientos generales . . . . .	80
5.2.	Descripción general . . . . .	80
5.3.	Acondicionamiento de señales	
Tarjeta de expansión . . . . .		81
5.4.	Interfaz del sistema y aplicación móvil . . . . .	84
5.4.1.	Pantalla de bienvenida y créditos . . . . .	84
5.4.2.	Pantalla de efectos . . . . .	85
5.5.	Módulo Bluetooth . . . . .	89
5.6.	Programación del DSP . . . . .	90
5.6.1.	Convertidor analógico-digital . . . . .	90
5.6.2.	Convertidor digital-analógico . . . . .	91
5.6.3.	SCI . . . . .	91
5.7.	Efectos de sonido . . . . .	92
5.7.1.	Retardos . . . . .	92
5.7.2.	Reverberación . . . . .	97
5.7.3.	Distorsión . . . . .	101
5.7.4.	Filtros digitales . . . . .	102
5.7.5.	Paneo de amplitud . . . . .	108
5.8.	Resumen . . . . .	111

<b>6. Pruebas y Resultados</b>	<b>113</b>
6.1. Análisis de la señal en el dominio del Tiempo . . . . .	114
6.1.1. Análisis de efectos de coro y eco. . . . .	114
6.1.2. Análisis de efectos de reverberación. . . . .	116
6.2. Análisis de la señal en el dominio de la Frecuencia . . . . .	118
6.2.1. Análisis de efectos de distorsión. . . . .	118
6.2.2. Análisis de banco de Filtros. . . . .	123
6.3. Análisis del paneo. . . . .	128
6.4. Análisis en tiempo real . . . . .	130
6.5. Resumen . . . . .	132
<b>7. Conclusiones</b>	<b>133</b>
7.1. Trabajo a futuro . . . . .	135
<b>Bibliografía</b>	<b>135</b>
<b>Glosario de abreviaturas</b>	<b>140</b>
<b>A. Convención de notación eléctrica</b>	<b>143</b>
<b>B. Propiedades función impulso</b>	<b>145</b>
<b>C. Resultados experimentales de los tiempos de procesamiento de los efectos</b>	<b>147</b>
<b>D. Código principal</b>	<b>149</b>



# Índice de Figuras

1.1. Esquema del sistema completo. . . . .	3
2.1. Diagrama de bloques de efecto coros. . . . .	13
2.2. Diagrama de muestras de reflexiones tempranas, reflexiones tardías y tiempo de reverberación en efecto de reverberación. . . . .	16
2.3. Respuesta experimental del efecto de reverberación que muestra el sonido directo, reverberaciones tempranas, tardías y piso de ruido. . . . .	18
2.4. Representación de SoftClipping (recorte suave) y HardClipping (recorte duro) de señal senoidal aplicando efecto por distorsión. . . . .	20
2.5. Respuesta en frecuencia de una señal periódica antes y después de aplicar efecto por distorsión. . . . .	21
2.6. Configuración de audición estereofónica estándar. . . . .	26
3.1. Representación de un amplificador como bipuerto. . . . .	30
3.2. Modelo simplificado de un amplificador como un bipuerto. . . . .	31
3.3. Representación de un amplificador con fuente y carga. . . . .	32
3.4. Modelos de un amplificador diferencial. . . . .	32
3.5. Función de transferencia ideal de un amplificador operacional. . . . .	33
3.6. Circuito de Offset a la salida de la amplificación. . . . .	36
3.7. Circuito divisor de Voltaje como equivalente de Circuito de Offset en DC. . . . .	37
3.8. Circuito paso Altas como equivalente de Circuito de Offset en AC. . . . .	37
3.9. Gráfica de respuesta a diferentes tipos de filtros Lineales . . . . .	40
3.10. Módulo HC-05. . . . .	42
3.11. Corte lateral del diseño de un acelerómetro con tecnología MEMS. . . . .	44
3.12. Representación del sistema de coordenadas de un acelerómetro en un teléfono inteligente . . . . .	45
3.13. Modelo de un Giroscopio de Ajuste Dinámico, DTG. . . . .	46
3.14. Giroscopio MEMS . . . . .	46
3.15. Representación del efecto Hall . . . . .	47
3.16. Total de usuarios de teléfonos inteligentes, en miles de millones. . . . .	48
3.17. Ejemplo de un SoC de un teléfono inteligente, el Snapdragon 845. . . . .	49
3.18. Logo de Android y su composición. . . . .	50
3.19. MIT App Inventor . . . . .	50
4.1. Vista frontal del DSP TM320F28377S. . . . .	56
4.2. Representación del subsistema analógico de un sistema digital. . . . .	57

4.3. Modulación por codificación de pulsos o PCM. . . . .	57
4.4. Ejemplos del muestreo de una señal senoidal de 20 kHz. . . . .	58
4.5. Espectro en frecuencia de una señal muestreada . . . . .	60
4.6. Ejemplo de los ajustes realizados a la función de transferencia de un ADC. . . . .	61
4.7. Circuito T/H . . . . .	62
4.8. Simulación del circuito T/H. . . . .	62
4.9. Esquema de un DAC de resistencias en escalera. . . . .	63
4.10. Esquema de un DAC con arreglo R-2R. . . . .	64
4.11. Esquema de un comparador y su señal de salida. . . . .	65
4.12. Esquema de un ADC flash . . . . .	65
4.13. Esquema de un ADC de tipo Rampa . . . . .	66
4.14. Esquema de un ADC de aproximaciones sucesivas con su señal de salida . . . . .	67
4.15. Esquema de un ADC $\Sigma\Delta$ . . . . .	67
4.16. Módulo ADC contenido en el DSP TMS320F28377s . . . . .	69
4.17. Módulo TIMER contenido en el DSP TMS320F28377s . . . . .	70
4.18. Módulo DAC contenido en el DSP TMS320F28377s . . . . .	72
4.19. Diagrama de bloques de un protocolo de comunicación SCI. . . . .	75
4.20. Ejemplo gráfico de la trama del protocolo de comunicación SCI. . . . .	75
4.21. Ejemplo gráfico de el sobremuestreo en la transmisión de información, con s=4. . . . .	76
4.22. Módulo SCI en contenido en el DSP. . . . .	77
5.1. Diagrama general del sistema. . . . .	80
5.2. Diagrama de conexiones con la tarjeta. . . . .	81
5.3. Tarjeta de Expansión. . . . .	82
5.4. Circuito eléctrico de la alimentación de la tarjeta de expansión. . . . .	82
5.5. Conexión del LED RGB a los EPWM. . . . .	83
5.6. Circuito de acondicionamiento de la señal de entrada de audio estéreo. . . . .	83
5.7. Circuito de acondicionamiento de la señal de salida de audio estéreo. . . . .	84
5.8. Mensaje de bienvenida en la aplicación. . . . .	85
5.9. Entorno de programación de bloques de la pantalla de Bienvenida. . . . .	85
5.10. Pantalla principal de la aplicación, mostrando los elementos cuando no se ha realizado la conexión con el módulo Bluetooth. . . . .	86
5.11. Esquema de rotación del sensor de orientación del teléfono. . . . .	87
5.12. Pantalla principal, mostrando los elementos cuando se ha realizado la conexión con el módulo. . . . .	88
5.13. Banco de filtros con sus elementos gráficos. . . . .	88
5.14. Diagrama del funcionamiento de buffer de memoria para los efectos de retardo. . . . .	93
5.15. Diagrama de flujo de la rutina principal del efecto de retardos y subrutina de suma de retardo. . . . .	95
5.16. Diagrama de flujo de subrutina del efecto de retardos para recorrer valores en buffer y subrutina de limpieza de Buffer. . . . .	96
5.17. Diagrama del funcionamiento de buffer de memoria para los efectos de reverberación. . . . .	100
5.18. Diagrama de flujo de la rutina principal del efecto de reverberación y subrutina de suma de retardos. . . . .	100
5.19. Diagrama de flujo de las rutinas de los efectos de Distorsión y Saturación. . . . .	102



5.20. Respuesta de función de transferencia de filtro paso bajas tipo Butterworth de distintos tipos de orden $n$ . . . . .	105
5.21. Simulación de banco de las 10 bandas de Filtros Digitales con respecto a un ecualizador de una octava. . . . .	106
5.22. Diagrama de bloques de implementación de filtros tipo IIR en forma directa I.	107
5.23. Diagrama del funcionamiento de buffer de memoria para la implementación de filtros IIR. . . . .	107
5.24. Diagrama de flujo de la rutina principal del efecto de filtrado. . . . .	108
6.1. Simulación en Octave de los efectos eco y coro aplicados a señal de pulsos. . .	114
6.2. Aplicación del efecto de Retardo de la Señal . . . . .	115
6.3. Simulación en Octave de los efectos de reverberación aplicados a señal de pulso.	116
6.4. Aplicación del efecto de Reverberación. . . . .	117
6.5. Simulación en Octave de los efectos de distorsión aplicados a señal senoidal de 1[KHz]. . . . .	118
6.6. Simulación en Octave de los efectos de distorsión aplicados a señal senoidal de 20Hz en el dominio de la Frecuencia. . . . .	119
6.7. Aplicación de los efectos de distorsión. . . . .	120
6.8. Espectro de la señal de salida para los efectos de Distorsión. . . . .	121
6.9. Simulación del resultado de la suma de los filtros IIR del efecto del banco filtros con ganancia unitaria de 0 a 25[kHz] en frecuencia. . . . .	124
6.10. Simulación del resultado de la suma de los filtros IIR del efecto del banco filtros con ganancia unitaria ampliado de 0 a 1[KHz] en frecuencia. . . . .	124
6.11. Simulación del resultado de la suma de los filtros IIR con atenuación de 12dB en filtros pares. . . . .	125
6.12. Simulación del resultado de la suma de los filtros IIR con atenuación de 12dB en filtros impares. . . . .	125
6.13. Espectro de la señal de salida, comparación entre ningún efecto y aplicación de filtros. . . . .	126
6.14. Espectro de la señal de salida donde se suprimen las frecuencias de los filtros impares. . . . .	126
6.15. Espectro de la señal de salida donde se suprimen las frecuencias de los filtros pares. . . . .	127
6.16. Resultados experimentales de la aplicación del paneo. . . . .	129
6.17. Tiempo que el DSP tarda en procesar los efectos de Reverberación a dos segundos y Banco de filtros. . . . .	131
C.1. Tiempo de procesamiento del sistema cuando se aplica el banco de filtros. . .	147
C.2. Tiempo de procesamiento del sistema cuando se aplica el efecto de distorsión.	148
C.3. Tiempo de procesamiento del sistema cuando se aplica el efecto de Reverberación. . . . .	148
C.4. Tiempo de procesamiento del sistema cuando se aplica el efecto de Retardo de la señal. . . . .	148



# Índice de Tablas

2.1.	Clasificación de los efectos de audio de acuerdo a sus atributos perceptibles.	11
2.2.	Efectos de audio típicos basados en retardos. . . . .	12
2.3.	Ejemplos de tiempos de reverberación de salas de conciertos famosas dependientes de la frecuencia. . . . .	15
2.4.	Intervalos estándar para ecualizadores de una octava y un tercio de octava. .	24
3.1.	Configuraciones comunes con un amplificador operacional. . . . .	35
3.2.	Características del módulo Bluetooth HC05 . . . . .	41
3.3.	Ejemplos de los comandos AT más comunes con sus posibles respuestas . . .	43
5.1.	Asignación de colores para las instrucciones del SCI . . . . .	83
5.2.	Tabla de efectos y sus variaciones . . . . .	87
5.3.	Codificación de las instrucciones y valores . . . . .	89
5.4.	Parámetros específicos del módulo HC05 . . . . .	90
5.5.	Valores de la función exponencial correspondientes a cada retardo de los efectos de reverberación de uno y dos segundos. . . . .	99
5.6.	Frecuencias utilizadas en el diseño del banco de filtros de 10 bandas a partir del diseño de un ecualizador de una octava. . . . .	103
6.1.	Distorsión armónica de las simulaciones de los efectos de distorsión calculadas en Octave. . . . .	119
6.2.	Distorsión armónica del efecto de saturación en una señal senoidal con $f_0 = 1$ [kHz].	121
6.3.	Distorsión armónica del efecto de distorsión en una señal senoidal con $f_0 = 1$ [kHz].	122
6.4.	Señales senoidales que conforman la suma de la señal resultante utilizada en las pruebas del banco de Filtros. . . . .	123
6.5.	Valores obtenidos experimentalmente de la aplicación del paneo. . . . .	128
6.6.	Tiempo de la aplicación de los efectos del sistema. . . . .	130



# Capítulo 1

## Introducción

El procesamiento de señales es de suma importancia en el mundo actual. La mayoría de las ocasiones, los seres humanos, no conocemos ni la cantidad de algoritmos ni la tecnología que los sistemas digitales emplean comúnmente para poder realizar una tarea que consideramos sencilla, como una llamada por teléfono o el sistema de ecualización, que podemos encontrar en los reproductores de audio de distintos dispositivos.

Esta tesis está dirigida al procesamiento digital de audio, cuya aplicación la podemos encontrar en varios sistemas que aprovechan las capacidades del oído para crear una sensación inmersiva; estos sistemas pueden ser, por ejemplo, videojuegos o teatro en casa. Para conseguir todo esto, se utilizan los conocimientos de la ingeniería para lograr por medio del hardware, la obtención y entrega de las señales de entrada y salida de audio respectivamente, de tipo estéreo en posiciones específicas (izquierda-derecha). Por medio del software, se consigue el tratamiento de la señal de audio generando y aplicando en ella los efectos que alteren las diferentes características de dicha señal en tiempo real. Aunado a esto, en el presente trabajo se busca dar al usuario del sistema la capacidad de modificar a su conveniencia los parámetros de la señal en tiempo real por medio de la comunicación con un teléfono inteligente.

## 1.1. Objetivo

Analizar, diseñar y realizar un sistema embebido con salida binaural en un Procesador Digital de Señales (DSP) capaz de modificar las características de amplitud y frecuencia de una señal de audio estéreo de entrada, a partir de una selección de efectos de sonido predefinidos y realizada por el usuario a través de un Smartphone y por el movimiento de este dispositivo.

## 1.2. Definición del problema

En el mundo en el que vivimos, gran parte de las magnitudes físicas que analizamos y percibimos son de carácter analógico, sin embargo, el manejo y procesamiento de estas señales en los sistemas electrónicos y computacionales son de tipo digital, por lo que es válido realizar la pregunta ¿por qué es tan importante el mundo digital?, la manera más sencilla de responder esto es presentando las principales ventajas y desventajas que poseen los sistemas digitales. Una de las ventajas sobresalientes de la digitalización es la capacidad de almacenamiento de la información y su flexibilidad de proceso. Por un lado, el almacenar información nos permite tener acceso a ella en cualquier momento posterior a su obtención, ya sea para procesarla o transferirla a otro dispositivo de la manera más íntegra posible sin agregar ruido indeseado a la señal durante la transmisión a diferencia de los sistemas analógicos. La flexibilidad se refiere al cambio de la estructura principal de la señal para que el sistema tenga una salida diferente a la entrada y como consecuencia pueda ser usado en otras aplicaciones. Por otro lado, una de las desventajas de los circuitos digitales, es que resultan más caros de producir, por lo que regularmente sólo son destinados para proyectos que requieren un mayor grado de complejidad. A su vez, se debe tomar en cuenta que los sistemas analógicos son más rápidos que los digitales, por el simple hecho de que no es necesario convertir la señal entre dominios.

A lo largo del crecimiento de la tecnología, los sistemas digitales han evolucionado constantemente para darle una mejor experiencia al usuario, en especial los sistemas de audio digital que han mejorado tanto en calidad, como en la inclusión de herramientas que le permiten al escucha una mayor experimentación con diversos efectos de sonido. Con el auge de los sistemas de realidad virtual, han surgido diversos dispositivos que se enfocan en sumergir al usuario en el realismo de una producción tanto visual como sonora, sin embargo, en la actualidad la mayoría de estos instrumentos sólo utilizan elementos multimedia que son creados y grabados específicamente para este tipo de mecanismos.

En otro contexto, la idea de crear una experiencia de audio más envolvente ha dado lugar desde hace varias décadas a experimentos musicales muy interesantes que aprovecharon el sonido estéreo en los auriculares y los efectos de audio naturales (basados principalmente en las diferentes características de distintos espacios donde se pueden percibir los sonidos) para crear una ambientación casi real a donde el escucha pudiese percibir sonidos como si se encontrara, por ejemplo, en medio de una presentación musical.

La idea de este trabajo se basa en la mezcla de estos dos conceptos de percepción del audio, aunado a un control de efectos especiales predeterminados, tales como los que se usan en la producción y edición de audio en el cine, álbumes musicales, videojuegos, etc. De tal

manera que el usuario puede cambiar estos efectos en tiempo real desde la conexión con un dispositivo inteligente, en este caso un Smartphone -considerando que es un dispositivo de fácil acceso a la gran mayoría de la población actual-.

### 1.3. Metodología

El sistema embebido se ha realizado con base en el procesamiento de audio en un DSP modelo TMS320F28377s acompañado de los respectivos periféricos y sistemas de acondicionamiento para la detección de datos y la entrada, salida de audio binaural [1] respectivamente, como se puede observar en la Figura 1.

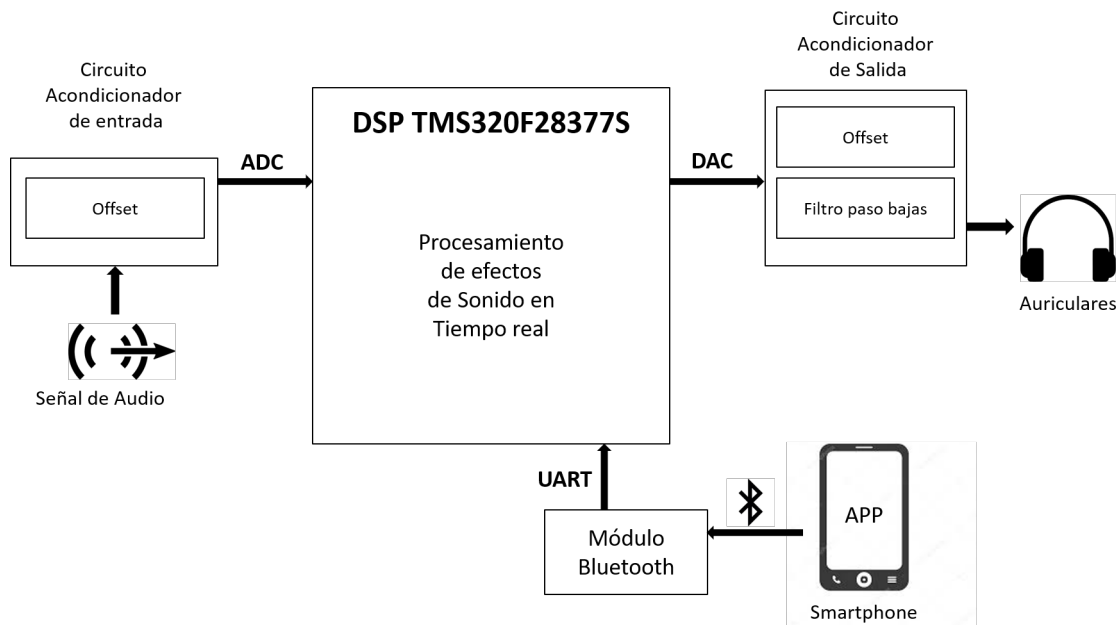


Figura 1.1: Esquema del sistema completo.

Para el concepto de los efectos de sonido se han seleccionado los siguientes efectos especiales a partir de la clasificación ahondada en el capítulo 2 [2]:

- Efecto por saturación de la señal:
  - Saturación
  - Distorsión
- Efecto por repetición de la señal:
  - Reverberación de un segundo
  - Reverberación de dos segundos
- Efecto por retardo de la señal:
  - Coros
  - Eco

- Efecto por filtrado de la señal:
  - Banco de filtros de diez bandas

Todo el procesamiento se realiza en el DSP.

Para la selección de efectos y filtros que se aplican a la señal de entrada, se ha desarrollado una aplicación móvil para smartphone (APP) capaz de conectarse al DSP. Existe una interfaz en línea [3] de código libre desarrollado por el MIT (Massachusetts Institute of Technology), que permite la creación de aplicaciones a la medida para celulares con sistema operativo Android; la interfaz consiste en colocar bloques para procesamiento de la información, la cual puede estar contenida en el mismo código o puede ser proveniente de los periféricos del smartphone. Particularmente, centramos nuestra atención a la comunicación bluetooth, pues existen periféricos que se pueden conectar al DSP por medio de comunicación serial (en el caso más común SCI) para intercambiar información con el smartphone y de esta manera, otorgar al usuario la libertad de elegir el resultado a la salida del sistema.

Para la mezcla del concepto de realidad virtual y audio envolvente, se aprovecha el canal doble del sonido estéreo para poder controlar la amplitud de la señal de audio que percibiría el usuario del sistema en el auricular izquierdo y/o derecho, de cualquier grabación disponible en formatos de audio estándar, basado en la detección del movimiento del smartphone en dos ejes (tomando como referencia el movimiento de la cabeza, serían movimientos derecha-izquierda y arriba-abajo) utilizando el giroscopio del mismo dispositivo.

Por último, la entrada y la salida de audio están dadas por conectores plug 3.5mm hembra, y su respectivo circuito de acondicionamiento de audio, donde la señal acondicionada entra al DSP por medio del Convertidor Analógico Digital (ADC) y la salida antes de ser acondicionada sale del DSP por medio del Convertidor Digital Analógico (DAC).

## 1.4. Descripción de Contenido

En el Capítulo 2: *Teoría del procesamiento de audio y efectos digitales*, se introducen los temas generales de esta tesis tales como las señales de audio, las señales discretas, procesamiento de audio y los efectos digitales de audio. Se recapitulan brevemente sus conceptos básicos, su historia y su clasificación. Posteriormente, a cada uno de los efectos utilizados en la presente tesis se les da una descripción teórica y matemática y se presentan algunos ejemplos de su implementación en sistemas reales de la industria audio-visual.

En el Capítulo 3: *Acondicionamiento, sensores e interfaz del sistema*, se describe el procesamiento que se le da a la señal de audio empleando electrónica analógica antes y después de ser digitalizada, además que se describe el funcionamiento de los sensores utilizados en el sistema y como a partir de éstos se modifica la señal de salida.

En el Capítulo 4: *Periféricos del DSP: ADC, DAC y protocolos de comunicación*, se describe el funcionamiento de los principales módulos del DSP: el subsistema analógico, que nos permite obtener y digitalizar una señal inicial analógica para ser procesada en el DSP



y posteriormente transformar la nueva señal procesada al dominio continuo y el módulo SCI que nos permite intercambiar información con otros sistemas digitales, en este caso, un smartphone.

En el Capítulo 5: *Diseño e implementación del sistema*, se utiliza la información de los capítulos anteriores para describir el procedimiento de diseño e implementación del sistema a partir del diagrama completo y de cada una de las partes que lo conforman, describiendo a su vez, cada una de estas partes de manera más extensa a lo largo del capítulo.

En el Capítulo 6: *Pruebas y Resultados*, se presentan de manera gráfica el montado del sistema. A su vez, se detallan los resultados obtenidos para cada efecto seleccionado y se hace una comparación con los resultados esperados, simulados previamente en otra plataforma.

En el Capítulo 7: *Conclusiones*, se presentan las conclusiones con base en los resultados obtenidos y se hace mención de la utilidad de este proyecto y cómo se puede incluir en trabajos posteriores cuyo objetivo sea de la misma índole.



## Capítulo 2

# Teoría del procesamiento de audio y efectos digitales

Los efectos de audio son herramientas que se utilizan en transmisiones, televisión, películas, juegos y sobre todo en producciones musicales. Aunque inicialmente su función principal era mejorar las grabaciones y realizar correcciones del audio en el proceso de producción, ahora se le suman un gran número de utilidades creativas, que no sólo se pueden aplicar a archivos de audio pre grabados sino también a señales sonoras en tiempo real.

Para poder diseñar e implementar un sistema con algunos de estos efectos de audio, tenemos que iniciar comprendiendo los conceptos de las señales de audio y sus propiedades fundamentales. En este capítulo nos centramos en las señales sonoras, sus principales características y su representación como señales digitales, así como los sistemas que conforman los efectos y a su vez, modifican dichas señales sonoras. Para el análisis tanto de las señales de audio y de los efectos digitales -que se ocuparon a lo largo de esta tesis-, se presenta un breve repaso a los conceptos, se mencionan sus propiedades más importantes y se les asigna una representación matemática y/o gráfica. Por último, en el caso de los efectos digitales, se relata de manera breve algunos de sus usos y aplicaciones en la industria cinematográfica y musical.

## 2.1. Definición y clasificación de los efectos de audio

Todo sonido es representado en el mundo real como un conjunto de formas de onda que viajan a través de un medio, en donde, generalmente para el sonido que nosotros percibimos, el medio en el que se desplazan estas ondas es el aire. Estas formas de onda se crean cuando la vibración de objetos ejerce presión en el aire y estas se desplazan a través de dicho medio por un lapso de tiempo determinado hasta llegar a nuestros oídos, es ahí cuando podemos percibir aquel tramo de sonido y por lo tanto, podemos decir que las ondas de sonido son continuas a través del tiempo.

Fue hasta el siglo XX que se mantuvo este único concepto de señal analógica continua en el tiempo para representar el sonido, sin embargo, con la llegada de la digitalización, también nació el audio digital: una manera de representar las ondas de sonido de manera discreta. Esto permitió no sólo guardar las muestras de una señal sonora, sino también procesarlas antes de reconstruirlas a señales analógicas que los humanos podemos escuchar.

El proceso de digitalización del audio también nos trajo algunos conceptos que son fundamentales de conocer al trabajar con este tipo de señales, tales como [4]:

- Frecuencia de muestreo: es la tasa a la que se toman las muestras de una señal analógica o se mandan para construir una nueva, se mide en Hertz (Hz) o ciclos por segundo. Mayores frecuencias de muestreo nos permiten grabar y/o procesar sonidos de frecuencias más altas, sin embargo, entre más muestras por segundo se deseen guardar, mayor será la capacidad de memoria que se ocupará para almacenar estos datos [5].
- Frecuencia Nyquist: el teorema de Nyquist o también llamado teorema de muestreo nos dicta lo siguiente:

“La velocidad mínima a la que una señal puede ser muestreada y reconstruida a partir de las muestras debe ser más de dos veces la frecuencia máxima contenida en la señal” [6].

Por lo tanto, esta frecuencia de muestreo mínima se llama frecuencia de Nyquist o frecuencia mínima de muestreo. De acuerdo con esta definición de la frecuencia de Nyquist, 44.1kHz es usualmente la frecuencia con la que se trabaja en producciones de audio-audiovisuales, ya que nos permite la reproducción de todo sonido con frecuencias igual o menores a 22.05 KHz. Esto cubre de manera general todas las frecuencias que percibe una persona promedio. Aunque aún existe el debate acerca de este intervalo de percepción general, es usualmente aceptado que muy pocas personas pueden percibir tonos arriba de los 20 kHz [2].

- Canales de audio: el número de canales del audio digital se refiere a que el contenido del audio se puede repartir en diversos canales, donde cada uno es representado con su propia señal. El caso más general es el estéreo, donde cada canal representa la salida para cada oído -izquierdo o derecho-.
- Escala de decibeles: usualmente, en parámetros de frecuencia del sonido lidiamos con cantidades que pueden abarcar un intervalo muy amplio de valores. La escala de de-

cibeles, es una manera muy útil que tenemos para representar esas cantidades como la relación entre dos valores de manera logarítmica. De manera típica estos valores representan potencias, por lo que nuestro decibel (dB) no tiene unidades. Un valor en decibels se obtiene tal que:

$$x = 10 \log_{10} \left( \frac{x^2}{x_0^2} \right) = 20 \log_{10} \left( \frac{|x|}{|x_0|} \right) \quad (2.1)$$

Otros conceptos que en la presente tesis utilizaremos repetidas veces son los efectos de audio y los efectos especiales. El uso de la palabra efecto de manera general, es usado desde el punto de vista de un sujeto que está observando un fenómeno y ocurre un suceso que denota una impresión o un cambio en la percepción del sujeto. Este cambio es el resultado de la causalidad de un primer efecto. Por lo tanto, si transportamos este ejemplo al terreno del audio, un efecto de audio es el cambio en una señal tal que esta transformación modifique cómo la señal cambiada es percibida respecto a la original, es decir, un efecto es la transformación de un sonido a otro sonido de tal forma que el sonido resultante se perciba de manera diferente por el sujeto que los escucha. A su vez, en los parámetros de diseño de efectos digitales, el concepto del efecto también corresponde a la herramienta que es usada para aplicar una o más transformaciones en los sonidos para alcanzar el cambio de percepción en el audio.

Ciertas modificaciones en los sonidos ocurren de manera natural, por ejemplo, debido a la interacción que tiene el sonido cuando pasa a través del aire al rebotar con objetos en el ambiente o al pasar por resonadores acústicos y habitaciones. Como estos cambios en nuestra percepción suceden todo el tiempo, los podemos considerar hasta cierto punto como “normales” para el oído humano. Sin embargo, existen otro tipo de modificaciones en los sonidos que casi nunca o específicamente nunca ocurren de manera natural. De esta manera se define a los “efectos especiales” como aquellos que procesan una señal de manera que el resultado final no se clasifique como un efecto natural tales como los causados por vibraciones en el aire o rebotes apenas distinguibles por el ser humano. Ejemplos de efectos especiales pueden ser los filtros que afectan la fase, pero no la frecuencia de la señal, filtros que por el contrario cambian la frecuencia sin afectar la fase, representación de rebotes y reflexiones suponiendo paredes en constante movimiento periódico, diferentes tipos de distorsión, entre muchos otros [4].

Los efectos de audio generalmente son utilizados para mejorar la experiencia del escucha, o darle una perspectiva diferente del sonido. Los efectos digitales de audio, aunque comúnmente son utilizados por compositores, interpretes e ingenieros -de audio-, son descritos por los desarrolladores e ingenieros que los diseñan en un Procesador Digital de Señales, por lo tanto, su clasificación, documentación y referencias varían y dependen siempre de las nuevas tecnologías y técnicas de procesamiento que van mejorando con el paso del tiempo [7].

Los efectos de audio digitales pueden organizarse dependiendo de su técnica de implementación, de manera que se propone la siguiente clasificación tal que los efectos que serán implementados en esta tesis puedan ser incluidos en alguna de las siguientes categorías:

- Por retardos
- Moduladores y demoduladores
- Por procesamiento no lineal
- Efectos espaciales
- Por filtrado

Esta clasificación es tomada a partir de [7] y de la Tabla 2.1 que se presenta a continuación:

Nombre del efecto	Atributos Perceptibles	
	Principal	Otros
<b>Efectos basados en retardos (D)</b>		
Vibrato	D	T
Flanging	D	T
Coros	D	
Slapback	D	
Eco	D	S
Doubling	D	
<b>Efectos basados en Filtros (F)</b>		
Ecualización	F	A
Wah-Wah	F	A
<b>Efectos basados en cambio y modulación de Amplitud (A)</b>		
Compresor	A	T
Limitador	A	T
Normalización	A	
Trémolo	A	
<b>Efectos basados en aspectos espaciales (S)</b>		
Cambio de distancia	S	A,T
Efecto Doppler	S	A
Reverberación	S	A,D,T
Paneo (2D,3D)	S	
Paneo espectral	S	A,T
Directividad	S	T
<b>Efectos basados en el timbre (T)</b>		
<i><b>Efectos de fase</b></i>		
Phaser	T	F
<i><b>Efectos de estructura del espectro</b></i>		
Cambio de textura	T	
Desplazamiento del espectro	T	
<i><b>Efectos no lineales</b></i>		
Distorsión	T	A
Fuzz	T	A
Saturación	T	A

(Sigue en la página siguiente)

Nombre del efecto	Atributos Perceptibles	
	Principal	Otros
Armonizador de espectro	T	
Mutación	T	A
Interpolación espectral	T	A

Tabla 2.1: Clasificación de los efectos de audio de acuerdo a sus atributos perceptibles (D por retardo, F por filtrado, A por amplitud, S por espaciales y T por timbre), adaptado de [7].

### 2.1.1. Efectos por retardo

Los seres humanos tenemos habilidades de percepción auditivas muy variadas, es por esto que en este campo suele ser muy difícil tener una generalización de parámetros auditivos fijos y en su lugar se utilizan intervalos para la descripción y clasificación de los eventos que podemos percibir. Una de las primeras personas que pudo medir correctamente los parámetros de percepción de diferentes efectos de sonido en diversas personas fue Carl E. Seashore, parámetros tales como el tiempo de discriminación que existe entre diferentes niveles de clasificación al percibir la llegada de distintos sonidos [4]. El nivel más simple es la habilidad de identificar cuando dos sonidos ocurren simultáneamente, pasando hasta el último nivel donde se tiene un tiempo en el que las personas podían diferenciar entre el sonido original y el sonido secundario como un eco completamente separado.

Resulta entonces, que los intervalos que arrojaron estas primeras experimentaciones fueron que sólo un porcentaje muy pequeño de la población de estudio podía discriminar simultáneamente dos sonidos diferentes con una separación de 5 milisegundos (ms) entre ellos, y la mayoría de esa población podía definitivamente identificar que dos sonidos no eran simultáneos cuando tenían 35 ms de separación. Como podemos notar existe un intervalo de percepción entre los 5 ms y los 35 ms en donde, si además proveemos al escucha con suficientes versiones ligeramente diferentes de un sonido original, podemos engañar su percepción, haciendo creer que realmente hay muchas fuentes únicas para cada sonido, sobrecargando su habilidad de elegir entre una individualmente y así determinar que sólo son copias de una misma fuente. El efecto de coros está basado en este principio [8].

Los efectos por retardos han sido popularizados por músicos famosos como Les Paul y Elvis Presley que hacían un gran uso de estas herramientas en sus discos. Originalmente, este efecto se alcanzaba poniendo una cinta en bucle con velocidad variable para posteriormente grabarlo y volverlo a reproducir con un ligero reposicionamiento de la cinta. Hoy en día la forma de lograr este efecto es muy similar, sumando la señal de vuelta con un buffer de muestras de retardos utilizando usualmente apuntadores circulares. La gran diferencia es que los ligeros cambios en la señal de origen se crean al momento de la suma de los retardos con la señal original de entrada, multiplicándolos por un factor de escala que regularmente es menor a la unidad. Técnicas más avanzadas pueden tener involucradas mejores formas de control para establecer el tiempo de los retardos y de la retroalimentación de manera más fácil y automática de manera que éste se adapte a la velocidad de la música o en todo caso

buscar alguno de sus múltiplos que lo haga [4].

Muchas veces se intentan crear efectos de reverberación usando una retroalimentación de efectos por retardos de manera modular, sin embargo, es común que estos efectos suenen muy poco realistas debido a que las adiciones de los retardos -generalmente aplicados con un escalamiento lineal descendente- no representan de manera lógica la forma de una habitación para el oído humano.

De manera más estándar en la industria musical, el efecto por retardos que considera el tiempo de retardo más bajo es el “flanger”, yendo desde la percepción natural entre dos sonidos hasta los 15 ms de tiempo de separación. La clasificación del efecto de coros abarcaría según esto, desde el término del efecto flanger hasta el inicio de la percepción de dos sonidos como completamente separados, evento que sucede aprox. a los 30 ms. Por último, si bien es cierto que muchos autores consideran el inicio del efecto eco inmediatamente posterior al límite del efecto de coros, en otros estándares se considera el inicio de esta clasificación hasta los 50 ms, dejando al intervalo intermedio los nombres del efecto “doubling” o “slapback”. A pesar de ello, el estándar para el límite superior del efecto de coros no está totalmente definido y muchas veces la implementación de este efecto se propone a los 100 ms en promedio. Esto aparece representado a continuación en la Tabla 2.2.

Nombre del efecto	Modulación	Rango del retardo
Flanger .....	Sinusoidal .....	0 – 15[ms]
Coros .....		10 – 30[ms]
Slapback .....		25 – 50[ms]
Eco .....		> 50[ms]

Tabla 2.2: Efectos de audio típicos basados en retardos, adaptado de [7].

Con base en lo anterior, podemos expresar un efecto por retardos a través de una función tal que la salida de muestras de audio esté dada por la expresión discreta  $y[n]$  y la entrada de muestras de audio sea dada por  $x[n]$ . El retardo está dado por un número entero de muestras  $N$  y la ganancia de la señal con retardo será un valor  $g$ . De esto obtenemos la ecuación (2.2)

$$y[n] = x[n] + g * x[n - N] \tag{2.2}$$

Una manera de representar la ecuación 2.2 para múltiples retardos se muestra en el diagrama de bloques del efecto coros de la figura 2.1, en donde podemos observar la representación del efecto para un número de  $N$  retardos para una sola entrada  $x[n]$ , sus respectivas ganancias en cada retardo y su respectiva señal de salida  $y[n]$ .

Esta función del efecto por retardos es una función lineal e invariante en el tiempo. Por lo tanto, si aplicamos la función de retardo a la suma de dos señales, por su linealidad obtendremos a la salida la suma de las dos señales con el retardo previo, de la siguiente



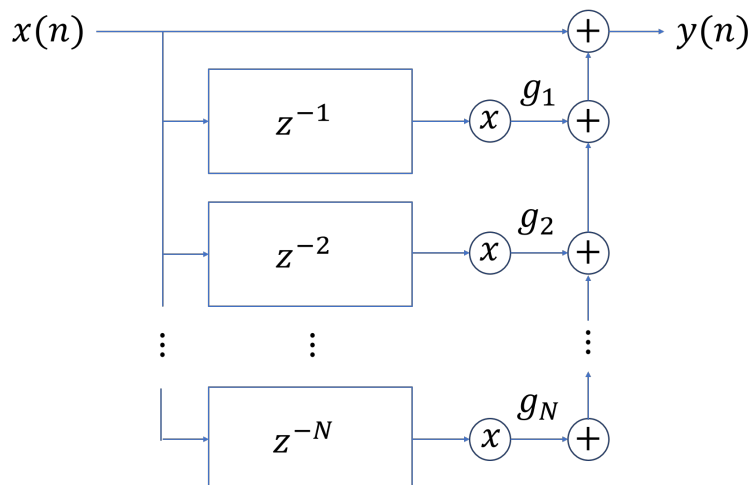


Figura 2.1: Diagrama de bloques de efecto coros, adaptada de [7].

manera:

$$y[n] = x_1[n] + x_2[n] + g * (x_1[n - N] + x_2[n - N]) \quad (2.3)$$

$$y[n] = (x_1[n] + g * x_1[n - N]) + (x_2[n] + g * x_2[n - N]) \quad (2.4)$$

$$y[n] = y_1[n] + y_2[n] \quad (2.5)$$

Así mismo, que la función sea invariante en el tiempo implica que, si se añade un desplazamiento en la entrada, tendremos el mismo desplazamiento a la salida, tal que:

$$y_d[n] = x[n - M] + g * x[n - M - N] = y[n - M] \quad (2.6)$$

Gracias a que la función por retardos es una función discreta, podemos utilizar como herramienta la transformada Z, y así encontrar la respuesta en frecuencia de este retardo básico, como se muestra a continuación:

$$Y(z) = X(z) + g * X(z) * z^{-N} \quad (2.7)$$

$$Y(z) = X(z)[1 + g * z^{-N}] \quad (2.8)$$

con función de transferencia:

$$H(z) = \frac{Y(z)}{X(z)} = 1 + g * z^{-N} \quad (2.9)$$

$$H(z) = \frac{z^N + g}{z^N} \quad (2.10)$$

Debido a que, de la función de transferencia  $H(z)$  no tenemos polos fuera del círculo unitario, la función básica por retardos debe ser estable para todos los casos. Sin embargo, para que nuestro sistema no presente un sobre flujo, y que el sonido del retardo tenga una mayor percepción que nuestro sonido original debemos cumplir con la siguiente condición:

$$|g_{FB}| < 1 \quad (2.11)$$

Este resultado coincide con nuestra intuición, donde, sólo cuando tengamos una ganancia menor a la unidad, la presencia de una muestra con retardo será más suave a través del tiempo.

### 2.1.2. Efecto de reverberación (efecto espacial)

En condiciones naturales cuando un escucha percibe un sonido, este sonido se propaga a través de un medio desde la fuente hasta el individuo y durante este viaje el sonido original es ligeramente modificado por el ambiente. Por lo tanto, se sabe que existen efectos espaciales impuestos por características físicas y geométricas del medio en donde se transmite el sonido de la fuente hasta los oídos del escucha. En cuestiones generales, el efecto de reverberación o los efectos espaciales son estos efectos creados a partir de la estructura donde opera la transmisión del sonido. Gracias a la existencia de estos efectos naturales el ser humano puede percibir la localización de fuentes de sonido y tener la capacidad de percibir algunas de las propiedades del espacio en el que se encuentran.

La reverberación está compuesta por reflexiones retrasadas variables de copias atenuadas de un sonido directo u original. En una habitación o en cualquier ambiente acústico existe un camino directo entre la fuente de sonido y el escucha, a pesar de ello, el sonido puede tomar caminos más largos si estos llegan a colisionar y rebotar con paredes, el techo o distintos objetos antes de llegar al escucha. Estos caminos producidos por rebotes son los causantes del retraso en las reflexiones que componen la reverberación.

Existen muchas propiedades de estas reverberaciones que se relacionan con la acústica de una habitación y se usan para hacer una descripción del espacio. El concepto más importante para describir dicha acústica, propia de la habitación, que dota a un sonido de ciertas propiedades es la respuesta al impulso. Esta respuesta al impulso regularmente se divide en tres partes: el sonido directo, las reflexiones tempranas o próximas y las reflexiones tardías [7].

El sonido directo es el primer sonido que llega a los oídos del escucha. Las reflexiones tempranas son las primeras reflexiones que no se separan de la fuente debido a que el oído humano las integra con el sonido directo. A pesar de que estas reflexiones no se perciben como de una fuente diferente debido al tamaño muy pequeño del retardo que las separa, sí contribuyen al cambio en el color del sonido y la percepción del tamaño de la fuente. Las reflexiones tardías se empiezan a percibir después de cierto tiempo que usualmente recibe el nombre de “tiempo de mezclado”. Estas reflexiones se caracterizan por ser percibidas como procedentes de otras fuentes, sin embargo, es tal la densidad de estas primeras reflexiones tardías que sus componentes individuales no se pueden observar en la respuesta. A diferencia de las reflexiones tempranas, las reflexiones tardías sí nos dan pistas para conocer el tamaño de la habitación y/o la distancia a la que se encuentra la fuente de sonido [7].

La reverberación es más que una serie de ecos anidados. Como vimos antes, el eco es el resultado de una versión atenuada y retrasada de un sonido, que se puede percibir arriba de retrasos mayores a 50 ms. En el caso de la reverberación de un cuarto típico, existen muchas reflexiones (como se observa en la Figura 2.2) las cuales, en un principio, debido a su corto periodo de llegada, las percibimos como una combinación de todas las reflexiones -reflexiones tempranas-. Pero a medida que decae el tiempo de llegada, -reflexiones tardías-, las percibimos de manera más espaciada y con una caída en su amplitud de forma exponencial [2].

Las habitaciones pequeñas tienden a tener tiempos de reverberación de cientos de milisegundos, en cambio, habitaciones más grandes usualmente poseen un tiempo de reverberación mayor, debido a que el sonido viaja distancias mayores entre cada reflexión. Las salas de conciertos, diseñadas para tener efectos de reverberación mucho más notorios, llegan a tener tiempos de reverberación de hasta 1.5 a 2 segundos. Las catedrales y otros ambientes muy ricos en reverberación pueden llegar a poseer tiempos de reverberación de más de 3 segundos[9]. Algunos ejemplos se observan en la Tabla 2.3.

Auditorio	Tiempo de Reverberación [s]		
	$f = 125Hz$	$f = 500Hz$	$f = 2000Hz$
Royal Festival Hall, London	1.4	1.5	1.4
Opera House, San Francisco	-	1.7	-
Carnegie Hall, New York	1.8	1.8	1.6
Symphony Hall, Boston	2.2	1.8	1.7
Musikvereinsaal, Vienna	-	2.05	-
Concertgebouw, Amsterdam	2.2	2.1	1.8
Royal Albert Hall, London	3.4	2.6	2.2

Tabla 2.3: Ejemplos de tiempos de reverberación de salas de conciertos famosas dependientes de la frecuencia, adaptado de [10, 9].

Algunos de los métodos para crear ambientes de reverberación de manera digital, consisten de manera principal de dos tipos de filtros: de respuesta al impulso infinita (IIR) y filtros paso todo (en el intervalo de frecuencias del oído) combinados con filtros supresores de banda. Con ello se busca producir la caída gradual en la serie de sus reflexiones, así como sus retardos entre cada una, sin embargo, aunque ésta es de las técnicas más utilizadas, existen otros métodos para intentar representar este efecto. [4] Ahora bien, una constante que todos los métodos comparten a la hora de caracterizar la reverberación de un espacio acústico dado, es el tiempo de reverberación que se denota como: “el tiempo que le toma al nivel de presión del sonido o su intensidad decaer una constante de 60dB - equivalente a  $1/1,000,000$  de su intensidad original -, o lo que es lo mismo,  $1/1,000$  de su amplitud de onda original – esta caída también recibe el nombre de piso de ruido-” [11].

Con base en esto, al igual que consideramos una función que describe el efecto del retardo simple, podemos representar de forma muy parecida a la función del efecto de reverberación,

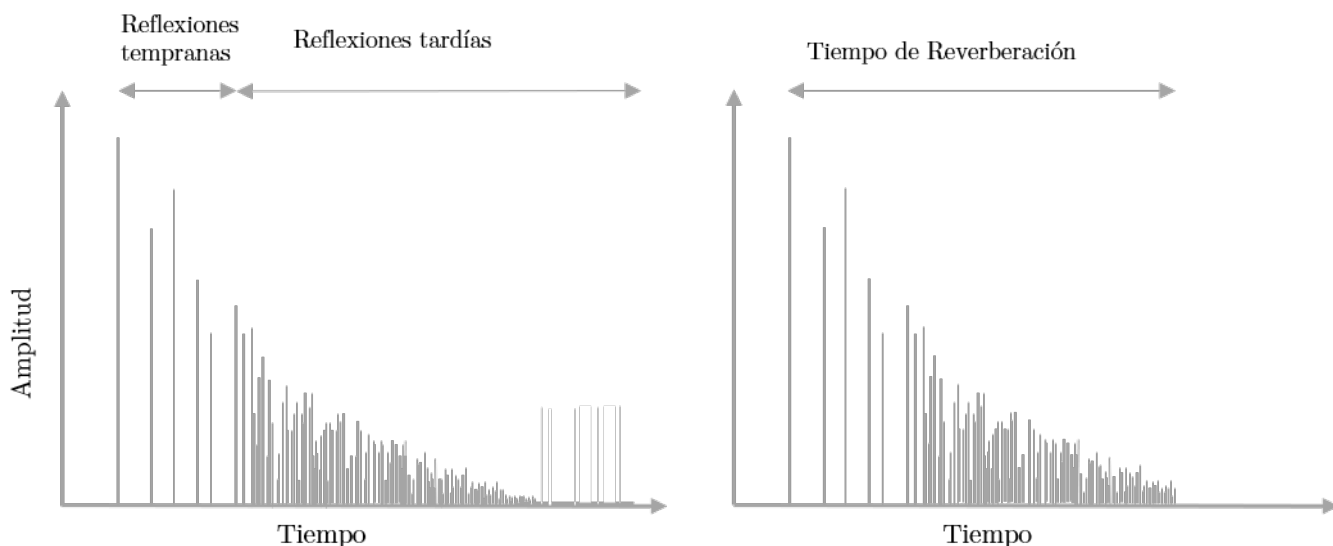


Figura 2.2: Diagrama de muestras de reflexiones tempranas, reflexiones tardías y tiempo de reverberación en efecto de reverberación, adaptado de [2].

dónde, para una muestra de reverberación tenemos la ecuación (2.12)

$$y_{rev}[n] = x[n] + a^N * x[n - N] \quad (2.12)$$

Dónde  $y_{rev}[n]$  y  $x[n]$  son las señales discretas a la salida y a la entrada del efecto de reverberación respectivamente,  $N$  es el número de muestras que abarca el retardo de la señal sumada y que su máximo está limitado por el tiempo de reverberación del efecto dependiendo de las características físicas de la habitación que se quiere simular.

A diferencia del caso del efecto por retardos, aquí tenemos una ganancia definida por una magnitud exponencial de base  $a$ . Esta constante está directamente relacionada al tiempo de reverberación de la señal de reverberación que obtendremos, -que es una variable específica del efecto- y del concepto de piso de ruido, de tal manera que, sabiendo que el piso de ruido equivale a un factor del 0.001 de la amplitud original y esta ganancia se presenta en el tiempo de reverberación donde tendremos nuestro retardo máximo a considerar, y nuestra variable  $N$  es igual a una  $N_{max}$ , tenemos que la constante  $a$  está dada por la ecuación (2.13).

$$a^{N_{max}} = \frac{1}{1000} = 0.001$$

$$a = \sqrt[N_{max}]{0.001} \quad (2.13)$$

Dónde, de nueva cuenta  $N$  equivale al número de muestras totales de nuestro tiempo de reverberación dependiendo de nuestra frecuencia de muestreo.

De igual manera y debido a que nuestra función que describe al efecto de reverberación comparte las mismas propiedades de linealidad e invariancia en el tiempo de nuestra función por retardos, también comparte las propiedades de estabilidad en el tiempo tanto para la

función simple como para una función retroalimentada, siempre y cuando cumpla con las mismas condiciones de ganancia para este último caso. Así que, recordando la condición del efecto por retardos para cumplir con la estabilidad tal que su ganancia  $g_{fb}$  tiene que ser menor que la unidad, podemos transportarlo a la función de reverberación dónde observamos que ahora  $a^N$  tiene que ser menor a la unidad y por lo tanto inducimos que:

$$|a| < 1 \quad (2.14)$$

Otra limitación que suele presentar la variable  $N$ , pero en este caso como valor mínimo, está dado por el concepto de las reverberaciones tempranas y reverberaciones tardías. Como mencionamos antes, regularmente las reverberaciones tempranas se perciben en el oído como parte del sonido original, mientras que las reverberaciones tardías, con retrasos superiores a los  $50ms$ , son percibidas ya como una anidación de ecos. Es por eso, para la representación del efecto de reverberación se pueden despreñar las sumas de reverberaciones tempranas y nuestra variable  $N$  inicie con un valor de muestreo equivalente al menos a los  $50ms$  proporcionales a la percepción de un retardo de tipo eco, como se observa en (2.15)

$$\vec{N} = [Fs * 50ms \text{ del retardo del eco} \rightarrow Fs * \text{ tiempo de reverberación del efecto}] \quad (2.15)$$

Una señal de audio creada digitalmente o grabada en un lugar artificial sin la adición de efectos puede llegar a sonar artificial. La adición de la reverberación en grabaciones es usado para compensar el hecho de que no siempre podemos escuchar música o películas en ambientes diseñados para este rubro, por ejemplo, el ambiente dentro de un auto o en una habitación pequeña no puede recrear el sonido majestuoso de una sinfonía de orquesta en un auditorio, y si consideramos que la mayoría de música en estos días se escucha a través de audífonos, no obtenemos el efecto de una reverberación natural en absoluto.

En el ámbito musical los efectos de reverberación son implementados para proveer una caída del sonido más gradual, en lugar de tener un corte limpio que conlleve a un silencio abrupto. En las percusiones, regularmente se usan efectos de reverberación cortados con el objetivo de que los golpes suenen más poderosos o contundentes. Tanto en la industria cinematográfica como en el desarrollo de realidad virtual, regularmente se busca simular habitaciones reales que puedan ser convincentes y procedan a crear una imagen de video estéreo de manera virtual. Este tipo de sonidos son conocidos como “Q-sound” o sonido 3-D virtual y son producidos generalmente usando las primeras reflexiones tardías que se crearían en una habitación lógica [2].

Aunque la mayoría de veces la adición de efectos de reverberación buscan el concepto de recrear una habitación real, como se mencionó antes, existen otros casos en los que el productor busca utilizar el efecto a tal grado de tener una experiencia sobrecargada y menos natural. Este tipo de técnicas se han utilizado últimamente de manera popular en la creación de muchos temas de música electrónica.

### 2.1.3. Efectos por distorsión (efectos no lineales)

Una gran parte de los efectos que comúnmente se utilizan en los sistemas de sonido cumplen con la descripción de sistemas lineales. En estos sistemas lineales tenemos que, si dos señales

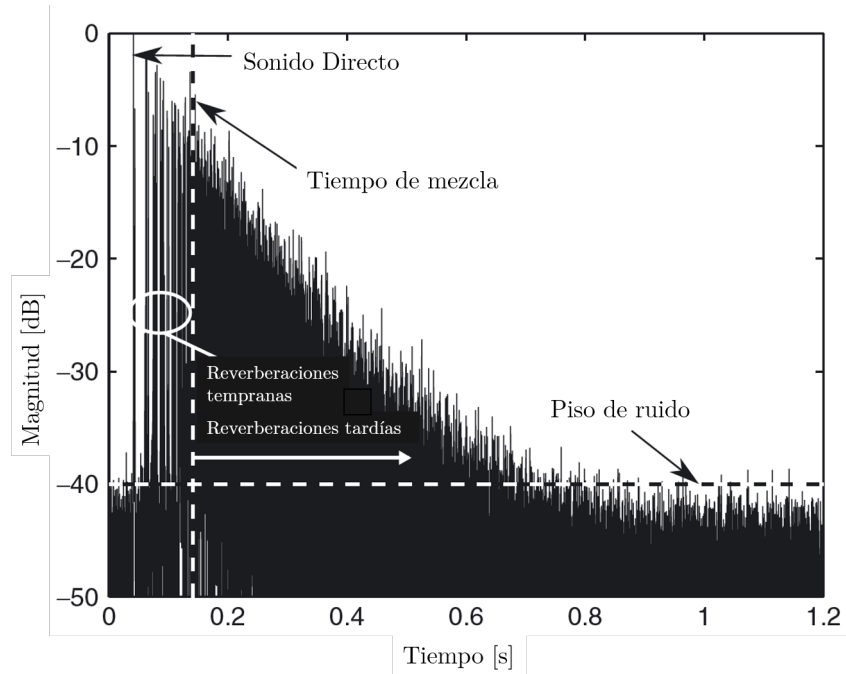


Figura 2.3: Respuesta experimental del efecto de reverberación que muestra el sonido directo, reverberaciones tempranas, tardías y piso de ruido, tomado de [7].

son sumadas a la entrada del sistema y posteriormente esta suma es procesada, el resultado es equivalente a procesar cada entrada de manera individual y después, sumar los dos resultados. Lo mismo ocurre si dichas entradas son multiplicadas por un valor escalar antes de su procesamiento: el resultado será el mismo que si procesamos las dos entradas y después multiplicamos su resultado por los respectivos valores escalares.

Los efectos conocidos como saturación, distorsión y fuzz -comúnmente englobados todos en la categoría de efectos por distorsión- no cumplen con esta descripción de linealidad. De hecho, si consideramos ciertas formas de onda en la señal de entrada, podemos encontrar que para la mayoría de estos efectos y sus variantes existen funciones que los describen adecuadamente y cuyas curvas características son no lineales.

Para nuestro primer efecto por distorsión, definiremos **la saturación** como un estado en el que un dispositivo de efectos de audio es casi lineal a un nivel de entrada bajo o de señal pequeña es llevado a través de un aumento escalar en su señal de entrada a la región no lineal de su curva característica. Regularmente esta transición en la operación de la región lineal a la región no lineal de la curva característica se busca que sea suave o brusca dependiendo de cómo queremos que se perciba el efecto por el escucha, sin embargo, se debe tener claro que el sonido característico del efecto está dado sólo por el intervalo de la zona en la que se trabaja en la región no lineal [7].

La función matemática que describe este efecto con un recorte suave de la señal se observa

a continuación en la ecuación (2.16)

$$f(x) = \begin{cases} 2x & ; 0 \leq x \leq \frac{1}{3} \\ \frac{3 - (2 - 3x)^2}{3} & ; \frac{1}{3} \leq x \leq \frac{2}{3} \\ 1 & ; \frac{2}{3} \leq x \leq 1 \end{cases} \quad (2.16)$$

A partir de la ecuación (2.16) podemos observar que cuando la señal de entrada  $x$  normalizada es menor a un tercio de la unidad el efecto trabaja dentro de los parámetros de linealidad, y mientras esta señal aumenta su nivel, existe una transición gradual a la zona no lineal.

A diferencia del efecto de saturación, nuestro segundo efecto **de distorsión**, -el efecto llamado de la misma manera-, opera de manera principal en la región no lineal de la curva característica para todas las señales de entrada hasta llegar al nivel máximo de la entrada, dónde la salida también es truncada a un nivel máximo relativo. Una característica que este efecto comparte con la saturación, es su clasificación por producir un corte “suave” o “duro” en la señal de salida, que, de manera práctica, es una característica que se percibe en las esquinas de los picos positivos y negativos en la señal de salida: un recorte suave de la señal producirá unas esquinas más redondeadas en los picos, mientras que un recorte duro en la señal producirá esquinas más afiladas, tal como se puede observar en la figura 2.4.

El recorte duro de una señal casi siempre se relaciona con un truncamiento en el nivel máximo de una señal de datos en el ámbito digital, o con un desbordamiento en el tema de los procesadores, sin embargo, en términos analógicos, muchos amplificadores se basan en transistores para producir este recorte duro. En el ámbito musical, se utilizan diodos de silicio para alcanzar este mismo efecto [2].

La función matemática que describe el efecto de distorsión con un recorte suave de la señal se muestra a continuación en la ecuación (2.17).

$$f(x) = \operatorname{sgn}(x)(1 - e^{-|Gx|}) \quad (2.17)$$

Dónde la función  $\operatorname{sgn}$  de  $x$  refiere al signo original del valor de entrada  $x$  que también puede ser representada por  $\operatorname{sgn}(x) = \frac{x}{|x|}$ . El valor  $Gx$  en la potencia del número de Euler es el valor de la entrada  $x$  multiplicada por un escalar  $G$ .

Este valor  $G$  es la ganancia que se le aplica a la señal de entrada antes de ser procesada por su la función no lineal, y mientras más crezca este valor escalar notaremos un mayor cambio en el cómo suena el efecto. Por lo tanto, deducimos que, para casi toda señal de entrada al sistema, una mayor ganancia producirá una mayor distorsión en la salida, sin embargo, se debe notar que el aplicar una mayor ganancia no varía la amplitud máxima de la salida, debido a que el nivel de recorte permanece en el mismo lugar.

Para el último de los casos de nuestro efecto de distorsión, el efecto “fuzz” tenemos como principal diferencia que este es un efecto completamente no lineal, y al trabajar todo el tiempo

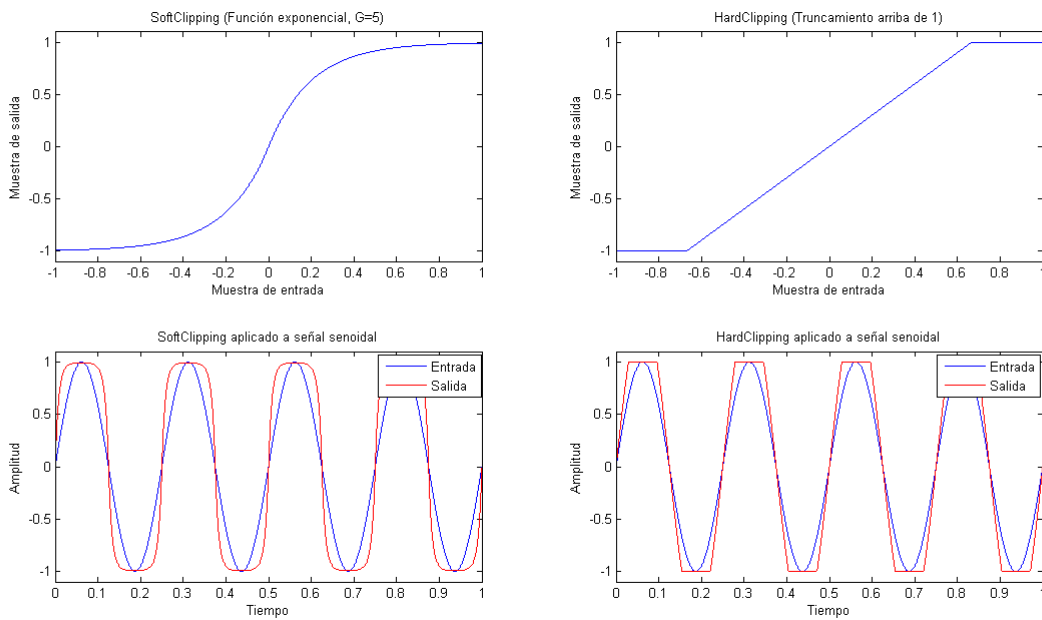


Figura 2.4: Representación de SoftClipping (recorte suave) y HardClipping (recorte duro) de señal senoidal aplicando efecto por distorsión, adaptado de [7].

en la parte no lineal de su curva característica presenta cambios más drásticos para la señal de entrada. Generalmente en términos musicales, este efecto se percibe poco placentero y muy sobrecargado, debido a que resulta en la pérdida de nitidez de la señal original.

Gracias al recorte y por ende a la modificación de la forma de onda de la señal de entrada, o en términos teóricos, a que nos encontramos operando en la región no lineal de una curva característica es evidente que añadimos distorsión armónica a la señal original. Debido a esto, podemos considerar que los efectos de distorsión crean un sonido espectralmente más rico, como se observa en la figura 2.5, lo cual es usado en la industria musical para ayudar a algunos instrumentos a lograr prominencia en una mezcla. Por ejemplo, una cuerda de guitarra punteada representa un solo tono puro que se puede encontrar en el espectro clasificado como bajo, medio o alto, sin embargo, al agregar distorsión al punteo, se crearán armónicos que se pueden llegar a extender hasta el final o el inicio del espectro de audio.

Como consecuencia usualmente los efectos de distorsión se pueden combinar de manera útil con otros efectos de filtrado como el “wah-wah” o un post-ecualizador, ya que estos efectos pueden aprovechar la creación de armónicos para resaltar aún más estos nuevos intervalos de frecuencias que antes no existían. Por ejemplo, si se coloca un pedal “wah-wah” después de un efecto de distorsión, el efecto “wah-wah” puede ser mucho más pronunciado que si se usa en una señal de guitarra limpia.

Una de las mayores ventajas de procesamiento de este efecto es que sólo depende de la curva característica que define al efecto a utilizar, siendo independiente de la memoria del procesamiento a diferencia de todos los anteriores ya vistos. De esta manera decimos que la



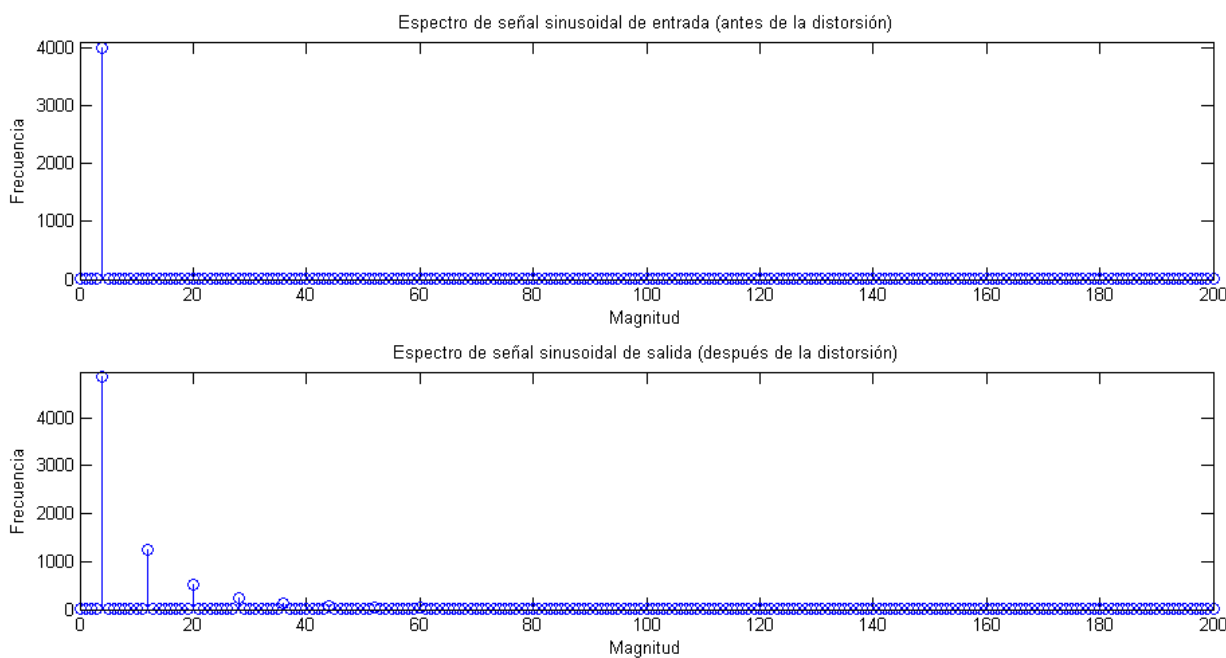


Figura 2.5: Respuesta en frecuencia de una señal periódica antes y después de aplicar efecto por distorsión, adaptado de [7].

muestra de salida del efecto sólo depende de la muestra de entrada actual y no de alguna otra entrada o salida anterior. Aun así, sabiendo que estos efectos no están limitados por la capacidad de memoria que tiene el sistema, sí están limitados a la magnitud de la señal que se puede procesar. En los sistemas digitales este límite suele estar determinado por el número de bits que soporta el convertidor analógico a digital y el convertidor digital a analógico.

#### 2.1.4. Efectos por filtrado.

La ecualización es uno de los efectos de audio más comunes y más utilizados. El concepto de ecualizar hace referencia al proceso de ajustar la ganancia relativa de distintas bandas de frecuencia en una señal. El nombre de ecualización viene del deseo de obtener una respuesta en la frecuencia plana o de igual magnitud en todo su intervalo en un sistema de audio. Compensando de esta manera las no linealidades del equipo o de la acústica de la habitación. Los picos y los valles en la respuesta en frecuencia de un sistema se describen generalmente como “coloración” en términos musicales, por lo tanto, la implementación de ecualizadores en este sistema es recurrente y tiene como objetivo remover o resaltar esta coloración [2].

Todos los efectos que conocemos de ecualización están basados en filtros, que son las herramientas que nos ayudan a corregir las deficiencias en la respuesta frecuencial de un sistema o el balance en los tonos de las fuentes, resaltando frecuencias originalmente débiles o atenuando algunas otras de nivel excesivo. Existen dos tipos principales de ecualizadores: *ecualizadores gráficos o de bandas* y los *ecualizadores paramétricos*. Los primeros se refieren a una composición de filtros dedicados al control de diversas bandas fijas -normalmente entre

5 y 31-, que varían entre ellas por una octava o una fracción de octava dependiendo del caso. Los ecualizadores paramétricos son aquellos que tienen la capacidad de ajustar la frecuencia central de una o más bandas, así como su extensión mediante el control del factor de calidad  $Q$ . En ambos casos se utiliza para cada banda o parámetro un ajuste de ganancia relativa (ganancia o atenuación) [12].

Las bandas en un ecualizador, generalmente se distribuyen de manera logarítmica en la frecuencia con el objetivo de relacionarse con la percepción logarítmica de la escala de frecuencias del oído humano, por lo tanto, se pueden expresar como múltiplos de fracciones de octavas -es por esto que existen denominaciones de ecualizadores tales como ecualizadores de octava, de tercio de octava, de media octava, etc. - [12].

La estructura de las bandas superiores e inferiores o bandas de corte en los ecualizadores logarítmicos está representada en la ecuación (2.18).

$$f_{s,k} = 2^\alpha f_{i,k} \quad (2.18)$$

Dónde:

- $\alpha$  se denomina a la fracción de octava que corresponde a cada una de las bandas que conforman nuestro ecualizador.
- $k$  es una banda cualquiera de nuestro ecualizador que cumple con el cálculo de la ecuación (2.18)
- $f_{i,k}$  y  $f_{s,k}$  son las bandas inferior y superior de la banda  $k$ , respectivamente.

Conociendo las bandas inferiores y superiores para una banda  $k$  cualquiera en nuestro ecualizador, a partir de la selección previa de su tipo de banda (ancho de banda / fracción de octava), también se conoce la frecuencia central de la banda  $k$ , cuya nomenclatura es  $f_{(o,k)}$ , la cual se define como la media geométrica entre los extremos, por lo tanto, a partir de la ecuación (2.18) tenemos:

$$f_{o,k} = \sqrt{f_{s,k} f_{i,k}} \quad (2.19)$$

$$f_{o,k}^2 = f_{s,k} f_{i,k} \quad (2.20)$$

Y para cada extremo tenemos:

$$f_{s,k} = 2^{\frac{\alpha}{2}} f_{o,k} \quad (2.21)$$

$$f_{i,k} = 2^{-\frac{\alpha}{2}} f_{o,k} \quad (2.22)$$

Dónde el ancho de banda relativo a la banda  $k$  se expresa de la siguiente manera:

$$B = \frac{f_{s,k} - f_{i,k}}{f_{o,k}} = 2^{\frac{\alpha}{2}} - 2^{-\frac{\alpha}{2}} \quad (2.23)$$

Por lo tanto, la relación entre frecuencias en bandas contiguas queda de la siguiente manera:

$$f_{i,k} = f_{s,k-1} \quad (2.24)$$

$$f_{s,k} = f_{i,k+1} \quad (2.25)$$

Esto implica qué:

$$f_{o,k} = 2^\alpha f_{o,k-1} = \frac{f_{o,k+1}}{2^\alpha} \quad (2.26)$$

Es decir, la relación entre las frecuencias centrales entre bandas contiguas es la misma que entre las frecuencias superior e inferior de cada banda.

Los ecualizadores más comunes son los de 10 bandas o de octava, en los que cada frecuencia central de sus bandas es el doble de la banda anterior, y los de tercio de octava, de 30 bandas, en los que cada frecuencia central es aproximadamente un 25 % mayor que la anterior. También existen los ecualizadores de doble octava, que en realidad son controles de tono de 5 bandas y se utilizan generalmente en correcciones de balance en los tonos y no como una solución de problemas severos de origen acústico [12].

Las frecuencias centrales, así como las frecuencias de corte inferior y superior de los ecualizadores de distintas resoluciones están normalizadas nacional e internacionalmente por los estándares ISO y por normas IEC. A continuación, se presentan en la tabla 2.4 las frecuencias estándar para los ecualizadores de octava y de un tercio de octava según las normativas antes mencionadas.

Una octava			Un tercio de octava		
Frecuencia Inferior	Media geométrica (normalizada)	Frecuencia Superior	Frecuencia Inferior	Media geométrica (normalizada)	Frecuencia Superior
$f_i$ (Hz)	$f_o$ (Hz)	$f_s$ (Hz)	$f_i$ (Hz)	$f_o$ (Hz)	$f_s$ (Hz)
22	31.5	44	22.4	25	28.2
			28.2	31.5	35.5
			35.5	40	44.7
44	63	88	44.7	50	56.2
			56.2	63	70.8
			70.8	80	89.1
88	125	177	89.1	100	112
			112	125	141
			141	160	178
177	250	355	178	200	224
			224	250	282
			282	315	355
355	500	710	355	400	447
			447	500	562
			562	630	708
710	1000	1420	708	800	891
			891	1000	1122
			1122	1250	1413
1420	2000	2840	1413	1600	1778
			1778	2000	2239
			2239	2500	2818

(Sigue en la página siguiente)

Una octava			Un tercio de octava		
Frecuencia Inferior	Media geométrica (normalizada)	Frecuencia Superior	Frecuencia Inferior	Media geométrica (normalizada)	Frecuencia Superior
$f_i$ (Hz)	$f_o$ (Hz)	$f_s$ (Hz)	$f_i$ (Hz)	$f_o$ (Hz)	$f_s$ (Hz)
2840	4000	5680	2818	3150	3548
			3548	4000	4467
			4467	5000	5623
5680	8000	11360	5623	6300	7079
			7079	8000	8913
			8913	10000	11220
11360	16000	22720	11220	12500	14130
			14130	16000	17780
			17780	20000	22390

Tabla 2.4: Intervalos estándar para ecualizadores de una octava y un tercio de octava [13, 14].

Como se puede observar en la Tabla 2.3 algunas de las octavas normalizadas sólo son aproximaciones, de tal manera que, en todos los casos de aproximación, el error siempre es menor al 2%. Es costumbre que el diseño de las bandas de los ecualizadores corresponda a la serie de frecuencias teóricas para cada fracción, aunque posteriormente se les rotule con las designaciones normalizadas [13].

Los ajustes de ganancia en los ecualizadores se basan en el aumento de la ganancia o en la atenuación de la misma según el caso elegido. De manera analógica, se representa este ajuste con un potenciómetro deslizante vertical para cada una de las bandas. Este potenciómetro está graduado en decibeles y su posición neutra o central se encuentra en los  $0dB$ , así se encuentra con una ganancia unitaria correspondiente a que la salida es igual a la entrada. En escala ascendente del potenciómetro, nos encontramos con el aumento de la ganancia de la banda respectiva, donde generalmente el valor máximo se encuentra en  $12dB$  de ganancia, es decir, un aumento de 4 veces en la salida con respecto a la entrada, aunque en otros equipos, el estándar permite conmutar entre  $6dB$  y  $12dB$  o tener una ganancia de hasta  $18dB$ . De la misma manera, tenemos la aplicación para la atenuación de las bandas, donde la posición del potenciómetro más baja corresponde típicamente a una ganancia de  $-12dB$ , aquí también puede ser de  $-6dB$  o hasta  $-18dB$  correspondiente de igual manera a una reducción de la señal original en un factor de 4 [12].

### 2.1.5. Paneo de sonido estéreo.

Cuando la señal de sonido viaja de una fuente a un espectador, las señales que recibe cada oído y sus respectivos canales serán diferentes entre ellas. Estas señales se representan por funciones llamadas *Funciones de transferencia relacionadas con la cabeza* (HRTF por sus siglas en inglés) y son dependientes de la dirección de la fuente de sonido con respecto al escucha que se desglosa en las diferencias temporales y espectrales que existen entre el canal derecho e izquierdo de los oídos. A las primeras se les llama Diferencias temporales

interaurales (ITD) y a las espectrales se les llama Diferencias interaurales de nivel (ILD) [15]. Estas diferencias proveen la información necesaria para que el escucha pueda localizar si la fuente de sonido se encuentra en el espacio izquierdo o derecho con respecto a su posición. Las ILD e ITD provienen de la información dentro del llamado cono de confusión, el cual se explica en seguida.

Supongamos un plano vertical llamado *plano medio* el cual divide el espacio con respecto al escucha en izquierda y derecha. Ahora bien, se tiene un ángulo llamado  $\theta$  que es el ángulo entre el plano medio y la fuente de sonido, donde este ángulo crece conforme las manecillas del reloj y cuando se establece en el plano medio, frente al escucha,  $\theta$  es igual a 0. El cono de confusión, está conformado por todo el conjunto de puntos al rededor del ángulo  $\theta$  que cumplan que la diferencia en la distancia de ambos oídos a cualquier punto del cono, es constante [7]. Por lo tanto, podemos verificar que a partir de localizar este ángulo  $\theta$  es posible obtener la información de las diferencias ILD e ITD que nos proporciona el cono de confusión, o viceversa (como funciona la audición).

Estos conceptos se aplicaron a la grabación y reproducción de audio, en donde cada canal representaba una señal (izquierda o derecha) con diferentes características, dependiendo de dónde se modela la dirección de la fuente de sonido, dando paso a lo que se llamó la configuración estereofónica estándar. Esta técnica se empezó a utilizar ampliamente después de la llegada de los discos de dos canales de 45°-45° y ranura única a finales de la década de 1950 para la industria del cine. Hoy en día el formato más utilizado es el posicionamiento de dos altavoces, separados por 60°, en los lados derecho e izquierdo del espectador, como se observa en la figura 2.6. Aunque en la realidad, en el uso doméstico y en los automóviles no se asegura que el escucha se sitúe siempre en el centro de los altavoces, en la industria musical aún se prefiere usar una salida con dos canales bajo esta suposición que una representación mono-fónica. En otros casos, el sistema sí se diseña con las condiciones para que el oyente se asuma en la mejor posición para escuchar los dos canales, como en el caso del audio en las computadoras o los sistemas con audífonos. En la actualidad existen otras técnicas de representación del sonido estereofónico, que proveen más ángulos de libertad, sin embargo, en el proceso se involucran funciones HRTF más complejas, ligadas al estudio de la posición en una habitación y la acústica de la misma.

A partir del concepto de la configuración estereofónica estándar, el paneo de amplitud es la técnica más usada para el posicionamiento de fuentes virtuales. En ella una señal de sonido se aplica a altavoces con diferentes amplitudes tal que

$$x_i(t) = g_i x(t), \quad i = 1, \dots, L \quad (2.27)$$

dónde  $x_i(t)$  es la señal aplicada al altavoz  $i$ ,  $g_i$  es el factor de ganancia correspondiente al canal y  $L$  es el número de fuentes virtuales. El escucha percibirá la dirección de la fuente virtual dependiendo de su respectivo factor de ganancia; esta dirección estimada se llama ángulo de paneo. A partir de la suposición de que el escucha se encuentra en la mejor posición de audición, como en la figura 2.6, podemos relacionar el ángulo de paneo con respecto a la ganancia de cada altavoz, utilizando el teorema de las tangentes [16] tal que

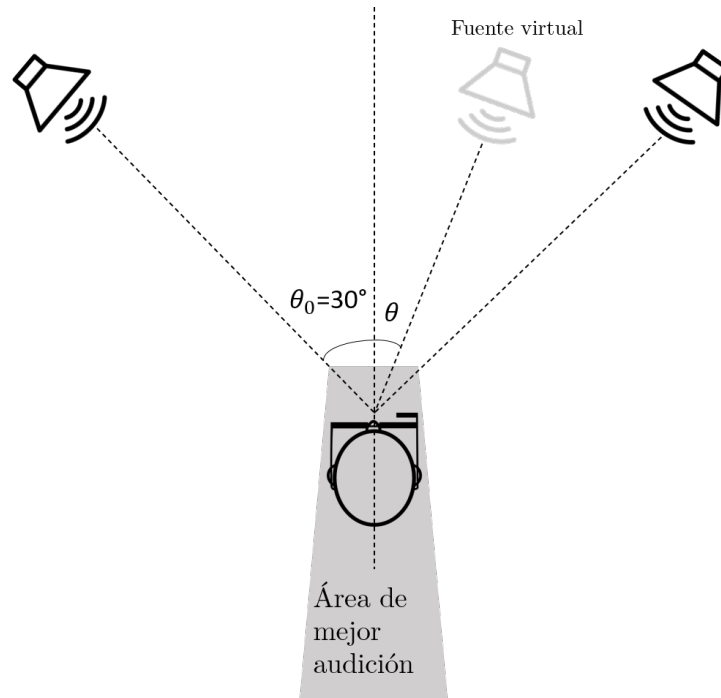


Figura 2.6: Configuración de audición estereofónica estándar, adaptado de [7].

$$\frac{\tan \theta}{\tan \theta_0} = \frac{g_1 - g_2}{g_1 + g_2} \quad (2.28)$$

Las leyes de paneo nos indican que, para evitar un aumento del nivel del audio original en las fuentes virtuales independientemente de su dirección, la suma de los cuadrados de los factores de ganancia deben ser normalizados [17] tal que

$$\sqrt{\sum_{i=1}^L g_i^2} = 1 \quad (2.29)$$

Como se mencionó anteriormente, el análisis presentado parte de la configuración estereofónica estándar y sólo es válido si las fuentes de sonido se encuentran a la misma distancia del escucha, el cual debe encontrarse en el centro y el ángulo entre el primer y la última fuente no es mayor a los 60°. Estas características definen la mejor área de audición para esta técnica, y siempre se parte de la suposición que el receptor se encuentra ahí.

El paneo por amplitud tiene como principal ventaja proveer una diferencia de ITD constante arriba de 1.1kHz, y una diferencia de ILD casi constante a partir de los 2kHz para un escucha en la mejor posición, lo cual limita un efecto de sonido cruzado (por el hecho de que el sonido llega de los dos altavoces a los dos oídos) en altas frecuencias. Adicionalmente, la coloración nula del audio con este método, y la robustez en el cálculo de la dirección con este efecto son probablemente las razones por las cuales el paneo de amplitud es la técnica más utilizada para la representación del movimiento de fuentes virtuales [17].

## 2.2. Resumen

En este capítulo se describieron los conceptos generales de audio, señales digitales y su muestreo, los efectos digitales, su clasificación y se mencionaron los efectos que harán presencia a lo largo de esta tesis. En particular para estos cuatro tipos de efectos especiales, se les dio una breve explicación teórica de su funcionamiento, se mencionan sus clasificaciones y características respectivas y posteriormente se les asigna a cada una las funciones matemáticas que les corresponden y próximamente nos ayudan en el diseño del sistema. Por último, de manera resumida, se hace mención de algunos ejemplos puntuales de dónde se hace uso de estos efectos en el campo de la música y las producciones audiovisuales.





## Capítulo 3

# Acondicionamiento, sensores e interfaz del sistema

Un sistema digital puede ser visto como una serie de bloques en cascada, compuesto de tres partes: la obtención de la señal, el procesamiento y la transferencia de la misma. La obtención de la señal se logra por medio de transductores que transforman la energía del medio en energía eléctrica, esto en el dominio analógico. Sin embargo, la señal muchas veces no cumple con los requerimientos necesarios para ser procesada digitalmente por lo que es necesario acondicionarla utilizando dispositivos electrónicos, como amplificadores operacionales para aumentar la amplitud de la señal, filtros para eliminar ruido no deseado, o algún otro circuito complementario para modificar su offset. La segunda parte toma la señal convertida de analógica en digital y la procesa en otro sistema digital como un microcontrolador, una FPGA o un DSP. El último eslabón transfiere la señal digital a otro sistema, o convierte la señal digital en analógica en donde es acondicionada nuevamente para ser transformada de energía eléctrica a otro tipo de energía haciendo uso de un transductor.

En este capítulo se abordan los temas correspondientes al acondicionamiento de la señal, además de los referentes a la interfaz del usuario con el sistema y el tipo de conexión usado para intercambiar datos: el protocolo Bluetooth. La interfaz se logra mediante una aplicación en un teléfono inteligente desarrollada en la herramienta MIT App Inventor que, manejando diversos bloques y utilizando los sensores que el mismo teléfono posee, se cambian parámetros que modifican la señal de salida.

### 3.1. Acondicionamiento de una señal

El acondicionamiento de una señal significa la manipulación de los parámetros de la misma de tal manera que cumpla los requerimientos de la siguiente etapa en un sistema[18]. Dependiendo de las condiciones del proyecto, el acondicionamiento puede incluir diversas etapas, como lo puede ser amplificación, transformación, conversión, modulación, aislamiento, entre muchos procesos más.

#### 3.1.1. Transductores

Antes de describir las diferentes etapas en el acondicionamiento de una señal, es importante mencionar la forma de obtener una señal en su forma más simple. Un transductor es un dispositivo que convierte una forma de energía en otra distinta [19]. En los últimos años, se ha popularizado el término *sensor* o transductor de entrada, esto debido a que son capaces de “sentir” magnitudes físicas del medio y transformarlas en otro tipo de energía, generalmente eléctrica. Además de sensor, existen los *actuadores* o transductor de salida, que “actúa” cuando reciben energía, generalmente eléctrica, y la transforman en otra.

#### 3.1.2. Etapa de amplificación

En la amplificación de una señal se aumenta la intensidad de alguna magnitud física; en el ámbito eléctrico, se puede amplificar voltaje, corriente o potencia. Una manera sencilla de describir matemáticamente un amplificador es usando un bipuerto de parámetros  $g$ , cuya representación se observa en la figura 3.1.

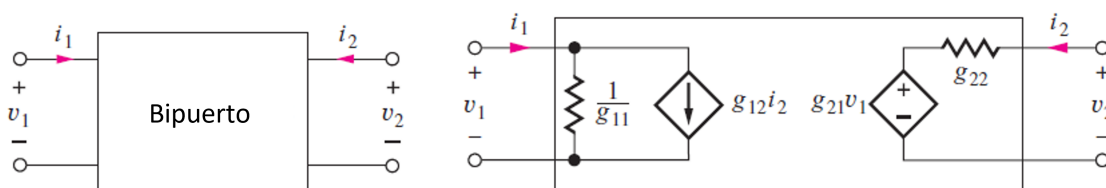


Figura 3.1: Representación de un amplificador como bipuerto.

A partir de este modelo podemos definir las siguientes ecuaciones:

$$\begin{aligned} i_1 &= g_{11}v_1 + g_{12}i_2 \\ v_2 &= g_{21}v_1 + g_{22}i_2 \end{aligned} \tag{3.1}$$

Donde los parámetros  $g$  están determinados por el análisis de corto circuito ( $i = 0$ ) y circuito

abierto ( $v = 0$ ):

$$\begin{aligned}
 g_{11} &= \left. \frac{i_1}{v_1} \right|_{i_2=0} = \text{Conductancia de entrada de circuito abierto} \\
 g_{12} &= \left. \frac{i_1}{i_2} \right|_{v_1=0} = \text{Ganancia de corriente de corto circuito} \\
 g_{21} &= \left. \frac{v_2}{v_1} \right|_{i_2=0} = \text{Ganancia de voltaje de circuito abierto} \\
 g_{22} &= \left. \frac{v_2}{i_2} \right|_{v_1=0} = \text{Resistencia de salida de corto circuito}
 \end{aligned}
 \tag{3.2}$$

Dado que estamos analizando un amplificador, se busca que la ganancia de voltaje ( $g_{21}$ ) sea mucho mayor a la ganancia de corriente ( $g_{12}$ ), por lo que se supone que ésta última tiende a cero ( $g_{12} = 0 \ll g_{21}$ ), que permite establecer el modelo más simple de un amplificador, donde  $A = g_{21}$  (Figura 3.2) [20]:

$$\begin{aligned}
 v_1 &= i_1 R_{in} \\
 v_2 &= A v_1 + i_2 R_{out}
 \end{aligned}
 \tag{3.3}$$

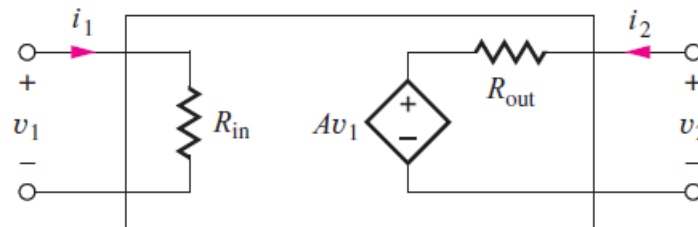


Figura 3.2: Modelo simplificado de un amplificador como un bipuerto.

El modelo del amplificador ideal se obtiene cuando a la Figura 3.2 le agregamos una fuente de voltaje y una carga y se puede observar en la figura 3.3.

Analizando la ganancia de voltaje obtenemos:

$$\begin{aligned}
 v_o &= A v_1 \frac{R_L}{R_{out} + R_L} \\
 v_1 &= v_i \frac{R_{in}}{R_I + R_{in}}
 \end{aligned}
 \tag{3.4}$$

Al obtener la ganancia de voltaje del circuito combinando las ecuaciones (3.4) :

$$|A_v| = \frac{v_o}{v_i} = A \frac{R_L}{R_{out} + R_L} \frac{R_{in}}{R_I + R_{in}}
 \tag{3.5}$$

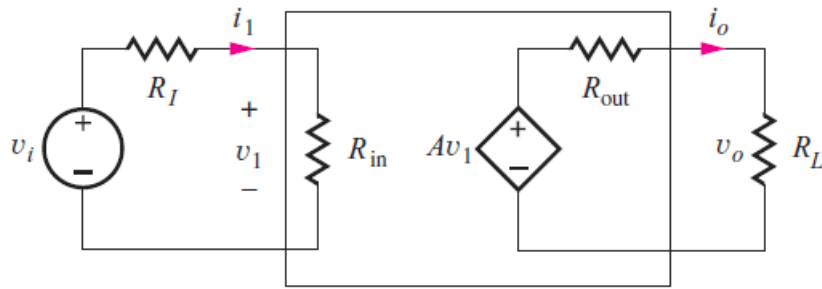


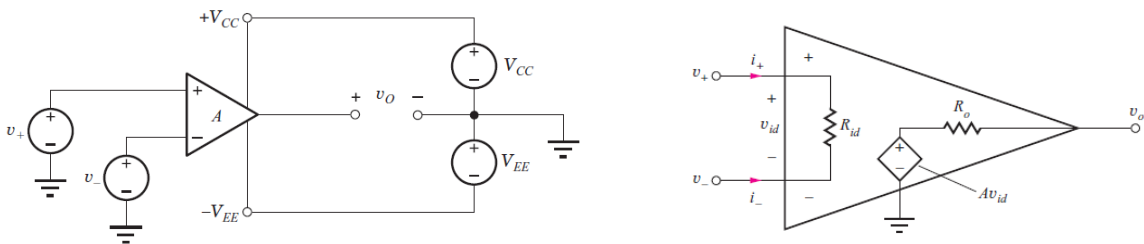
Figura 3.3: Representación de un amplificador con fuente y carga.

Donde, suponiendo que la  $R_{in} \gg R_I$  y  $R_L \gg R_{out}$  obtenemos que  $|A_v| \approx A$ . En el amplificador ideal se asume que  $R_{in} = \infty$  y  $R_{out} = 0$  para lograr una máxima transferencia de potencia.

### El amplificador operacional

El amplificador operacional (AO) u *opamp* (por su nombre del inglés) es un dispositivo electrónico capaz de realizar diversas operaciones matemáticas como suma, multiplicación por un escalar, diferencia e integración de señales, además de que permite acoplar sistemas eléctricos, todo esto en el dominio analógico. Las primeras implementaciones de un AO como circuito integrado se hicieron en la década de los 60's, ofreciendo ventajas en cuanto a costo, tamaño y potencia [20]. Las aplicaciones de un amplificador operacional generalmente están dirigidas a sistemas de control donde se desea tener una retroalimentación o para filtros activos.

El AO es una forma de un amplificador diferencial que, como su nombre lo indica, su salida está determinada por la diferencia entre dos señales de entrada. Las fuentes de poder positiva ( $V_{CC}$ ) y negativa ( $V_{EE}$ ) polarizan al dispositivo y limitan la salida que se puede tener de la forma  $-V_{EE} \leq 0 \leq V_{CC}$ . Las entradas del dispositivo se llaman entrada inversora ( $v_-$ ) y entrada no inversora ( $v_+$ ).



(a) Amplificador diferencial con fuentes de poder.

(b) Modelo de un amplificador diferencial.

Figura 3.4: Modelos de un amplificador diferencial, tomados de [20].

Existen dos regiones de interés cuando se trabaja con un amplificador diferencial. Cuando se ingresa un voltaje a través de las entradas del amplificador, se desea trabajar en la

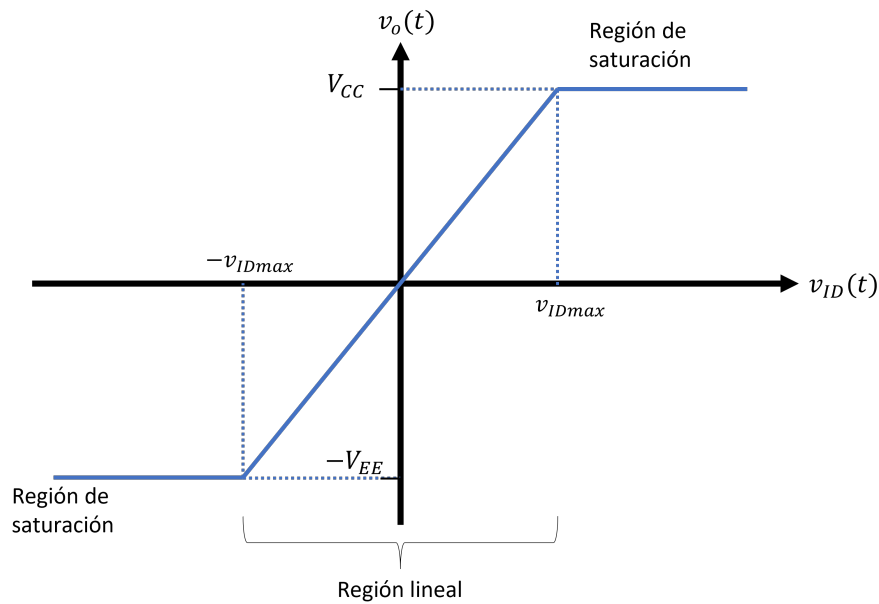


Figura 3.5: Función de transferencia ideal de un amplificador operacional.

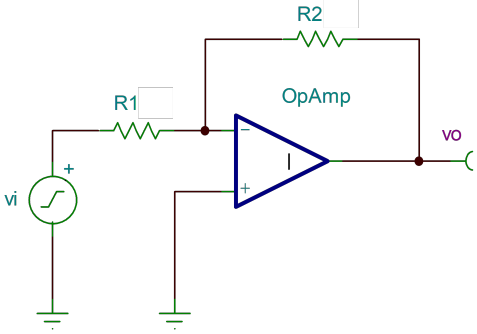
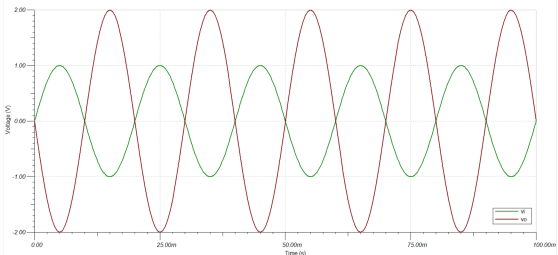
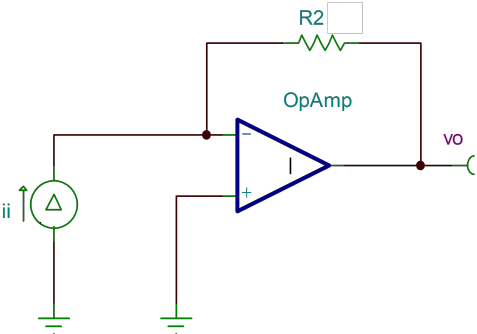
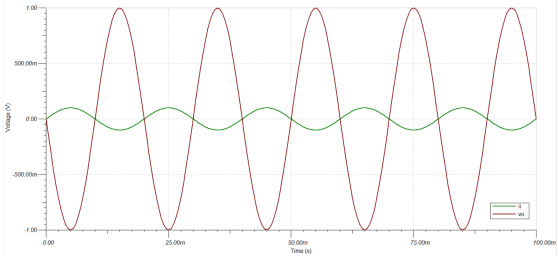
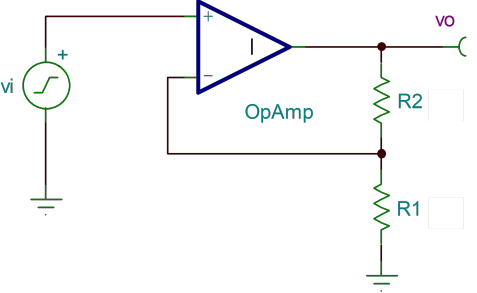
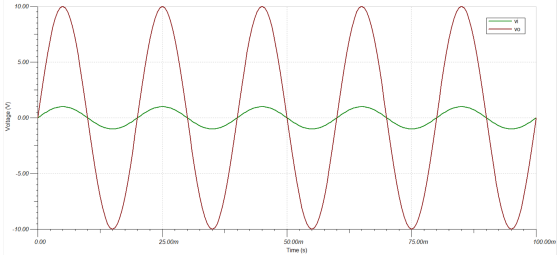
“Región lineal”, donde dicho voltaje se verá amplificado a la salida. Sin embargo, en la “Región de saturación” no ocurrirá dicho proceso puesto que se ha alcanzado el límite descrito anteriormente por las fuentes de poder. La ecuación que define el comportamiento de un amplificador diferencial es la siguiente:

$$v_O = A(v_+ - v_-) \quad (3.6)$$

El amplificador operacional ideal permite tener una idea aproximada del comportamiento de éste dispositivo en un circuito real. Las suposiciones son las siguientes:

- El voltaje a la entrada es igual a cero ( $v_{id} = v_+ - v_- = 0$ ).
- Las corrientes en la entrada inversora y no inversora son igual a cero.
- El CMRR es infinito (Factor de Rechazo al Modo Común).
- El PSRR es infinito (Factor de Rechazo a Fuente de Alimentación).
- Salida de voltaje infinita (No limitada por las fuentes de poder  $V_{CC}$  y  $V_{EE}$ ).
- Capacidad de brindar corriente ilimitada a la salida.
- Ancho de banda de lazo abierto infinita.
- Slew Rate (SR) infinito.
- Resistencia de salida igual a cero.
- Corrientes de polarización y de offset igual a cero.
- Voltaje de offset de entrada igual a cero.

En la tabla 3.1 se presetan las configuraciones más usadas en amplificadores operacionales, tomando como base un amplificador ideal. En un circuito real es necesario tomar en cuenta otros efectos y parámetros para tener un resultado más aproximado.

Configuración	Señal de salida
<p style="text-align: center;">Amplificador inversor</p> 	<p style="text-align: center;">Ganancia de voltaje <math>A_v = -\frac{R_2}{R_1}</math></p> 
<p style="text-align: center;">Amplificador de transresistencia</p> 	<p style="text-align: center;">Ganancia de transconductancia <math>A_{tr} = -\frac{v_o}{i_i} = -R_2</math></p> 
<p style="text-align: center;">Amplificador no inversor</p> 	<p style="text-align: center;">Ganancia de voltaje <math>A_v = 1 + \frac{R_2}{R_1}</math></p> 

Continúa en la página siguiente.

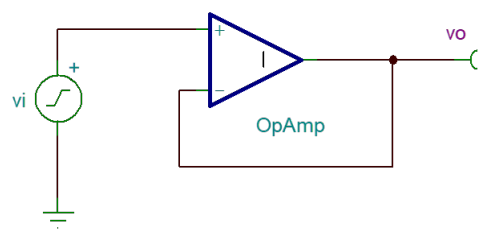
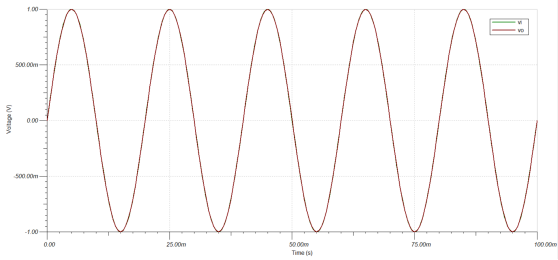
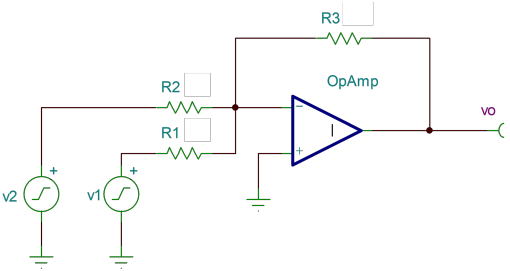
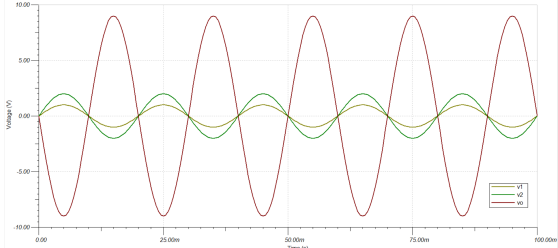
Configuración	Señal de salida
<p style="text-align: center;">Seguidor de voltaje</p> 	<p style="text-align: center;">Ganancia de voltaje <math>A_v = 1</math></p> 
<p style="text-align: center;">Amplificador sumador</p> 	<p style="text-align: center;"><math>v_O = -\left(\frac{R_3}{R_1}v_1 + \frac{R_3}{R_2}v_2\right)</math></p> 

Tabla 3.1: Configuraciones comunes con un amplificador operacional.

### 3.1.3. Etapa de offset

Cuando hablamos de una señal de corriente alterna periódica continua en el tiempo, su voltaje en DC de Offset, coeficiente de DC o también llamado componente de corriente continua, es la media o el promedio de la amplitud de la forma de onda de la señal [21]. Generalmente para cualquier tipo de señal senoidal, el valor de la media de su amplitud es cero cuando no existe una polarización en DC de la señal. A este tipo de señales se les conoce como formas de onda balanceadas en DC.

A pesar de que la compensación en DC para señales de AC no es muy conveniente, sobre todo cuando trabajamos con amplificadores (debido a que podría causar recortes en las zonas de mayor amplitud en la señal u otros cambios no deseados en el punto de operación del amplificador), existen diversas aplicaciones donde la suma de un componente de corriente continua es importante. En el campo de procesamiento de audio, podemos obtener compensaciones en DC indeseables durante la captura del sonido causado normalmente por equipos de baja calidad, causando recortes en la señal en la etapa de amplificación, o agregando distorsión en el intervalo de bajas frecuencias. Es aquí donde manualmente, se desea compensar la señal en DC, desplazando la forma de onda tal que su componente en DC sea cero de nuevo.

Otra de las aplicaciones más frecuentes de la compensación en DC se encuentra en desplazar una señal alterna sumando un componente en DC, de tal manera que los valores de la señal resultante sean únicamente positivos. Esto debido a que muchos convertidores ADC sólo operan en intervalos de voltaje positivos, mientras que en una señal alterna se abarcan valores tanto positivos como negativos.

La forma más común y más fácil de sumar un componente en DC a una señal alterna es agregando un divisor de voltaje en la etapa final de nuestro circuito, tal como se muestra en la figura 3.6.

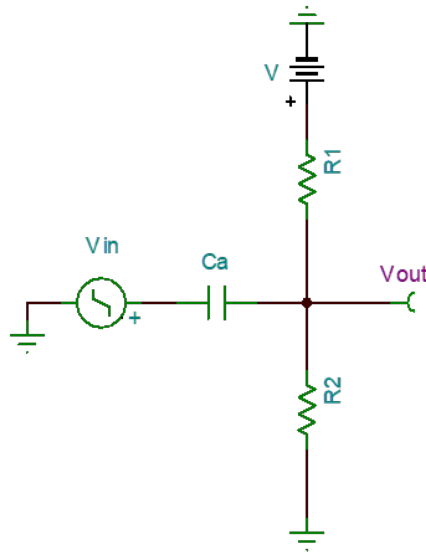


Figura 3.6: Circuito de Offset a la salida de la amplificación.

De aquí podemos observar que el divisor de corriente está conformado por las resistencias  $R_1$  y  $R_2$ , así como por el voltaje de polarización  $V^+$ . También notamos que se agrega un capacitor a la entrada del divisor que nos ayuda a acoplar nuestra señal de AC. Suponiendo que nuestra fuente de voltaje de entrada  $V_{in}$  posea una impedancia de entrada  $Z_{in}$  tal que  $Z_{in} \rightarrow \infty$  y recordando que el capacitor en corriente directa se puede representar como un circuito abierto, podemos empezar con el análisis en DC del circuito equivalente que se muestra en la figura 3.7.

De tal manera que el voltaje en DC a la salida de nuestro divisor está dado por las siguientes ecuaciones.

$$V = V_{R1} + V_{R2} = I(R_1 + R_2) \quad (3.7)$$

$$I = \frac{V}{R_1 + R_2} \quad (3.8)$$

$$V_{OUT} = V_{R2} = IR_2 \quad (3.9)$$

$$V_{OUT} = \frac{VR_2}{R_1 + R_2} \quad (3.10)$$



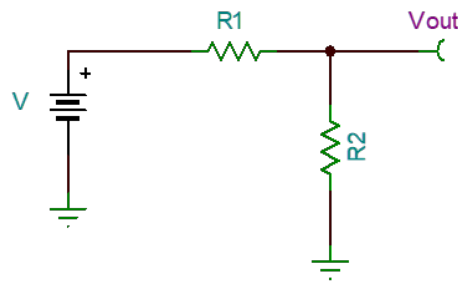


Figura 3.7: Circuito divisor de Voltaje como equivalente de Circuito de Offset en DC.

Así mismo, la suma total de nuestros componentes en DC y AC estaría expresada de la siguiente manera:

$$v_{OUT} = V_{OUT} + v_{in} \quad (3.11)$$

Cabe destacar que el capacitor de acoplamiento  $C_a$  en el análisis en AC, como se muestra en la figura 3.8, nos da como resultado un filtro paso altas que nos ayuda a eliminar componentes en DC de etapas anteriores. La frecuencia de corte de este filtro está dada por las siguientes ecuaciones:

$$f_c = \frac{1}{2\pi * CR_{eq}} = \frac{1}{2\pi * C(R_1 || R_2)} \quad (3.12)$$

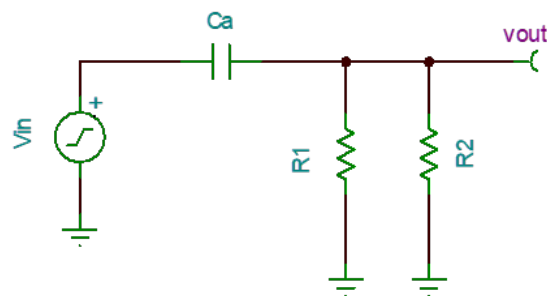


Figura 3.8: Circuito paso Altas como equivalente de Circuito de Offset en AC.

## 3.2. Filtros digitales

Los filtros son sistemas que nos permiten seleccionar cierto intervalo de frecuencias y dejan pasar todas aquellas señales dentro de este intervalo, desde la entrada hasta la salida del sistema, mientras que bloquean todas las demás señales fuera del mismo intervalo [6]. En el procesamiento digital de señales, los filtros son utilizados en diversas aplicaciones tales como separar señales mezcladas, remover ruido indeseable, hacer compensaciones en bandas de frecuencia, restaurar señales distorsionadas, entre muchas otras [22].

A diferencia de los filtros analógicos, los cuales en el diseño de manera pasiva se componen de diversas combinaciones de resistores, capacitores e inductores y de manera activa utilizan componentes pasivos combinados con amplificadores, los filtros digitales se implementan con una combinación de hardware y software y trabajan sólo con señales digitales. Asimismo, los filtros digitales son mucho más flexibles que los analógicos, debido a que al cambiar los coeficientes del filtro en el software podemos programarlos para tener una respuesta en frecuencia diferente sin tener que modificar el hardware, lo cual los hace además de flexibles, una alternativa más económica.

La clasificación más común de los Filtros Digitales Lineales se compone de los Filtros de Respuesta Finita al Impulso (FIR), los Filtros de Respuesta Infinita al Impulso (IIR) y los filtros Lattice [22].

### 3.2.1. Filtros FIR

Podemos ver a los filtros de tipo FIR como la convolución entre la entrada  $x(n)$  y la respuesta al impulso del filtro  $h(n)$ , por lo que la señal de salida  $y(n)$  se representa como:

$$y(n) = \sum_{i=0}^{N-1} h(i)x(n-i) \quad (3.13)$$

Si obtenemos la transformada Z de esta respuesta obtenemos entonces:

$$Y(z) = X(z) \sum_{i=0}^{N-1} h(i)z^{-i} \quad (3.14)$$

Y la función de transferencia está dada por  $H(Z) = \frac{Y(z)}{X(z)}$ , y además si queremos observar la ecuación 3.14 como la respuesta del filtro en el dominio de la frecuencia donde  $H(e^{j\omega}) = H(z)|_{z=e^{j\omega}}$  entonces obtenemos:

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n)z^{-e^{j\omega}} \quad (3.15)$$

Donde la respuesta  $H(e^{j\omega})$  representa la transformada de Fourier en tiempo discreto de  $h(n)$  tal que dicha señal es periódica en  $2\pi$ . La fase afecta directamente al retardo de las diferentes componentes en frecuencia que pasan a través del filtro donde existe un retraso ( $\alpha$ ) en el número de muestras en la secuencia al momento de procesarla.

### Diseño por medio de ventanas

En el proceso de diseño de este tipo de filtros se necesitan considerar los siguientes puntos [22]:

- Se selecciona la respuesta en frecuencia deseada  $H(e^{j\omega})$  en consideración de las especificaciones de nuestro filtro.

- Se calcula la respuesta al impulso para la función utilizando fórmulas de diseño.
- Como dicha respuesta al impulso de la función de transferencia deseada es una respuesta infinita es necesario truncarla en  $N$  puntos multiplicando dicha respuesta por una ventana de longitud  $N$  logrando una mejor respuesta espectral y obteniendo un filtro práctico.
- Como la  $h(n)$  obtenida no es causal ( $NC$ ) entonces se es necesario desplazarla a la derecha  $\alpha$  puntos para obtener una versión causal (como se mencionó anteriormente en las consideraciones de retardo para este tipo de filtros).
- Como el efecto de recortar  $h(n)$  ocasiona una respuesta en frecuencia diferente a la ideal, se grafica la  $|H(e^{j\omega})|$  y se compara con los requerimientos especificados. Si no cumple con dichos requerimientos, se incrementa el orden  $N$  o se cambia la ventana y se recalculan los coeficientes.

### Ventana de Hamming

Esta ventana es de las más utilizadas en aplicaciones prácticas, ya que sus características principales es que sus primeros lóbulos laterales se encuentran  $-41dB$  debajo del lóbulo principal, con pendiente de  $-6dB$  por octava, sin embargo, a pesar de obtener menor numero de oscilaciones en las regiones rechaza banda su banda de transición se vuelve menos abrupta y más ancha. La ecuación que la representa es:

$$W(n) = \begin{cases} 0.54 - 0.46 \cos \frac{2\pi n}{N-1} & ; \quad 0 \leq n \leq N-1 \\ 0 & ; \quad \text{para otra } n \end{cases} \quad (3.16)$$

### 3.2.2. Filtros IIR

Si un filtro IIR se quisiera representar de igual manera que un filtro tipo FIR como la convolución de la entrada con su respuesta al impulso (ecuación 3.13), debido a que en un filtro IIR esta respuesta es infinita, esta operación no sería realizable computacionalmente, por lo que es necesario representarlo como una ecuación en diferencias que retroalimente la salida retardada, es decir:

$$y(n) = \sum_{i=0}^q b(i)x(n-i) - \sum_{i=1}^p a(i)y(n-i) \quad (3.17)$$

Si a la ecuación 3.17 le aplicamos la transformada Z y despejamos  $H(z)$  obtenemos la función de transferencia del filtro dada por:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{B(z)}{A(z)} = \frac{\sum_{i=0}^q b(i)z^{-i}}{\sum_{i=0}^p a(i)z^{-i}} = \frac{b_0 + b_1z^{-1} + \dots + b_qz^{-q}}{a_0 + a_1z^{-1} + \dots + a_pz^{-p}} \quad (3.18)$$

Dónde  $p$  es el número de polos y  $q$  es el número de ceros de  $H(z)$  y  $a_i$  y  $b_i$  son los coeficientes del filtro. Toda estructura que se seleccione debe ser equivalente al mismo sistema.

### Diseño por prototipos analógicos o por aproximaciones polinómicas

Los prototipos analógicos existentes son funciones de transferencia normalizadas  $H(s)$  y nos dan una aproximación de una función ideal  $H(\Omega)$ . Las aproximaciones que más se utilizan en el diseño de filtros analógicos son las siguientes [22]:

- Butterworth
- Chebyshev tipo I y II
- Elíptico o de Cauer
- Bessel

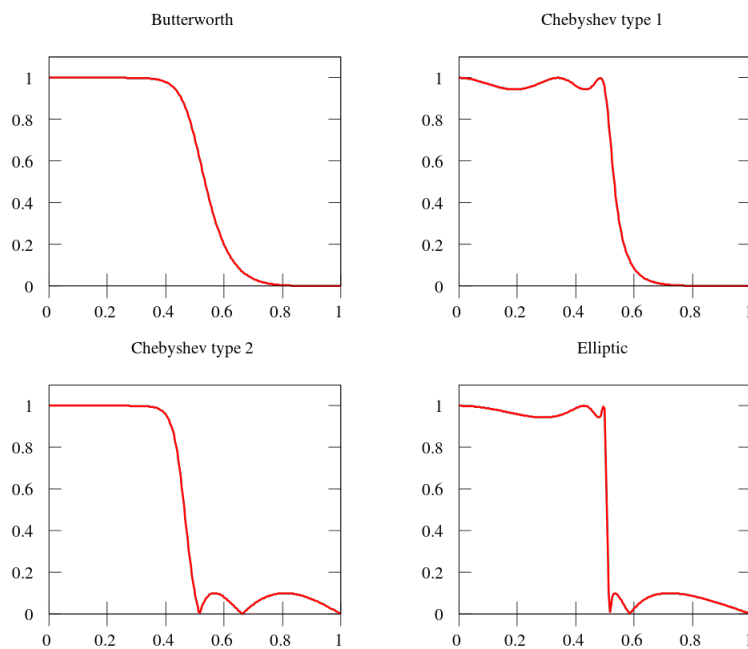


Figura 3.9: Gráfica de respuesta a diferentes tipos de filtros Lineales, tomado de [23].

### 3.3. Módulo Bluetooth

Bluetooth es un estándar de uso libre desarrollado por la asociación *Bluetooth Special Interest Group* (SIG), que permite el intercambio de información entre dispositivos electrónicos sin la necesidad de cables. Entre las características más importantes de la conectividad Bluetooth podemos encontrar:

- Opera en la banda ISM (industrial, científica y médica) de 2.4GHz a 2.485 GHz [24].
- Cuenta con el *Salto de frecuencia adaptable (AHF)* que le permite tener una menor interferencia de otros dispositivos que ocupan la misma banda de frecuencia.
- Permite que la comunicación pueda ser full-dúplex.
- Alcance aproximado de diez metros.

- Latencia de 200ms.

La principal ventaja de incorporar este estándar a dispositivos electrónicos es su bajo costo de producción y poco consumo de potencia. En la actualidad, se puede ver al Bluetooth incorporado en aplicaciones industriales de corto alcance y en aparatos de uso personal como computadoras o teléfonos inteligentes.

El módulo Bluetooth HC05—incorpora éste estándar en su versión V2.0+EDR(Velocidad de datos mejorada), siendo capaz de enviar y recibir información a una velocidad de hasta 3Mbps. En la tabla 3.2 se muestran algunas características técnicas de éste módulo:

Tabla 3.2: Características del módulo Bluetooth HC05

Hardware	Software
<ul style="list-style-type: none"> <li>- Sensibilidad de -80dbm.</li> <li>- Hasta +4dbm de potencia de transmisión</li> <li>- Dos modos de operación con salidas de 1.8V y 3.6V.</li> <li>- Utiliza el protocolo de comunicación serial SCI.</li> <li>- Antena integrada.</li> <li>- LEDs indicadores de estado.</li> </ul>	<ul style="list-style-type: none"> <li>- Tasa de bauds programable con hasta ocho posibles valores.</li> <li>- Instrucciones AT capaces de cambiar parámetros del módulo.</li> <li>- Auto emparejamiento y detección del último dispositivo conectado.</li> </ul>

Se pueden definir dos modos de operación del módulo HC05, el *primero* es el modo de operación normal donde, al energizar el módulo a través de los pines *VCC* y *GND*, se permite el intercambio de información con cualquier otro dispositivo. En este estado, el módulo es visible para dispositivos que tengan incorporado el Bluetooth como, por ejemplo, teléfonos móviles donde la información es codificada y enviada a través del pin *RXD*. El *segundo* modo de operación es para poder editar parámetros de la conexión mediante comandos AT; para entrar en este modo, es necesario energizar el módulo e ingresar un voltaje de mayor a 3.3V en el pin *EN*. Los comandos son recibidos por el pin *RXD* y la respuesta, en caso de ser exitosa, es enviada por *TXD*, como se puede ver en la imagen 3.10.

### 3.3.1. Comandos AT

Los comandos AT son una serie de cadenas de texto que juntas generan un comando específico para realizar ciertas acciones. Es particularmente usado por módems, para cambiar parámetros de la conexión.

En el caso del módulo HC05, los comandos AT permiten cambiar la dirección, el nombre con el que puede ser visto por otros dispositivos, la tasa de bauds, la contraseña de emparejamiento, entre otras cosas.

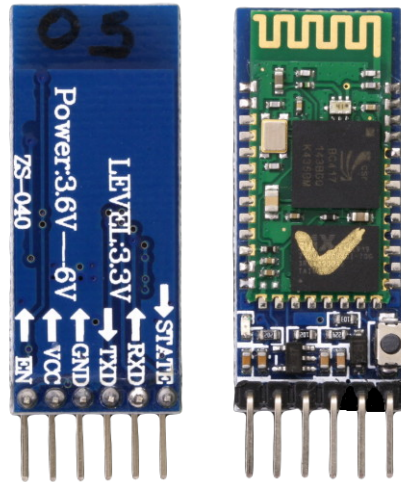


Figura 3.10: Módulo HC-05.

Comando	Respuesta	Descripción
AT	OK	Cuando se ingresa este comando al módulo, éste contesta con un “OK”, para indicar que la conexión fue exitosa, de lo contrario regresará “ERROR”.
AT+RESET	OK	Reinicia el dispositivo y entra en su operación normal.
AT+VERSION?	Versión OK	Cuando se usa “?”, el módulo responde con la versión almacenada en memoria Cuando se usa “=”, permite ingresar la versión manualmente y responde “OK”.
AT+ORLG	OK	Permite establecer los valores por defecto del módulo

(Continúa en la página siguiente.)

Comando	Respuesta	Descripción
AT+ADDR?	Dirección OK	Cuando se usa "?", el módulo responde con la dirección Bluetooth (la forma es determinada por el mismo estándar) Cuando se usa "=", permite ingresar la dirección manualmente y responde "OK".
AT+NAME?	Nombre del módulo OK	Cuando se usa "?", el módulo responde con el nombre del mismo, por defecto es "H-C-2010-06-01" y éste parámetro es el que es visible para los demás dispositivos Cuando se usa "=", permite ingresar el nombre manualmente y responde "OK".
AT+PSWD?	Contraseña OK	Cuando se usa "?", el módulo responde con la contraseña para poder emparejarse con otro dispositivo Cuando se usa "=", permite ingresar la contraseña manualmente y responde "OK".
AT+UART?	Parámetros de la conexión serial UART OK	Cuando se usa "?", el módulo responde con las especificaciones de la conexión serial, éstas son: Tasa de bauds, bit de paro y bit de paridad. Las posibles tasas de bauds son: 9600, 19200, 38400, 57600, 115200, 230400, 460800. Cuando se usa "=", permite ingresar los parámetros de la conexión serial y responde con "OK"

Tabla 3.3: Ejemplos de los comandos AT más comunes con sus posibles respuestas

### 3.4. Sensores de teléfonos inteligentes: Acelerómetro, giroscopio y magnetómetro

Los teléfonos inteligentes tienen múltiples funciones que facilitan la interacción con el usuario, una de estas funciones es la orientación; ésta nos permite saber, por ejemplo, cuando el celular está en una posición vertical u horizontal para desplegar más o menos información en pantalla. Su utilidad no sólo se reduce a esto, sino también a aplicaciones más complejas como un medidor de pasos o sistemas de navegación, etc [25].

Para poder determinar la orientación de manera precisa y rápida, se recolectan datos de tres diferentes sensores: el acelerómetro, el giroscopio y magnetómetro.

#### 3.4.1. Acelerómetro

Un acelerómetro es un dispositivo capaz de medir la aceleración, es decir, la tasa de cambio de velocidad en un determinado instante de tiempo. Por ejemplo, cuando un acelerómetro se deja en reposo en una superficie plana como una mesa, se tendrá un valor de  $9.81 \frac{m}{s}$  o  $1g$  debido a la fuerza de gravitación de la Tierra. Existen diseños simples como un péndulo[26] o sistemas más sofisticados como acelerómetros piezoeléctricos, microelectromecánico (MEMS), capacitivos, etc.

Los acelerómetros MEMS son particularmente importantes debido al gran uso que tienen en dispositivos móviles como teléfonos inteligentes. En términos generales, esta tecnología está compuesta de masa estándar y elementos elásticos como vigas voladizas. La medición de la aceleración se logra de dos formas, ambas incluyen el movimiento de la masa antes mencionada: La *primera* es midiendo la capacitancia que se forma entre la masa y un electrodo auxiliar, y la *segunda* es mediante un transductor de presión. Para garantizar una alta sensibilidad y un bajo nivel de ruido, es necesario que la masa sea lo suficientemente grande; la producción a gran escala se ve beneficiada por esto [27].

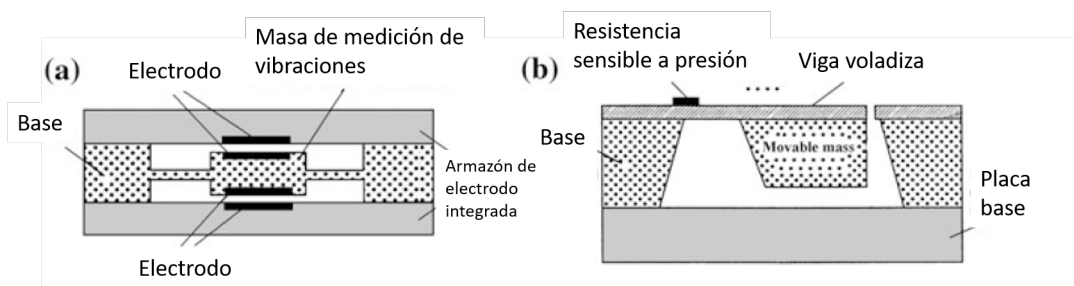


Figura 3.11: Corte lateral del diseño de un acelerómetro con tecnología MEMS, tomado de [27].

Los teléfonos inteligentes cuentan con un acelerómetro capaz de medir la aceleración en tres coordenadas: X, Y y Z. De esta manera, cada una de las “caras” del teléfono tiene asignada una coordenada, como se puede apreciar en la imagen 3.12. Retomando el ejemplo inicial, si un celular está en reposo sobre una superficie plana, el sensor Z tendrá una lectura de  $1g$  positiva.



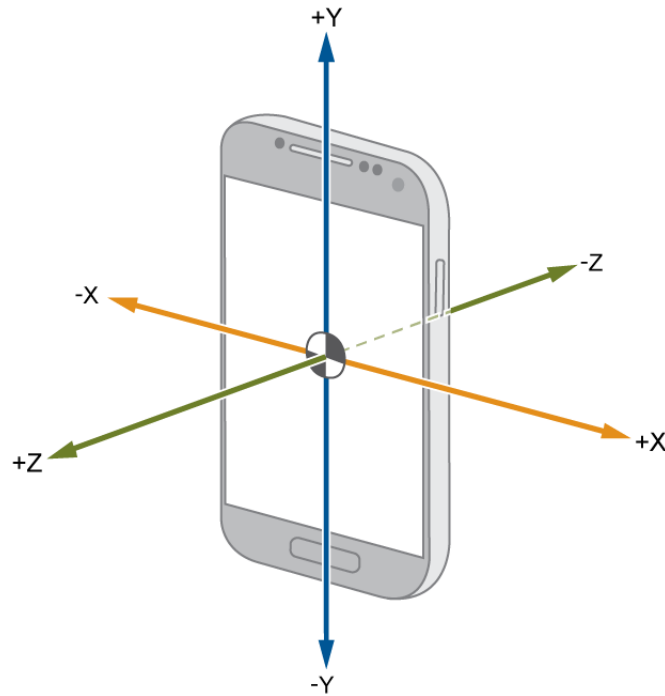


Figura 3.12: Representación del sistema de coordenadas de un acelerómetro en un teléfono inteligente, tomado de [28].

### 3.4.2. Giroscopio

Un giroscopio es un dispositivo capaz de medir la velocidad angular sobre un eje fijo del espacio. Su nombre fue otorgado por el físico francés Léon Foucault en el siglo XIX. Forma parte de los sensores inerciales dentro de la categoría de medidores de velocidad angular (la otra clasificación es velocidad lineal). La principal ventaja y la razón por la que se usan en dispositivos electrónicos es por que no se ven afectados por campos electromagnéticos externos ni son tan susceptibles a aceleraciones externas, como es el caso de los magnetómetros y acelerómetros, respectivamente [25].

Las primeras aplicaciones de los giroscopios fueron para propósitos militares y aeroespaciales, por ejemplo, para estabilización de aeronaves o misiles[29]. Actualmente, se usan en una gran variedad de ramas que van desde medicina, robótica, automotriz y electrónicos de uso personal.

Existen tres tipos de giroscopios: los de masa giratoria, los ópticos y los vibratorios. El primer tipo de giroscopios cuentan, como su nombre lo indica, cuentan con una masa que gira constantemente respecto a un eje móvil. En la imagen 3.13 se puede observar este tipo de dispositivos que también se les conoce como los DTG (Giroscopio de Ajuste Dinámico). Los giroscopios ópticos están basados en el efecto Sagnac que postula que el cambio de fase de dos señales contra-propagándose en un interferómetro de anillo rotatorio es proporcional a la velocidad angular en la trayectoria dentro del mismo.

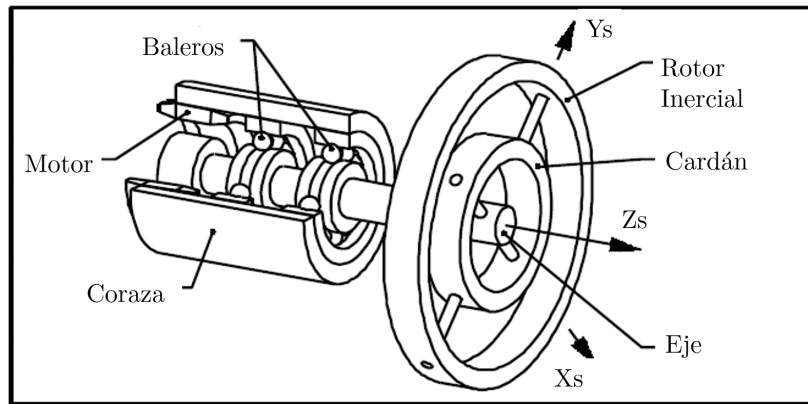
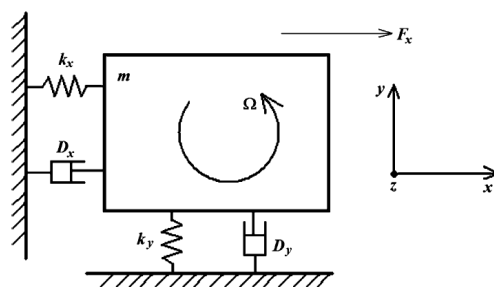


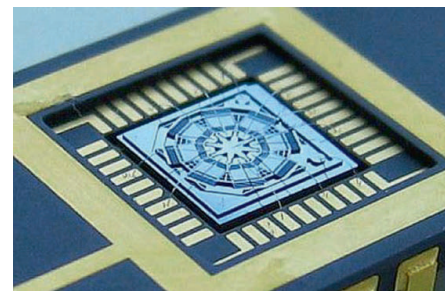
Figura 3.13: Modelo de un Giroscopio de Ajuste Dinámico, DTG, tomado de [29]

Finalmente, los giroscopios vibratorios están basados en el efecto Coriolis que establece que un objeto en movimiento dentro de un sistema de referencia en rotación, experimentará una aceleración relativa. La primera demostración de este efecto se dio en 1851 por Foucault; Un péndulo presentaba una desviación aparente en sentido antihorario (derecha a izquierda) debido a estar en el hemisferio norte y por la rotación de la Tierra [30].

Sin embargo, implementar un sistema rotatorio capaz de medir la aceleración de un objeto resulta complicado por el movimiento en sí, además de otros factores como la fricción y el desgaste, por lo que se diseñaron estructuras vibratorias que no presentan estos inconvenientes. En términos generales, se tiene una masa de prueba vibratoria suspendida por vigas flexibles que en conjunto forman un sistema masa-resorte-amortiguador o, en otras palabras, un sistema de segundo orden. El concepto de los giroscopios vibratorios se puede observar en la imagen 3.14.



(a) Esquema del sistema masa-resorte-amortiguador del giroscopio.



(b) Vista del chip de un giroscopio.

Figura 3.14: Giroscopio MEMS, tomado de [31].

### 3.4.3. Magnetómetro

Un magnetómetro mide la fuerza de los campos magnéticos alrededor del dispositivo. Estando aislado de campos de gran intensidad, estos sensores se utilizan en sistemas de navegación

para determinar el polo norte magnético [32], lo que permite el desarrollo de aplicaciones como brújulas digitales.

Existen tres tipos de magnetómetros MEMS utilizados comúnmente por dispositivos como teléfonos inteligentes: el primero son los magnetómetros magneto-resistivos ordinarios (MR) y anisótropos (ARM) que utilizan la propiedad de ciertos materiales de variar su resistencia eléctrica estando en presencia de un campo magnético. Este efecto fue descubierto en 1857 por Lord Kelvin, pero no fue sino hasta la década de 1960 que se comenzaron a desarrollar películas magnéticas capaces de otorgar una lectura de alta resolución [33].

El segundo tipo son los magnetómetros de efecto Hall que establece que cuando en un material conductor o semiconductor por el cuál circula una corriente, estando en presencia de un campo magnético, los portadores de carga se reagruparán en el material generando un campo eléctrico perpendicular, así como una diferencia de potencial entre los extremos de la placa.

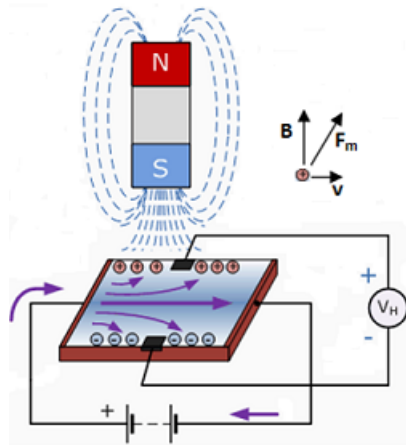


Figura 3.15: Representación del efecto Hall, tomado de [34].

El tercer tipo de magnetómetros MEMS son los que utilizan la fuerza de Lorentz. Esta fuerza esta descrita como la que experimenta una partícula de carga  $q$  cuando es sometida a un campo electromagnético, descrito matemáticamente como:

$$\vec{F} = q(\vec{E} + \vec{v} \times \vec{B})$$

En estructuras basadas en silicio, cuando un campo magnético externo entra en contacto con el material y se hace circular una corriente eléctrica, se experimentará una fuerza de Lorentz que es perpendicular a ambos vectores y que deforma el material, pudiendo ser detectado por técnicas capacitivas, piezoeléctricas u ópticas. Para aumentar la sensibilidad, el sistema es generalmente usado en resonancia.

### 3.5. Aplicación Móvil

Un teléfono inteligente se define como un dispositivo electrónico de uso personal que permite incorporar las funciones de un teléfono convencional y una computadora a menor escala. En

otras palabras, se pueden realizar llamadas o mensajes, y además la posibilidad de navegar en la WEB, reproducir archivos multimedia, edición de textos, etc [35]. De acuerdo con [36], la cantidad de usuarios de teléfonos inteligentes en todo el mundo al 2021 son 3.8 mil millones, que comprende casi el 50 % de la población, con una tendencia a seguir incrementando.

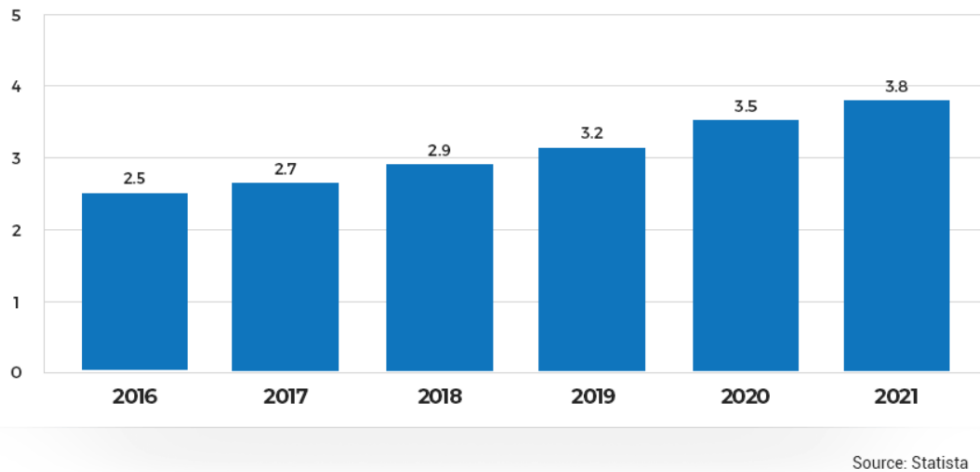


Figura 3.16: Total de usuarios de teléfonos inteligentes, en miles de millones.

El *software* de un teléfono inteligente se compone, principalmente, del sistema operativo (SO) y las aplicaciones que éste pueda incluir. Actualmente, existe una cantidad considerable de SO para smartphones, pero la realidad es que este mercado está dominado por Android de Google y IOS de Apple.

En cuanto al *hardware* de estos dispositivos, podemos identificar dos partes: el Sistema en Chip (SoC) y los sensores. El SoC es considerado el cerebro, ya que cuenta con varios módulos que controlan las funciones de los teléfonos inteligentes; se pueden mencionar las siguientes partes:

- Unidad central de procesamiento (CPU). Su trabajo es ejecutar las instrucciones del sistema y asignar las tareas a cada uno de los módulos.
- Unidad de procesamiento de gráficos (GPU). Se encarga de manejar las tareas relacionadas a la visualización de gráficos en 2D y 3D.
- Unidad de procesamiento de imágenes (ISP). Procesa la información proveniente de las cámaras del celular en archivos que puedan ser entendidos por los demás módulos.
- Procesador Digital de Señales (DSP). Es el encargado de procesar señales en tiempo real que requieren métodos matemáticos más avanzados.
- Unidad de procesamiento neuronal (NPU). Utilizado en sistemas de gama alta para tareas de inteligencia artificial.
- Codificador y decodificador de video. Es el encargado de manejar los formatos de video.

- Módems. Se encarga de traducir las señales inalámbricas en información que pueda ser entendida por los demás módulos que la requieran. Se pueden encontrar módems de 4G, 5G, WI-FI y Bluetooth.

La otra parte se compone de los sensores y actuadores del sistema, como lo pueden ser cámaras, giroscopios, acelerómetros, sensores de luz, etc. Hablando específicamente de los sensores, Android [37] los clasifica en tres ramas: Sensores de movimiento, sensores de posición y sensores de entorno. Retomando el tema 3.4, el acelerómetro y el giroscopio entran dentro de la categoría de sensores de movimiento y el magnetómetro en la de entorno.

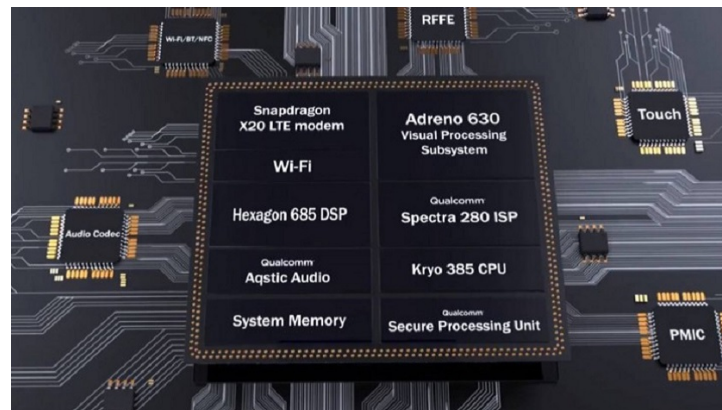


Figura 3.17: Ejemplo de un SoC de un teléfono inteligente, el Snapdragon 845, tomado de [38].

### 3.5.1. Android

Android es un Sistema Operativo de código libre para dispositivos móviles basado en Linux. Se encuentra principalmente en teléfonos inteligentes y tabletas, pero esto ha cambiado con la inclusión en relojes y bandas inteligentes, televisores y hasta automóviles [39].

La principal ventaja de Android es que es gratuito y permite modificar el código fuente para generar capas de personalización de acuerdo a las marcas que así lo deseen. Al año 2020 existen once versiones de este SO, siendo cada una más eficiente, segura y sencilla de comprender para el usuario que la versión anterior.

Finalmente, las aplicaciones se pueden desarrollar utilizando un entorno de desarrollo integrado (IDE) proporcionado por Google llamado “Android Studio” que utiliza una variación del lenguaje de programación Java con funciones y bibliotecas para manejar los dispositivos móviles, como sensores, cámaras, etc.

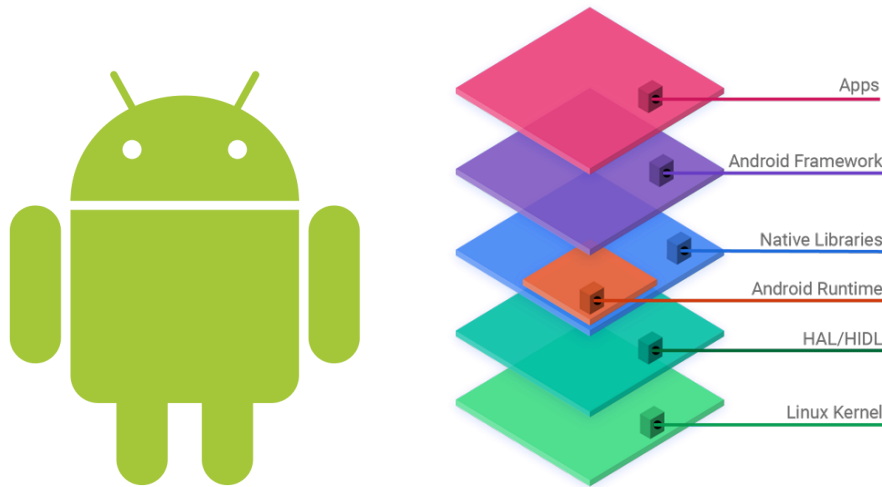


Figura 3.18: Logo de Android y su composición, tomado de [39].

### 3.5.2. MIT App Inventor

MIT App Inventor es un herramienta en la nube capaz de crear aplicaciones para sistemas operativos Android y IOS [40], es desarrollado por el MIT. Su principal característica radica en el uso de bloques para realzar diversas instrucciones; por ejemplo, para el uso de una instrucción condicional se tiene un bloque *if else* en donde de un lado se añaden las instrucciones en caso de que la condición se cumpla y en el otro en caso de que sea falsa.

Entre otras cosas, permite publicar los proyectos creados en las distintas tiendas de aplicaciones para que puedan ser compartidas e instaladas en tantos dispositivos como se desee.



Figura 3.19: MIT App Inventor, tomado de [40].

### Pruebas y emulación

Para realizar la prueba de las aplicaciones, se puede hacer uso de tres métodos: El primero permite conectarse inalámbricamente con un dispositivo compatible (Con SO Android o IOS) utilizando una aplicación; el segundo método conecta mediante un cable USB con el dispositivo y el tercero permite realizar la emulación del sistema operativo en la computadora.

## Creación de Apps

La creación de aplicaciones para Android se lleva a cabo mediante el uso de dos pantallas: la pantalla de Blueprint y la pantalla de código (O bloques, en el caso de MIT App Inventor).

La pantalla de Blueprint contiene de manera gráfica los diversos elementos que serán mostrados en pantalla. Existen dos tipos de elementos en esta sección: los elementos visibles y los elementos no visibles. Los elementos visibles son aquellos que se muestran en pantalla y la mayoría de los casos permite interactuar con el usuario; los segundos son los que no se muestran en pantalla pero permite inicializar módulos como el Bluetooth. A continuación se muestra una lista con una breve descripción de cada una de las categorías:

- **User Interface.** Contiene elementos básicos que se pueden mostrar en pantalla como cuadros de texto, botones, listas desplegables, interruptores, entre otras cosas.
- **Layout.** Permite agrupar elementos en bloques contenedores que pueden o no tener su propia barra de desplazamiento. Pueden ser tanto verticales como horizontales.
- **Media.** Permite inicializar elementos no visibles como la cámara o una grabación de pantalla, además de elementos visibles como reproductores de música o video.
- **Drawing and Animation.** Genera un lienzo en donde se pueden identificar gestos hechos por el usuario, además de realizar imágenes simples.
- **Map.** Permite agregar un contenedor para moverse en un mapa. Se pueden tener marcadores y otras cosas.
- **Sensors.** Contiene elementos no visibles que permiten obtener información de los sensores del dispositivo. Entre los más importantes, están el acelerómetro, el sensor de orientación, el giroscopio, de luz, un magnetómetro e incluso un escáner de códigos QR y temporizadores.
- **Social.** Permite realizar llamadas o conectar con la lista de contactos del dispositivo, enviar mensajes o una publicación de Twitter.
- **Storage.** Crea bases de datos almacenadas en la nube o localmente. También da la posibilidad de interactuar con archivos del dispositivo.
- **Connectivity.** Inicializa la conectividad con servicios WEB o el módulo Bluetooth.

La pantalla de código cuenta con dos categorías: la de bloques incorporados en la aplicación y los bloques pertenecientes a cada uno de los elementos puestos en la pantalla de Blueprint. De igual manera, se describirán brevemente cada una de las categorías de los bloques incorporados:

- **Control.** Contiene instrucciones condicionales como *if/else* e instrucciones como iterativas como *for*, *foreach* y *do-while*.
- **Logic.** Permite agregar bloques de verdadero o falso y evaluaciones de dos bloques.
- **Math.** Devuelve el resultado de operaciones matemáticas. Además de operaciones básicas, permite evaluar funciones como potencia, senos, cosenos, conversión de números, etc.

- **Text.** Realiza operaciones con cadenas de texto como concatenar, comparar, buscar letras o patrones, entre otras.
- **List.** Crea una lista ligada en donde se pueden agregar, quitar, buscar o reordenarlos elementos.
- **Dictionary.** Funciona de la misma manera que el anterior pero con diccionarios.
- **Colors.** Devuelve el color seleccionado. Se puede elegir entre bloques o seleccionándolo en el estándar RGB.
- **Variables.** Inicializa variables locales o globales en el proyecto.
- **Procedures.** Permite realizar procesos o funciones para ser llamados en cualquier parte del código.



## 3.6. Resumen

En este capítulo se presentaron tres elementos que componen el acondicionamiento de una señal: la amplificación, el filtrado y el aumento del offset. Los dos primeros, cuando se trabaja con una señal analógica, pueden ser realizados mediante el uso de un amplificador operacional en alguna de sus configuraciones o en el caso de los filtros también se presentaron las alternativas digitales, con sus respectivas ventajas. El aumento del offset se logra usando un divisor de tensión.

También se describió de manera general como se comporta la orientación de un teléfono inteligente, utilizando tres sensores: un acelerómetro, un giroscopio y un magnetómetro. La importancia de un celular en este proyecto radica en su implementación como una interfaz entre el usuario y el sistema, cuya aplicación fue desarrollada para Android usando la plataforma MIT App Inventor.



## Capítulo 4

# Periféricos del DSP: ADC, DAC y protocolos de comunicación

El surgimiento de las máquinas digitales ha hecho posible la implementación de diversos algoritmos del procesamiento digital de señales para resolver una cantidad innumerable problemas. Una parte fundamental de éstas máquinas son los ADC y los DAC, pues son un puente que conecta al mundo digital con el analógico. Su relevancia es tal, que a menudo se desarrollan sistemas digitales basados en cierto tipo de componentes analógicos que definen su funcionamiento en general. Por ejemplo, un DSP con un convertidor flash, cuya principal aplicación puede ser el procesamiento de video. Otra parte importante de un sistema digital es la capacidad de intercambiar información con otros dispositivos o sistemas, con el fin de distribuir las diferentes tareas existentes en un proyecto para aumentar la eficiencia y los tiempos de ejecución. Todo esto se logra utilizando una comunicación serial.

Se presenta, de manera general, las partes más importantes para entender el funcionamiento de los módulos que componen al subsistema analógico y a la comunicación serial SCI. Además, se describen las ventajas con las que cuenta el DSP TMS320F28377S al implementar los módulos antes mencionados como, por ejemplo, la cantidad de canales del convertidor analógico-digital o la tasa de Bauds a la que permite trabajar la comunicación serial SCI.

## 4.1. DSP: TM320F28377S

El procesador digital de señales (DSP) modelo TM320F28377S es la parte más importante de ésta tesis pues es el encargado de convertir la señal de audio analógica en digital, recibir e interpretar las modificaciones deseadas por el usuario, procesar la señal antes mencionada y transmitir el resultado por el convertidor digital analógica. A continuación se describirán las características más importantes de dicho DSP, cuya fotografía se puede observar en la Figura 4.1.

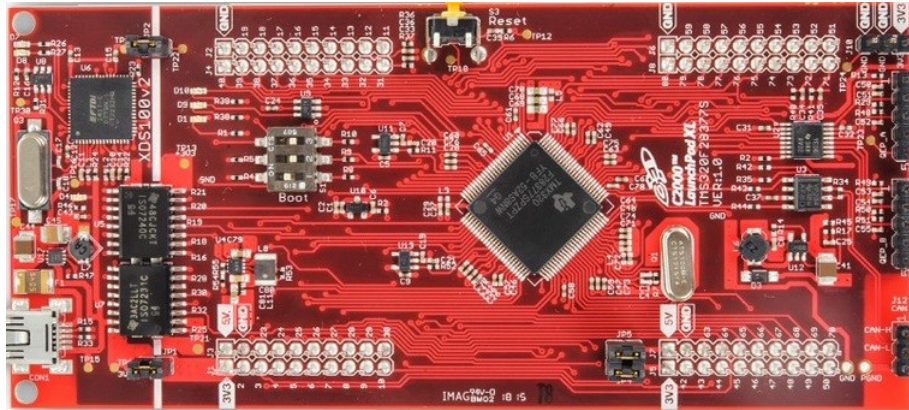


Figura 4.1: Vista frontal del DSP TM320F28377S, tomado de [41].

- Puede realizar hasta 400 Millones de instrucciones por segundo (MIPS) a 32 bits.
- Cuatro unidades de procesamiento: la Unidad Aritmética Lógica (ALU), la Unidad de Punto Flotante (FPU) encargada de realizar operaciones en un formato IEEE 754, la Unidad Trigonométrica-Matemática (TMU) y la Unidad de Viterbi (VCU-II) para realizar operaciones con números complejos.
- Unidad CLA (Acelerador de operaciones matemáticas) capaz de efectuar operaciones en paralelo con el CPU.
- Un MB de memoria Flash y 164 KB de memoria RAM.
- Seis canales de DMA (Direct Access Memory) para mover bloques de información entre la memoria del DSP.
- 169 Pines de entrada/salida de propósito general (GPIO).
- Dos controladores para una interfaz serial CAN, tres módulos SPI, dos módulos I2C y cuatro módulos SCI/UART.
- Hasta cuatro convertidores analógico-digital con resolución de 16 bits y 16 canales.
- Tres temporizadores de 32 bits y 12 módulos PWM con mejoras.
- Entre otras características.

## 4.2. Subsistema analógico

Muchas de las señales y magnitudes físicas que se utilizan para el procesamiento de señales y que se manejan en los sistemas digitales en general son de naturaleza analógica. Los sensores y transductores proveen los voltajes equivalentes de las señales de interés, pero hasta este momento un sistema digital no es capaz de procesarlas.

Un Convertidor Analógico Digital (ADC, por sus siglas en inglés) es un puente de conexión entre el mundo digital y el analógico pues, como su nombre lo indica, se encarga de convertir un voltaje del dominio analógico al discreto, y de esta manera la señal pueda ser analizada en un sistema digital. Ahora bien, una vez teniendo la información deseada y dependiendo de la aplicación, podemos regresar al mundo analógico por medio de un Convertidor Digital Analógico (DAC). Las principales aplicaciones de estos dispositivos se encuentran en áreas como instrumentación, audio, vídeo, entre otras. Sin embargo, también se pueden utilizar en aplicaciones más elaboradas capaces de controlar motores o fuentes conmutadas. La imagen 4.2 nos permite ver gráficamente la conexión entre estos subsistemas en el presente proyecto, en donde el ADC se encarga de adquirir y traducir una señal al dominio digital y el DAC se encarga de transferir la señal a un sistema analógico.

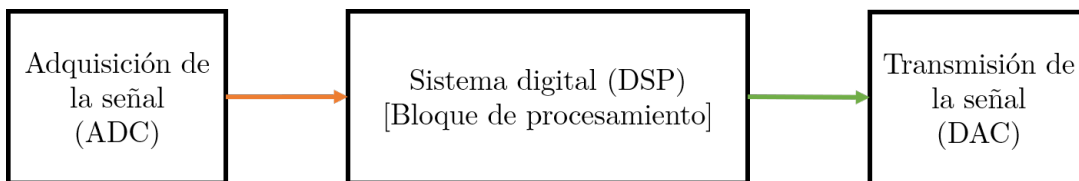


Figura 4.2: Representación del subsistema analógico de un sistema digital.

Antes de describir lo que es un ADC y un DAC propiamente, es importante hablar de la teoría que existe detrás de estos módulos; esto nos ayudará a tener un mejor entendimiento de su funcionamiento.

### 4.2.1. Modulación por codificación de pulsos

La modulación por codificación de pulsos (PCM), por sus siglas en inglés, es la manera más conocida de convertir una señal analógica continua en un conjunto de bits[42]. La codificación PCM está compuesta por tres partes principales: el muestreo de la señal, la retención y la cuantización, como se puede ver en la imagen 4.3.

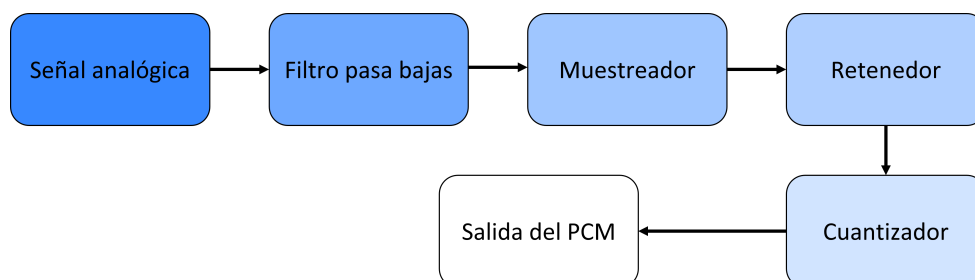


Figura 4.3: Modulación por codificación de pulsos o PCM.

### Teorema de Nyquist

El concepto más importante de toda conversión de una señal analógica en digital es la frecuencia de Nyquist. Como se ha explicado con anterioridad, nos indica que, para muestrear una señal, es necesario que esta frecuencia sea, como mínimo, dos veces mayor al ancho de banda de la señal analógica a muestrear.

Su relevancia es tal, que elegir de manera equivocada la frecuencia de muestreo puede derivar en diversos problemas. Por ejemplo, al elegir una frecuencia menor a la de Nyquist se genera un solapamiento de la señal o *aliasing*, que ocasiona que la señal digital no pueda ser reconstruida de manera correcta o sea una señal completamente diferente a la analógica. Elegir una frecuencia por encima de la frecuencia de Nyquist genera un *sobremuestreo* y esto requiere que el sistema digital utilice un exceso de memoria para almacenar la cantidad de información obtenida.

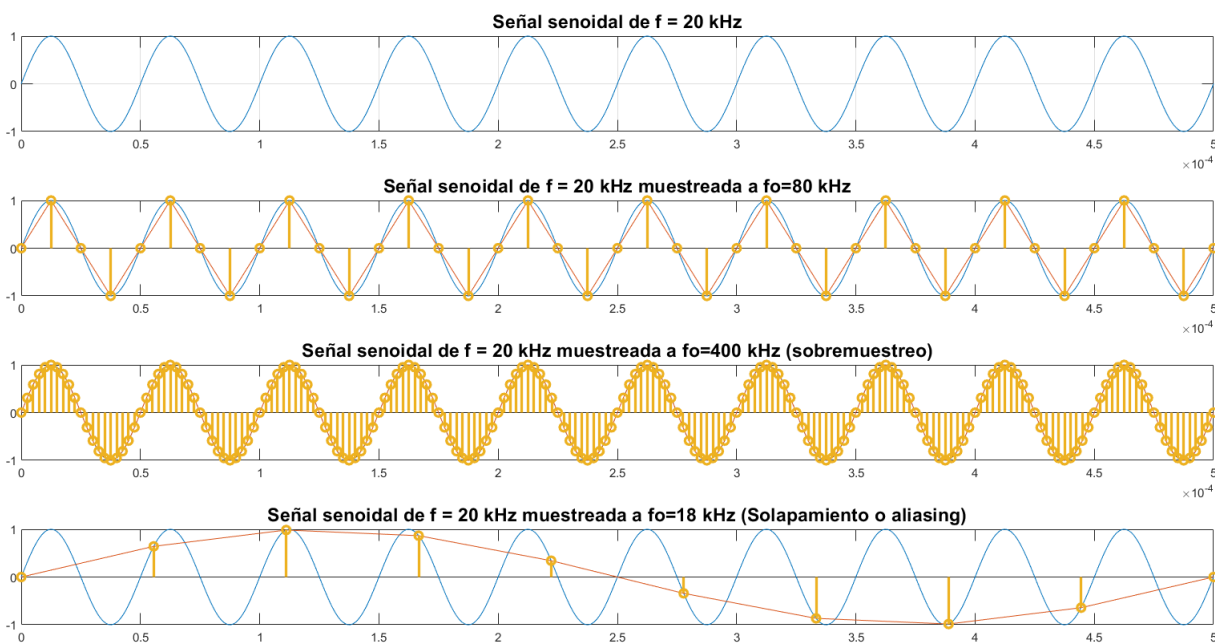


Figura 4.4: Ejemplo del muestreo de una señal senoidal de 20 kHz.

En la figura 4.4 podemos apreciar como la frecuencia de muestreo afecta la representación discreta de la señal. Para una frecuencia de muestreo de 80 kHz se cumple el teorema de Nyquist, y aunque la forma de la señal se asemeja más a la de una triangular, podemos observar que la naturaleza de la función (los cambios de signo en la magnitud) se mantiene. Cuando existe un sobremuestreo, la señal es visiblemente más parecida a una señal senoidal con las implicaciones que esto conlleva. Finalmente, en el solapamiento se puede ver que no hay relación alguna entre la señal analógica y discreta.

El filtro pasa bajas limita en banda la señal analógica de entrada y elimina armónicos indeseados. Su función es tal, que nos permite conocer con antelación la frecuencia de Nyquist y así muestrear correctamente la señal.

## Muestreo en el tiempo y la frecuencia

En términos generales, el muestreo de una señal analógica significa obtener el valor discreto de su amplitud a  $N$  bits cada cierto intervalo de tiempo, en otras palabras, consiste en convertir una señal analógica continua en una señal analógica discreta[43].

Para poder definir matemáticamente el proceso de muestreo, se utiliza la función impulso o delta de Dirac. Ésta función es peculiar y tiene bastantes aplicaciones en el análisis de señales, sin embargo, no puede ser completamente reproducida en la realidad, ya que no existe ningún instrumento o dispositivo capaz de imitar su comportamiento de amplitud infinita. Se define al impulso  $\delta(t)$  como:

$$\delta(t) = \begin{cases} \infty & \text{si } t = 0 \\ 0 & \text{si } t \neq 0 \end{cases} \quad (4.1)$$

A partir de esto, un tren de impulsos está definido de la siguiente forma:

$$\delta_T(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \quad (4.2)$$

donde  $T_s$  es el intervalo de tiempo que existe entre cada uno de los impulsos y corresponde al inverso de la frecuencia de Nyquist ( $T_s = 1/f_s$ ).

Al multiplicar la ecuación 4.2 por una señal  $x_s(t)$  y por las propiedades de la función impulso (véase Apéndice B) podemos obtener matemáticamente la definición del muestreo

$$x_s(t) = x(t)\delta_T(t) = \sum_{n=-\infty}^{\infty} x(nT_s)\delta_T(t - nT_s) \quad (4.3)$$

Ahora bien, utilizando la transformada de Fourier en la ecuación 4.3, tenemos qué:

$$X_s(\omega) = X(\omega) * \delta_T(\omega) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X(\omega)\delta_T(\omega - n\omega_s); \quad \omega_s = 2\pi f_s = \frac{2\pi}{T_s} \quad (4.4)$$

Finalmente, utilizando nuevamente las propiedades de la función impulso

$$X_s(\omega) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X(\omega - n\omega_s) \quad (4.5)$$

Ambas ecuaciones, 4.4 y 4.5, indican que el espectro en frecuencia de la señal muestreada se repite  $n$  veces, es decir, es periódico, y cada una de las repeticiones está centrada en  $n\omega_s$ , por lo tanto es simétrica.

La figura 4.5 muestra el espectro de una señal muestreada. Se puede observar que, al cumplir con el Teorema de Nyquist, el espectro resultante es periódico, simétrico y continuo.

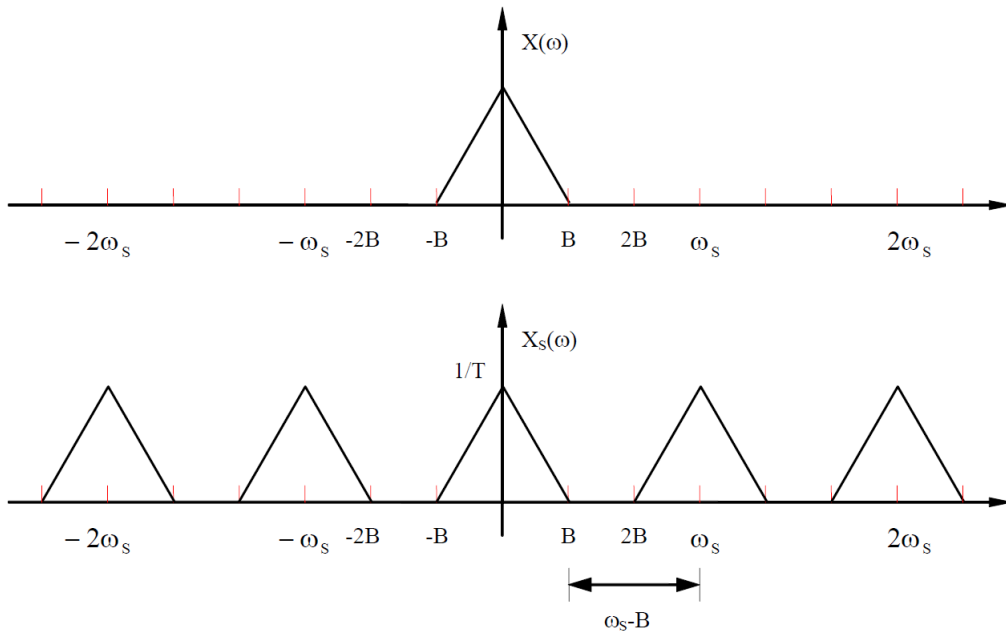


Figura 4.5: Espectro en frecuencia de una señal muestreada, tomada de [5].

### Cuantización

La cuantización se refiere al proceso de representar la amplitud de una señal analógica, cuyos valores están definidos en el campo de los número reales, en otro dominio, generalmente es el de los enteros. El valor original de la señal analógica es redondeado (o truncado) al valor más cercano de la nueva escala, por lo que siempre existirá un error que sin importar el diseño del convertidor estará presente y es fundamental para determinar su calidad. Este error se llama de cuantización, y se define de la siguiente manera:

$$e_q(t) = x(n) - x_q(n) \quad (4.6)$$

donde  $x_q(n)$  es el valor cuantizado de  $x(n)$ .

Finalmente, cada valor cuantizado es codificado de manera binaria debido a que facilita la implementación de algunos convertidores, aunque no es la única codificación que existe. Esto da como resultado que se agrupen cierta cantidad de bits para representar cada muestra cuantizada, y se define a  $L$  como la longitud de palabra o **resolución**[44]. Podemos obtener el valor de  $L$  como  $N = \log_2(M)$ , donde  $M$  son los posibles valores (niveles de cuantización) que pueden ser asignados a la muestra ( $M < 2^L$ ).

### Precisión

La precisión de un convertidor analógico digital está definida por la calidad de los niveles de la cuantización. Esto se refiere, qué tanto estos niveles se encuentran correctamente distribuidos con respecto al intervalo de voltajes que se deseen convertir y que la separación entre cada uno de los niveles sea la misma.



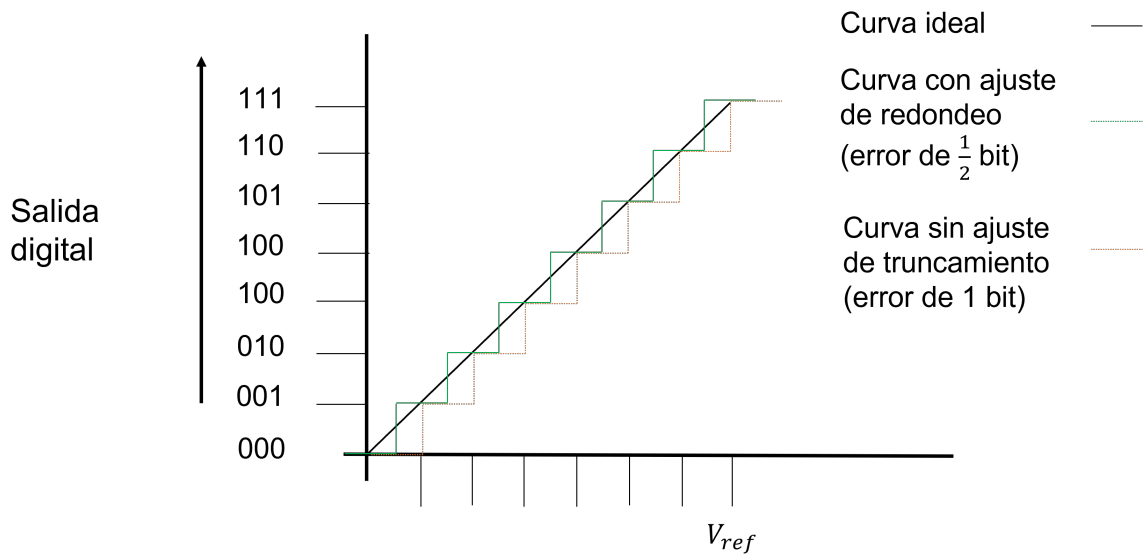


Figura 4.6: Ejemplo de los ajustes realizados a la función de transferencia de un ADC, con  $N = 3$  bits.

Como se puede ver en la imagen 4.6, entre más resolución tenga un ADC, más niveles de cuantización posee y el error de cuantización será cada vez menor. Dependiendo del sistema se puede tener más resolución; por ejemplo, si se tiene una longitud de palabra de ocho bits (256 niveles) para cuantizar 10 V, el mínimo valor que se podrá representar será  $min_V = \frac{10}{256} \frac{V}{nivel} = 39.0625 \frac{mV}{nivel}$ , por lo que una señal de 15 mV no podría ser representada correctamente. Sin embargo, si se tienen los mismos ocho niveles en otro sistema en donde el intervalo sea de 0 a 1 V, el mínimo valor que se puede representar será  $3.90625 \frac{mV}{nivel}$ , y si bien el valor más cercano que se puede representar para 15 mV será el nivel 4 o  $15.625 mV$ , se tiene un menor error entre el valor real y el cuantizado que en el primer caso.

#### 4.2.2. Muestreo y retención (Sample & Hold o S/H)

Hasta ahora, se mostró como se transforma una señal analógica en una señal digital por medio de la codificación PCM, sin embargo, la implementación de impulsos para obtener cada muestra es aplicable solo de manera teórica ya que como se mencionó anteriormente, no existe ningún dispositivo que sea capaz de representar la delta de Dirac. El S/H es una técnica que se utiliza para obtener el valor de la muestra en determinado tiempo de manera experimental.

Para darse una idea de como funciona este concepto, es necesario describir un circuito que antecede al S/H, y este es el "seguimiento y retención" (o Track & Hold [T/H]). El T/H es un circuito eléctrico que consiste en un interruptor y un capacitor, como se puede observar en la figura 4.7. Su funcionamiento se basa en cómo actúa la frecuencia  $t$  en el interruptor, con el fin de que, durante un número entero de intervalos de tiempo no consecutivos que sumados duran la mitad del período de la señal analógica, el circuito retenga el valor de la señal que se registra al momento del cambio de estado del interruptor, mientras que durante la mitad complementaria del periodo, el voltaje de salida seguirá al resto la señal de entrada.

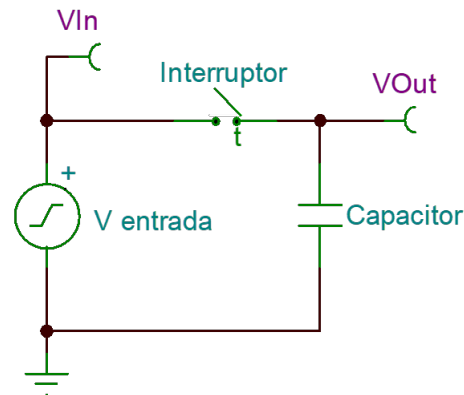


Figura 4.7: Circuito T/H

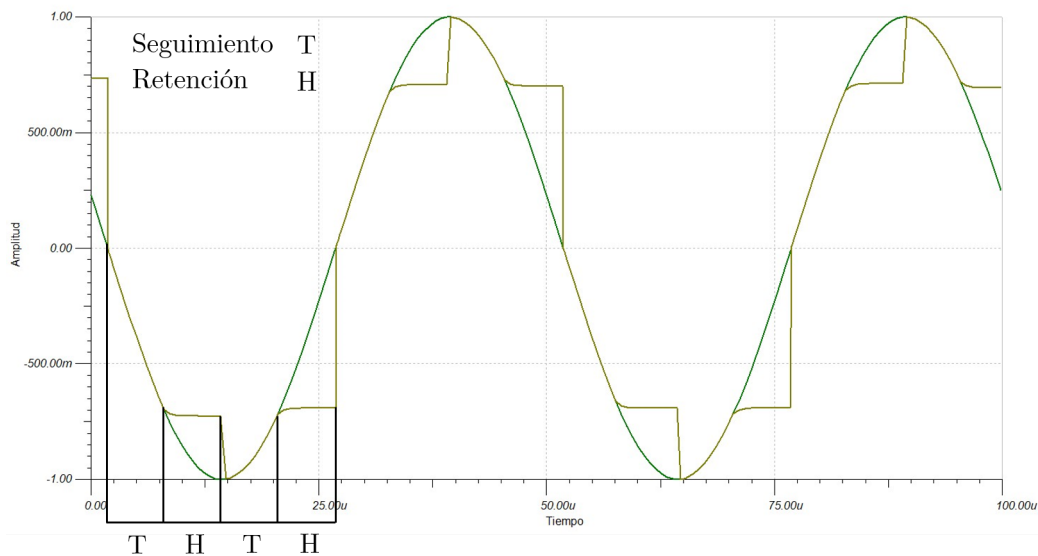


Figura 4.8: Simulación del circuito T/H.

Un circuito S/H se forma con dos circuitos T/H colocados en cascada, donde el interruptor del segundo circuito se activa de manera inversa al primero, es decir, primero sigue y después retiene los valores de la señal. De esta manera, se logra muestrear la señal en un ciclo completo de la frecuencia de Nyquist. Dependiendo del convertidor analógico digital, se puede tener uno u otro de los circuitos antes mencionados o ambos funcionando al mismo tiempo.

### 4.2.3. Convertidor Digital-Analógico (DAC)

Un Convertidor Digital-Analógico (DAC, por sus siglas en inglés), como su nombre lo indica, es el encargado de transformar una palabra de L-bits en una señal analógica. Los DAC tienen dos aplicaciones fundamentales dentro de un sistema digital: como ya se mencionó con anterioridad, se encargan de traducir una señal que ya ha sido procesada de manera digital al dominio analógico. También son utilizados por la gran mayoría de los Convertidores Analógico-Digital para comparar la muestra reconstruida y la muestra de entrada.

## Conversión de la información

La entrada de un DAC es una palabra de  $L$ -bits que se encuentra codificada por el sistema digital, por lo que el primer paso es saber como se representa la información. Existen varias formas de representarla, entre las que se pueden mencionar son la de magnitud y signo o a partir del código Gray, sin embargo, la más usada es la representación directa, donde no se tienen números negativos y cada valor binario es el mismo que el de su equivalente decimal.

Otro parámetro de entrada de un DAC es el voltaje de referencia, el cual este toma sentido cuando se conoce como se ha representado la información. En el caso de la manera directa, una palabra con todos los bits en alto significa que el voltaje de salida es igual al voltaje de referencia y, por lo tanto, una palabra con todos los bits en bajo significa un voltaje de salida de  $0V$ . Dicho de otra manera, los valores están dados por la división de la escala de  $[0 - V_{ref}]$  en  $2^L$  niveles. Debido a que los convertidores analógico-digital utilizan un DAC para realizar la comparación con el voltaje muestreado y de esta manera obtener el valor más cercano, a continuación se describe el funcionamiento de los DAC más comunes, donde las características que se buscan son: resolución, velocidad de respuesta y estabilidad.

## DAC de resistencias en escalera

Este tipo de DAC fue de los primeros en ser implementados y utilizados (alrededor de la década de 1950) por su fácil aplicación. Utiliza un arreglo de resistencias, que se encargan de dividir el voltaje en los distintos niveles, e interruptores controlados por un circuito decodificador que permiten o no el paso del voltaje dependiendo de la palabra de entrada. Esta conversión es más rápida que otras configuraciones y aunque teóricamente puede tener una buena resolución puesto que por cada bit en la longitud de palabra se necesita una resistencia y un interruptor, en la práctica presenta muchas limitaciones y sus aplicaciones están orientadas a sistemas de baja resolución[45].

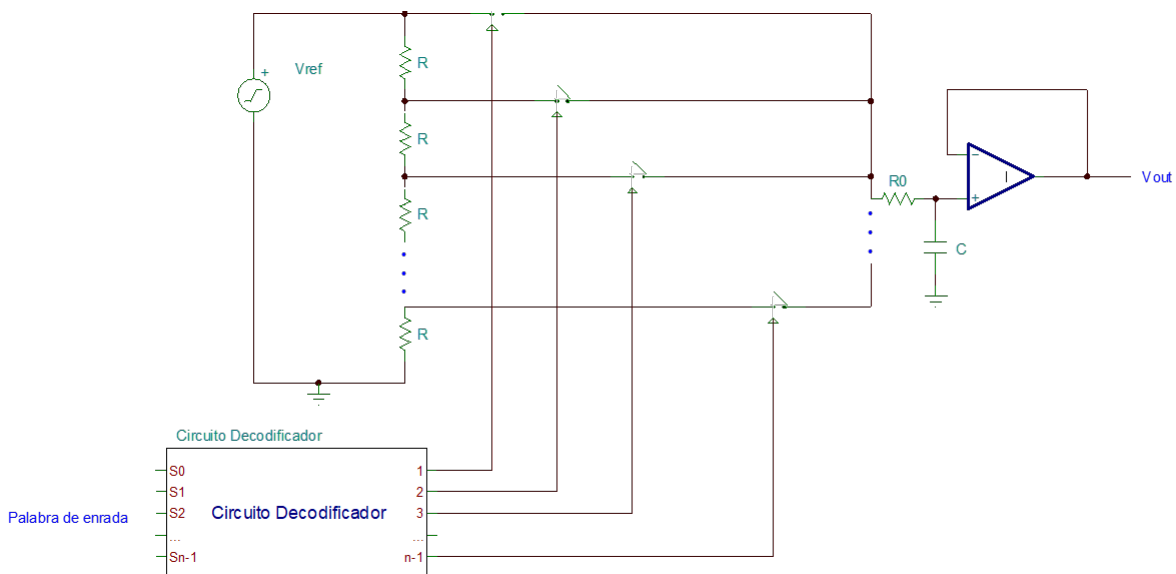


Figura 4.9: Esquema de un DAC de resistencias en escalera.

### DAC de arreglo R-2R

Al igual que la configuración anterior, el arreglo R-2R utiliza resistencias para dividir el voltaje de referencia a lo largo de la red y los bits de la palabra de entrada cambian la posición de los interruptores. Cada etapa 2R divide entre dos el voltaje anterior; el bit más significativo divide entre dos el  $V_{ref}$ , el siguiente bit divide nuevamente entre dos, por lo que se tiene un valor de  $0.25V_{ref}$  y así hasta acabar con la palabra de  $L - bits$ . Este tipo de arreglo es fácilmente escalable, por lo que se pueden tener desde resoluciones bajas de 4-6 bits hasta resoluciones más grandes de 16 bits sin perder velocidad de manera considerable.

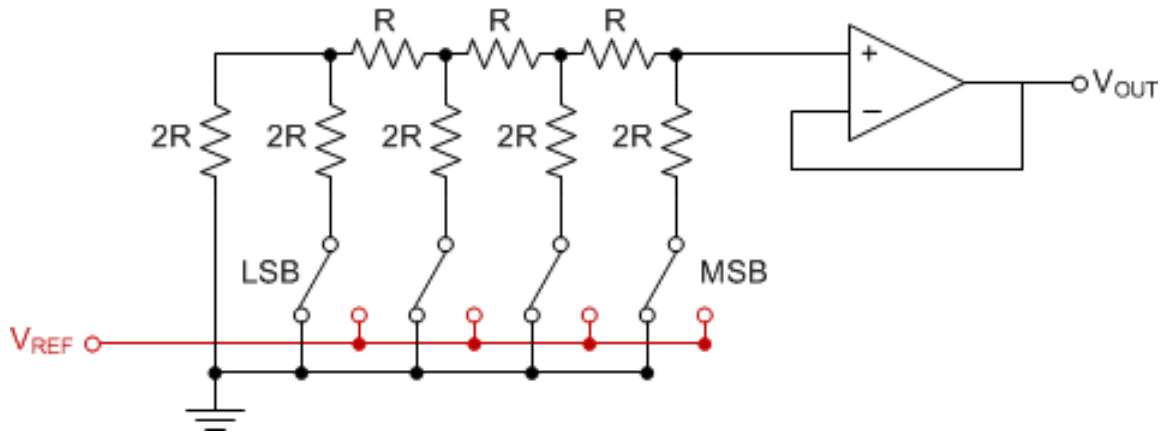


Figura 4.10: Esquema de un DAC con arreglo R-2R.

### 4.2.4. Convertidor Analógico Digital (ADC)

El convertidor analógico digital (ADC) es el dispositivo electrónico encargado de realizar el muestreo y transformación de una señal de voltaje analógico en información para el sistema digital. Existen diferentes tipos de convertidores, donde cada uno se especializa en cierta área dependiendo de lo que se desea convertir, por ejemplo, un convertidor flash lleva su nombre por realizar la conversión más rápidamente, pero con una menor resolución y son especialmente usados para procesamiento de video[42].

### 4.2.5. Tipos de ADC

Los convertidores descritos a continuación son los utilizados de manera más común en sistemas digitales. Las principales características que se buscan en estos convertidores son: velocidad de conversión, resolución y máxima tasa de muestreo. Cabe destacar que se considera que la señal antes de entrar a un convertidor ya ha sido acondicionada para no generar errores o conversiones erróneas.

Como preámbulo, es conveniente mencionar que la base de los ADCs son los comparadores. Un comparador es un dispositivo electrónico que mide una señal eléctrica y entrega una salida cuando ésta supera cierto valor de umbral. Podemos observar el diagrama básico de un comparador en la figura 4.11 que se muestra a continuación

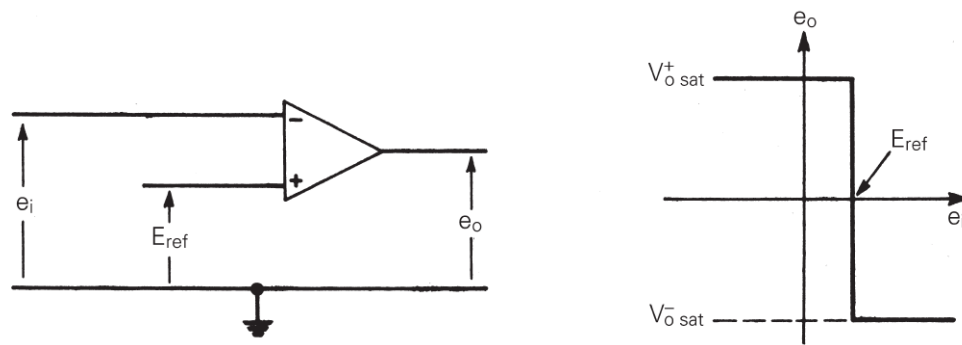


Figura 4.11: Esquema de un comparador y su señal de salida, tomado de [46]

De la figura 4.11 podemos observar que, de acuerdo con esta configuración, la salida de un comparador es binaria. Cuando la señal en la entrada no inversora es mayor a la inversora, la salida será aproximadamente el voltaje de saturación positivo del amplificador operacional, mientras que, siendo menor que la inversora, la salida será el voltaje negativo de saturación.

### ADC Flash

El convertidor flash recibe este nombre por ser el más rápido entre los convertidores, alcanzando tasas de muestreo del orden de giga Hertz (GHz) a cambio de una resolución de 6-8 bits. Su funcionamiento es sencillo: Utiliza  $2^L - 1$  comparadores en paralelo con sus respectivas resistencias a la entrada, creando así divisores de voltaje que ayudan a reducir la tensión en la entrada cada uno de ellos, y un codificador que se encarga de traducir la información a formato binario. El procesamiento en paralelo hace que los convertidores flash sean relativamente más grandes que otros ADC. Su alta velocidad hace que este convertidor sea especialmente útil para procesamiento de vídeo.

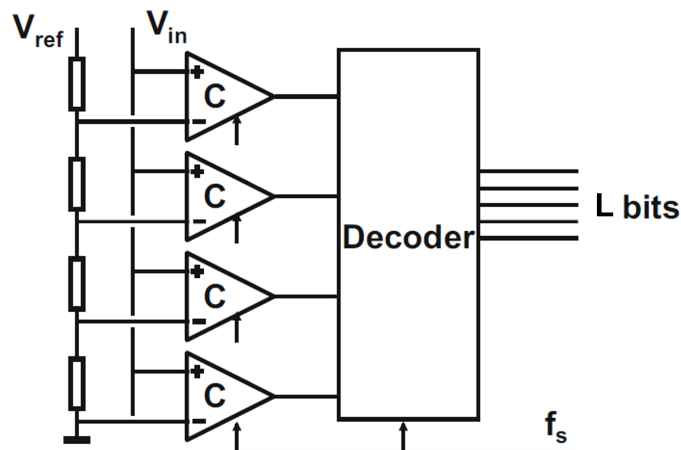


Figura 4.12: Esquema de un ADC flash, tomado de [43]

### ADC Rampa

El convertidor rampa o pendiente, está catalogado dentro de los convertidores "lineales" debido a su forma de operación. Cuando se ingresa una muestra se genera una rampa por medio de dispositivos digitales, y utilizando un convertidor digital-analógico se comparan ambos valores. Esta forma de conversión es caracterizada por ser lenta, ya que cada ciclo de reloj aumenta en uno el valor de la rampa hasta alcanzar el valor a comparar. Sus velocidades alcanzan el orden de los kilo Hertz (kHz), con una resolución de 10-12 bits y son utilizados para el procesamiento de imágenes.

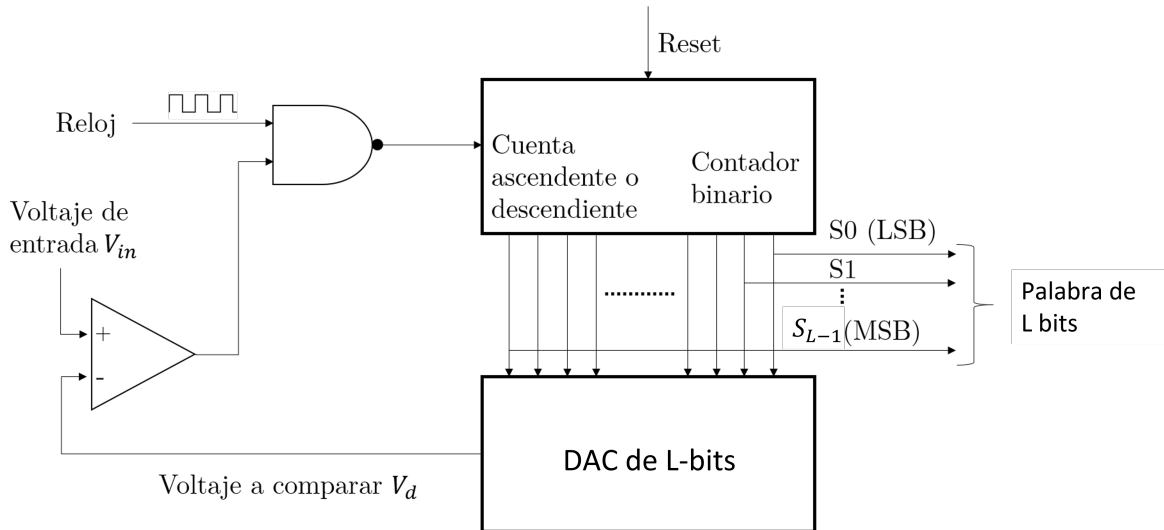


Figura 4.13: Esquema de un ADC de tipo Rampa, tomado de [47].

### ADC de aproximaciones sucesivas

El convertidor de aproximaciones sucesivas (SAR, por sus siglas en inglés) se basa en comparar el valor de entrada con una cadena de bits, por lo que para un ADC de resolución  $L$ , se necesitan la misma cantidad de ciclos de reloj para obtener el valor convertido. La primera operación se logra poniendo en alto el valor del bit más significativo (MSB, por sus siglas en inglés), por lo que se compara el valor de la muestra con  $0.5V_{ref}$ ; si el valor de entrada es más grande se guarda el valor en alto del bit, en caso contrario se guarda el valor en bajo. La siguiente comparación se hace con el segundo bit más significativo, por lo que ahora se compara la muestra con  $0.75V_{ref}$  o  $0.25V_{ref}$  dependiendo del resultado del primer caso, guardando de igual manera la respuesta en alto o bajo dependiendo de la comparación y posteriormente realizando las comparaciones sucesivas para cada elemento de la cadena de bits hasta terminar con el bit menos significativo. Se puede decir que esta implementación es un punto medio entre la conversión rampa y la conversión flash, pues ofrece velocidades del orden de MHz y resolución de 12-16 bits.

### ADC Sigma-Delta

Los convertidores Sigma-Delta llevan su nombre por utilizar la modulación delta y una retroalimentación que compara la señal de entrada con pequeños incrementos ( $\Delta$ ) de la señal reconstruida.

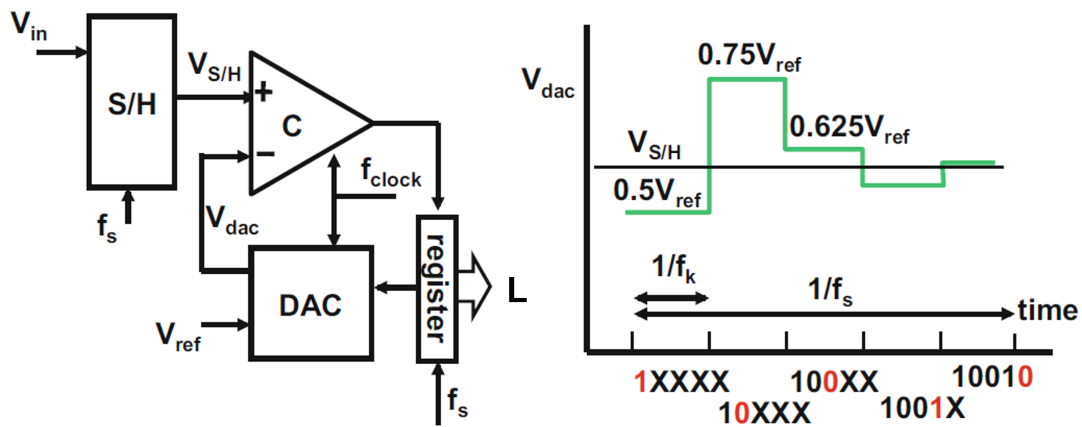


Figura 4.14: Esquema de un ADC de aproximaciones sucesivas con su señal de salida, tomado de [43].

A grandes rasgos, este tipo de convertidores utilizan los siguientes conceptos de procesamiento de señales para su funcionamiento, que mejoran considerablemente la resolución de los ADC: el sobremuestreo y el moldeo del ruido de cuantización (*Noise Shaping*). Sus principales aplicaciones son para convertir señales de en el intervalo de baja y muy baja frecuencia, como la voz o señales biométricas.

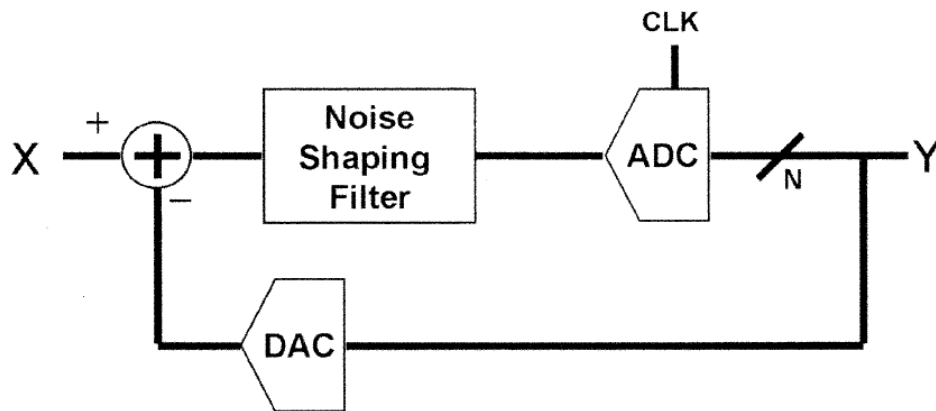


Figura 4.15: Esquema de un ADC  $\Sigma\Delta$

#### 4.2.6. ADC en el DSP TMS320F28377s

El módulo analógico del DSP cuenta con cuatro ADC de aproximaciones sucesivas. Las características específicas son enlistadas a continuación:

- Resolución seleccionable de 12 o 16 bits.
- Posibilidad de establecer los voltajes de referencia.
- Permite establecer dos modos: el modo diferencial y el modo único (single).
- Entradas multiplexables de hasta 16 canales en modo único u 8 canales en modo diferencial.
- Múltiples fuentes de disparo (EPWMs, TIMERS, XINTs, etc.)
- Cuatro banderas de interrupciones.
- Cuatro bloques de pos-procesamiento de la información.

Se puede definir una ecuación lineal que define al voltaje que convertirá el ADC a partir de su resolución, la cuál es la siguiente:

$$ADC_{result} = 2^N * \left( \frac{ADC_{iny} - V_{REFLO}}{V_{REFHI} - V_{LOFHI}} \right) \quad (4.7)$$

Para inicializar el módulo ADC del DSP, es necesario seguir las siguientes instrucciones:

1. Se deben configurar los parámetros más importantes del ADC como: Modo de operación, resolución, ADC a utilizar, el preescalador, donde se dispara la interrupción (Se puede hacer al principio o al final de la conversión) y encender el módulo interno del ADC seleccionado. La razón por la que se agrupan de esta manera todas éstas características es debido a que se recomienda su configuración usando la función `Adc-SetMode()`, que se puede encontrar en el C2000Ware.
2. Esperar a que se encienda el módulo (aproximadamente 1000[ms]).
3. Realizar la configuración de la fuente de interrupción del SOC, en caso de ser necesario.
4. Realizar la configuración del SOC:
  - Seleccionar el canal a utilizar con el registro `ADCSOCxCTL[CHSEL]`.
  - Configurar el tiempo de la ventana de adquisición de datos con `ADCSOCxCTL[ACQPS]`.
  - Seleccionar la fuente de disparo del ADC con `ADCSOCxCTL[TRIGSEL]`.
5. Habilitación de las interrupciones.
6. Seleccionar la prioridad de los canales del ADC.
7. Habilitar el funcionamiento del disparador del SOC para iniciar la conversión.



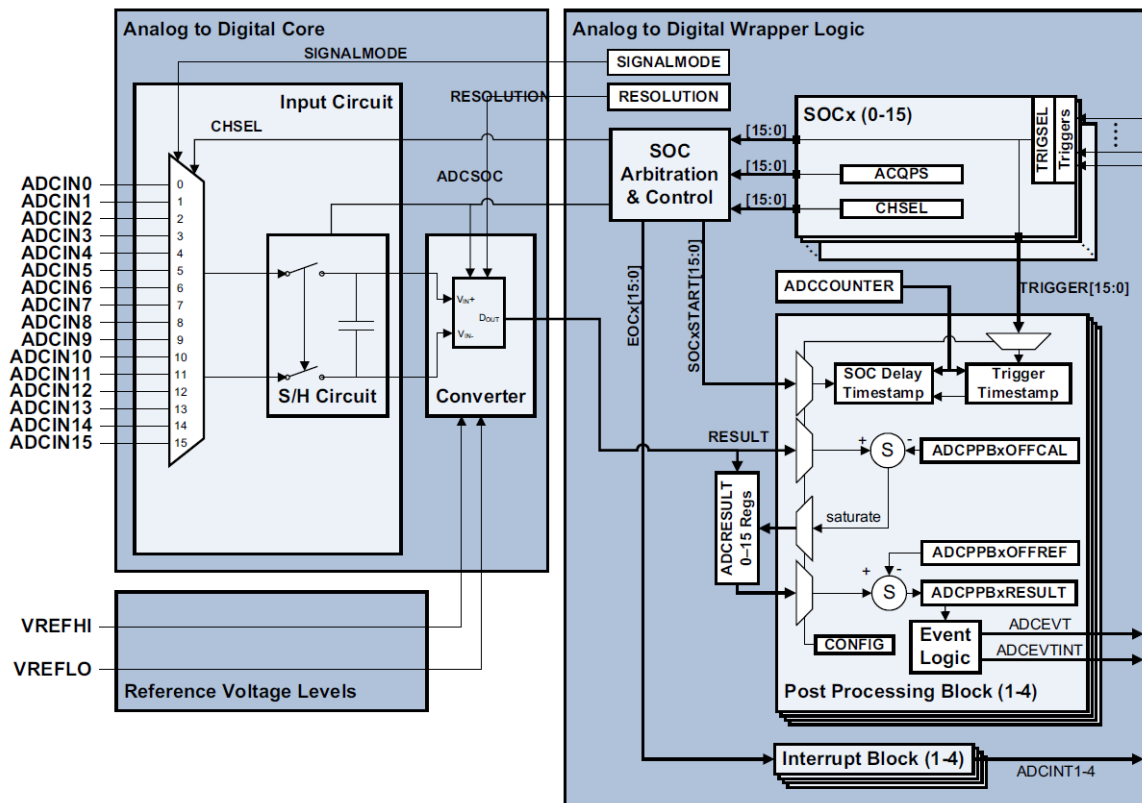


Figura 4.16: Módulo ADCC contenido en el DSP TMS320F28377s, tomado de [48].

### Principio de operación SOC

Se denomina SOC (Start-of-Conversions) a un conjunto de registros que indican el inicio de todo el proceso de muestreo y obtención de la señal del ADC: en primer lugar está el disparador, que indica el inicio de la señal; segundo, la duración de la ventana de adquisición, y finalmente el canal que convertirá el ADC. Cada uno de los cuatro convertidores cuenta con hasta 16 SOC's, uno para cada canal (si se trabaja en el modo único).

- Disparadores del ADC: existen diferentes métodos para indicar el inicio de la conversión del ADC, como los son los EPWMs, TIMERS, GPIOs o los disparadores por software. Los EPWMs y TIMERS pueden ser configurados para que cada cierto intervalo de tiempo habiliten al ADC. Tanto el uso de GPIOs y el disparo por software ofrecen un mayor control de inicialización, ya que disparan el proceso de conversión solo cuando es necesario y pueden ahorrar recursos si el proyecto lo requiere.
- Ventana de adquisición de datos: se refiere al tiempo en el que el módulo ADC tomará la muestra de la señal desde que se inició la conversión. El mínimo valor de la ventana de adquisición es de 75ns o 15 ciclos de reloj.
- Canales del ADC: cada módulo del ADC cuenta con 16 canales multiplexados para muestrear diferentes señales. Es importante destacar que algunos canales están conectados con otro módulo como el DAC o comparadores, por lo que es necesario consultar el manual técnico para saber qué canal es mejor utilizar en cada caso.

## TIMERS y EPWMs

Los relojes en los sistemas digitales nos ayudan a sincronizar las diferentes tareas durante el procesamiento digital.

Los TIMERS son bloques digitales internos del DSP capaces de generar una señal digital después de transcurrido un determinado intervalo de tiempo y su utilidad se centra en generar interrupciones en el sistema con el objetivo de poder realizar diferentes procesos en paralelo. En cuanto a los ADC, se puede configurar un TIMER para poder muestrear una señal cada cierto intervalo de tiempo o generar una muestra específica en el tiempo en un instante definido.

EL DSP cuenta con tres TIMERS (0, 1 y 2) de 32 bits, esto quiere decir que cada uno puede tener  $2^{32} - 1$  posibles valores con un pre-escalador de 16 bits cada uno. El reloj de entrada en cada uno de estos TIMERS es el SYSCCLK o reloj del sistema.

Su operación es la siguiente:

1. El registro PRDH:PRD de 32 bits almacenan el valor de la cuenta y cargan en el registro TIMH:TIM el valor cada que se reinicia la cuenta.
2. La cuenta del registro TIMH:TIM decrete una vez cada  $(TDDR:H:TDDR+1)SYSCCLK$ .
3. En el caso del ADC, cada que la cuenta del registro TIMH:TIM llega a cero, se genera una señal que habilita a un SOC.

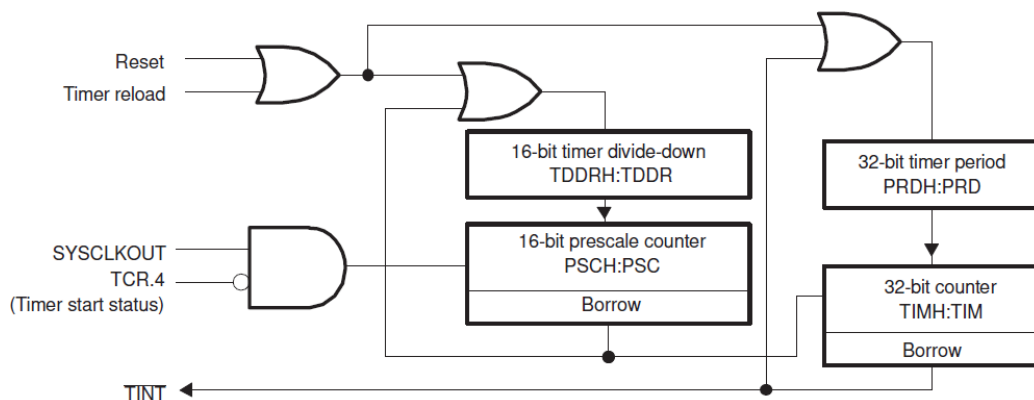


Figura 4.17: Módulo TIMER contenido en el DSP TMS32028377s, tomado de [48].

Los PWM (Pulse Width Modulation o modulación por ancho de pulso) son dispositivos electrónicos capaces de cumplir con las mismas tareas que un TIMER, pero por su construcción también permite, a partir de la salida de sus periféricos, controlar otros dispositivos externos como motores o de fuentes conmutadas. Además, el DSP cuenta con una versión mejorada de los PWM llamados ePWM (enhanced PWM o PWM mejorado), que agrega funciones de protección para estos módulos.

El DSP cuenta con 12 ePWM independientes entre sí que están sincronizados al reloj del sistema. Cada uno cuenta con distintos registros como pre-escaladores y divisores así como con la posibilidad de tener una cuenta ascendente o descendente. También cuentan con dos comparadores que pueden producir una señal por separado, las cuales pueden ser utilizadas por el DSP como interrupciones, o específicamente para el módulo ADC y DAC, como controladores de eventos que posteriormente disparan un SOC (Event Trigger).

### **Prioridad de conversión de un ADC**

En los sistemas electrónicos, las interrupciones son eventos que modifican el flujo del programa[49]. Estas son comúnmente utilizadas para realizar algún sub-proceso o para intercambiar información entre periféricos. Las interrupciones tienen una prioridad que debe ser definida en el sistema antes de ser utilizadas, y ésta significa que tan importante es para el sistema atender dicha interrupción. Si son disparadas dos interrupciones en el mismo instante, el sistema atenderá primero la de mayor prioridad. Esto aplica de igual manera a los SOCs del ADC, pero existen diferentes maneras de asignar prioridades:

1. Round Robin: su operación consiste en colocar cada uno de los SOCs en orden, uno después del otro, donde el SOC15 es el anterior al SOC0, y de esta manera, cerrar la lista circular. El registro RPPOINTER es el que otorga la prioridad al SOC inmediatamente mayor al valor que contenga este registro y su valor se reasigna cuando existe una conversión. Inicialmente, el registro tiene el valor de 16, lo que otorga la más alta prioridad al SOC0; si llegase una conversión en el SOC4, el módulo realizaría el procesamiento correspondiente y el RPPOINTER cambiaría su valor a 4, siendo el SOC5 el más alto en cuanto a prioridad se refiere. Siguiendo el ejemplo, si llegaran dos señales, una en el SOC0 y otra en el SOC7, se realizaría primero la conversión en el SOC7, poniendo en espera al SOC0.
2. Alta prioridad: este modo combina el Round Robin con la posibilidad de otorgar prioridad a ciertos registros. Al modificar el valor del registro SOC PRIORITY, se otorgará una mayor prioridad a los registros anteriores al número escrito, es decir, si es escrito el número 3, los SOCs del 0 al 2 tendrán alta prioridad, siendo el SOC0 el más importante. Los demás, funcionarán de acuerdo con el Round Robin.
3. Burst Mode: el modo burst, implica que varios SOCs son convertidos al mismo tiempo y sigue el funcionamiento del Round Robin. Por ejemplo, se puede convertir el SOC0 y SOC1 al mismo tiempo, donde el RPPOINTER tomará el valor del SOC más alto.

Además de manejar cada uno de los modos de manera individual, el módulo ADC es capaz de combinar los tres al mismo tiempo. Como hemos visto, la base siempre es el Round Robin, pero se puede dar alta prioridad a los SOCs que sean necesarios y al mismo tiempo, convertir más de dos señales cuando un trigger lo indique.

### 4.2.7. DAC en el DSP TMS320F28377s

El módulo digital-analógico cuenta con cuatro DACs capaces de entregar un voltaje de salida con una resolución de hasta doce bits. Cada uno cuenta con una resistencia de pull-down que permite entregar un voltaje estable a la salida. Existen dos formas de sincronizar la salida de un DAC, utilizando el reloj SYSCLK donde los cambios de voltaje toman efecto inmediatamente, o utilizando un EPWM, para que la salida cambie cada que se dispara un evento. La ecuación que define el voltaje de salida es:

$$V_{DACOUT} = \frac{DACVALA * V_{DACREF}}{4096} \quad (4.8)$$

donde  $V_{DACOUT}$  es el voltaje a la salida del DAC,  $DACVALA$  corresponde al valor que contiene el registro del mismo nombre y  $V_{DACREF}$  es el voltaje de referencia (El DSP permite seleccionar un voltaje externo como referencia o usar el interno de 3[V]).

Su inicialización consta de cuatro pasos:

1. Habilitar el reloj del DAC.
2. Establecer el voltaje de referencia  $V_{DACREF}$  utilizando el registro DACREFSEL.
3. Habilitar la salida del DAC con el registro DACOUTEN.
4. Esperar un momento para permitir al DSP habilitar el módulo (aproximadamente 10[ms]).
5. En caso de ser necesario, habilitar el registro DACCTL[LOADMODE] para establecer la sincronización con un EPWM.

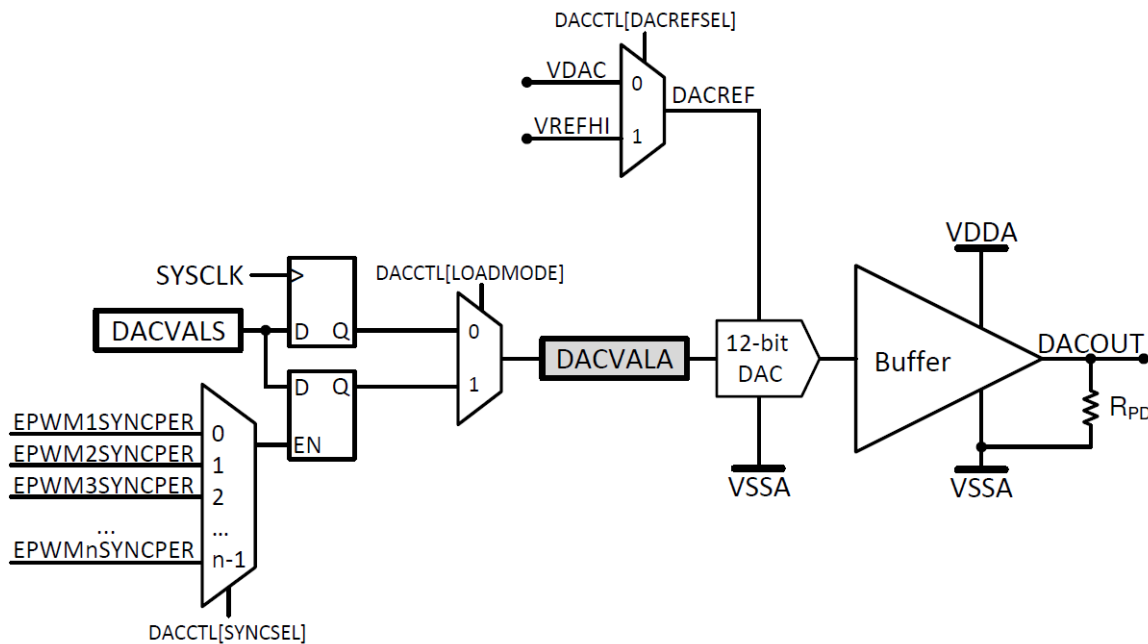


Figura 4.18: Módulo DAC contenido en el DSP TMS320F28377s, tomado de [48].

## 4.3. Comunicación serial

Una forma en la que dos o más sistemas digitales pueden intercambiar información se logra mediante la comunicación serial. Un ejemplo de este intercambio de información es la conexión entre un microcontrolador, cuya función puede ser la de adquisición de datos, y un DSP que procesa la información recibida.

En este tipo de comunicación, la información es enviada a través de tramas de bits que han sido codificadas y que pertenecen a uno de los distintos protocolos de comunicación que, a su vez, también indican la manera en la que deben estar conectados los sistemas para lograr la transferencia de datos.

### 4.3.1. Características de una comunicación serial

A continuación se describirán algunas características que poseen las comunicaciones seriales, entre las que se puede mencionar la forma en la que se transmite la información entre dos sistemas, si el sistema es síncrono o asíncrono o la velocidad en la que se comunican los sistemas.

#### Forma de transmisión

Existen tres maneras en las que un sistema puede transmitir información: simplex, half-duplex y full-duplex.

- Simplex: la información fluye de manera unidireccional, es decir, un sistema envía información y el otro la recibe.
- Half-Duplex: la información puede viajar entre ambos sistemas, es decir, ambos pueden enviar y recibir información pero no al mismo tiempo.
- Full-Duplex: la información puede viajar entre ambos sistemas y al mismo tiempo.

#### Tipo de sincronización

Existen dos tipos de sincronización para una comunicación serial: síncrona y asíncrona.

Si la comunicación es síncrona, uno de los sistemas conectados envía una señal de reloj, al otro sistema por medio de un cable. Su principal ventaja es que presentan un mejor rendimiento en cuanto a velocidad pero son más difíciles de implementar.

Si la comunicación es asíncrona, cada uno de los sistemas conectados cuenta con su propio reloj, los cuales deben funcionar a la misma frecuencia para poder transmitir información. Su principal ventaja es que son sistemas más fáciles y baratos de implementar pero sus aplicaciones son para comunicación de baja velocidad.

### Velocidad de transmisión

La velocidad de transmisión es un parámetro que nos indica que tan rápido se transmite la información y por lo general, se mide en [bits/s]. Independientemente si el sistema es síncrono o asíncrono, se puede establecer la velocidad con base en la frecuencia de reloj a la que trabajan los sistemas.

#### 4.3.2. Trama de datos

Como se mencionó con anterioridad, cada uno de los diferentes protocolos cuenta con una forma específica de enviar una trama de datos. Como una generalidad, todas cuentan con cierto número de bits de datos, bits de sincronización y bits de paridad.

- Bits de datos: estos bits contienen la información que se desea transmitir. Dependiendo del protocolo de comunicación se puede tener más información añadida a esta parte de la trama, como la dirección lógica del receptor.
- Bits de sincronización: estos bits se usan para indicar el inicio de una trama de comunicación y/o el final o una trama intermedia, en donde el receptor envía bits de respuesta al transmisor para poder continuar con la comunicación.
- Bits de paridad: existen bits que no pertenecen a la información que se desea transmitir ni tampoco forman parte de la sincronización, estos son los bits de paridad. Regularmente estos bits son usados para la detección de errores. Existen dos tipos de paridad: en la primera, se cuenta el número de bits en alto y, en caso de ser necesario, se envía un bit más en alto para que el total de 1's sea un número non; la segunda forma es el mismo concepto pero el total de 1's siempre debe ser par (de ahí sus nombres **paridad non** y **paridad par**). En ambos casos cuando se hace la comparación se debe obtener el mismo resultado de 1's tanto en la trama del receptor como en la que se enviaron los datos.

#### 4.3.3. Protocolo de comunicación SCI

SCI (Serial Communication Interface) es un tipo de comunicación serial asíncrona también conocida como UART (Universal Asynchronous Receiver Transmitter). La principal característica de este protocolo de comunicación es que sólo se necesita un par de conexiones para realizar el intercambio de información con el dispositivo electrónico. Las características generales del SCI son descritas a continuación.

- Utiliza el formato estándar de no retorno a cero (NRZ).
- Se realiza la conexión de transmisor y receptor del periférico con el dispositivo electrónico y en caso de ser necesario, entre el receptor y transmisor de los mismos. Esto quiere decir que la comunicación puede ser full-duplex o half-duplex.
- Su velocidad de transmisión está expresada como tasa de Bauds o baudaje (Baud Rate) variable.
- La longitud de la trama puede ser de uno hasta ocho bits que depende del dispositivo electrónico.

- Posee bits de inicio, de paro y de paridad.
- Se clasifica como un protocolo de comunicación asíncrono.

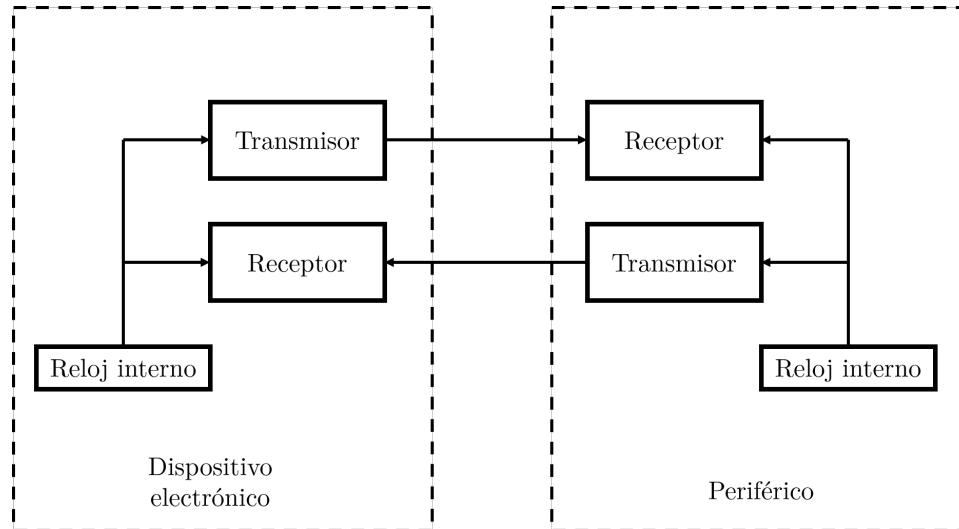


Figura 4.19: Diagrama de bloques de un protocolo de comunicación SCI.

### Transmisión de la información

Dado que la interfaz de comunicación es asíncrona, debe ocurrir un evento que indique que se va a transmitir información a cualquiera de los receptores. Inicialmente, la interfaz tiene un valor constante en alto; cuando se requiere transmitir información, se envía un bit en bajo, al que podemos identificar como bit de inicio, seguido de la trama de  $N$  bits y, en caso de haber sido habilitado, un bit de paridad, para finalizar regresando al estado inicial en alto. Un ejemplo gráfico de la trama lo podemos ver en la figura 4.20.

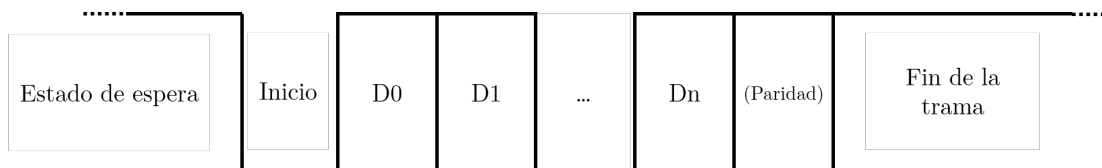


Figura 4.20: Ejemplo gráfico de la trama del protocolo de comunicación SCI.

### Sincronización

Siendo la interfaz de comunicación de tipo asíncrona, los relojes de los dispositivos electrónicos a conectar son completamente independientes entre sí. Sin embargo, tenemos que definir la tasa de Bauds, la cual, para transmisiones binarias equivale a la velocidad de transmisión de bits por segundo [50]. Ambos dispositivos deben tener la misma velocidad de transmisión, de lo contrario se generarán errores en las tramas receptoras.

Una técnica para evitar errores al enviar información es el sobremuestreo que consiste en muestrear el nivel de la señal  $s$  veces por cada bit que haya en el receptor. Por lo tanto,

si existen perturbaciones en las conexiones en forma de ruido, éstas no serán tomadas en cuenta como parte de la información (por ejemplo, un nivel bajo como el bit de inicio, pero de menor duración).

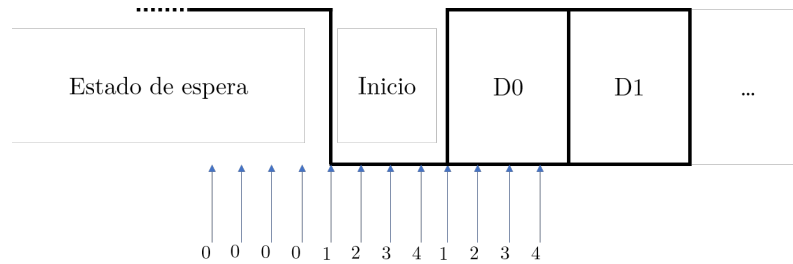


Figura 4.21: Ejemplo gráfico de el sobremuestreo en la transmisión de información, con  $s=4$ .

#### 4.3.4. SCI en el DSP TMS320F28377s

El módulo SCI manejado por el DSP agrega ciertas funciones o mejoras en el procesamiento de los datos para que, tanto las tramas de datos como los errores puedan ser procesados y se realicen diversas acciones dependiendo de cada caso (pueden ser mediante encuesta o interrupciones). Las características de este módulo son las siguientes:

- La tasa de Bauds programable es de hasta 64,000 posibles valores. Lo que permite al dispositivo adaptarse a los diferentes valores existentes en los periféricos. Además, permite la opción de “Auto-Baudaje”, en donde el DSP detecta la velocidad de transmisión del dispositivo electrónico externo y ajusta los valores internos del módulo para poder enviar o recibir información.
- Cuenta con cuatro banderas de interrupción: paridad, sobrescritura, trama y de ruptura.
- Utiliza Doble buffer tanto en la transmisión como en la recepción.
- Interrupciones independientes entre transmisión y recepción.
- Posee dieciséis niveles FIFO (Primero en entrar, primero en salir, por sus siglas en inglés) para transmisión y recepción.
- Se puede configurar como “Address mode”, que permite la transmisión de datos entre múltiples procesadores a través de una sola línea.
- Entre otras características.



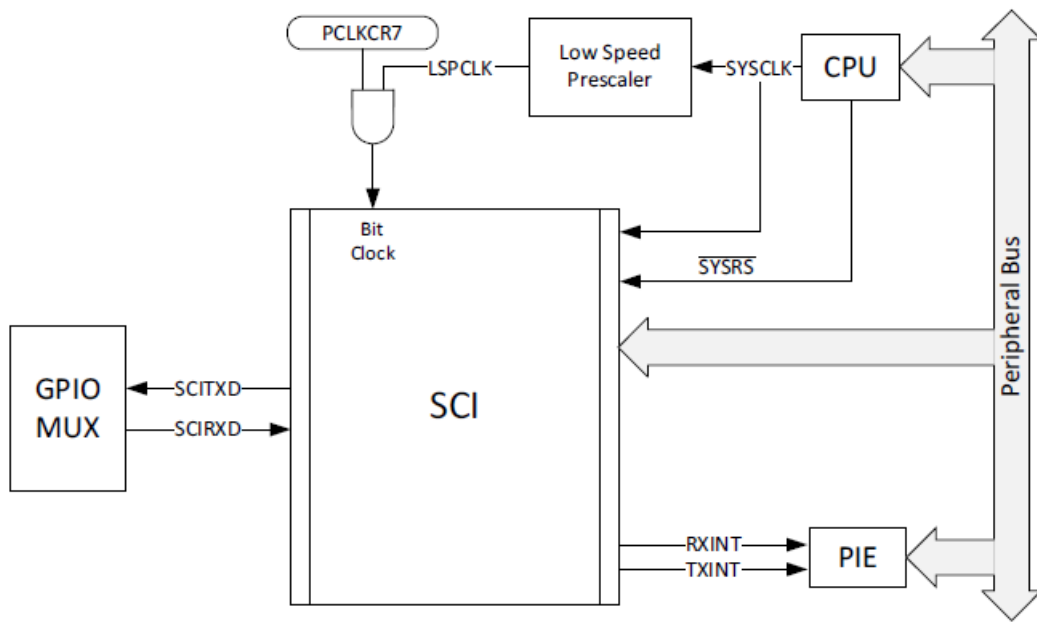


Figura 4.22: Módulo SCI en contenido en el DSP, tomado de [48].

Para inicializar el módulo SCI en el DSP, es necesario seguir las siguientes instrucciones:

1. Configurar el registro SCICCR donde se limitan parámetros como tipo de paridad, bits de paro, longitud de palabra, entre otras cosas.
2. Habilitar transmisor y receptor del módulo con el registro SCICTL1.
3. Habilitar las interrupciones de recepción y transmisión.
4. Establecer la tasa de Bauds en los registros SCIHBAUD y SCILBAUD de acuerdo con la siguiente ecuación:

$$\text{Tasa de Bauds } T_B = \frac{\text{LSPCLK}}{(((\text{SCIHBAUD} \ll 8 + \text{SCILBAUD}) + 1) * 8)} \quad (4.9)$$

5. Configuración de la FIFO de transmisión y recepción, en caso de utilizarlas.

## 4.4. Resumen

En este capítulo se presentaron diversos conceptos referentes a la comunicación entre un sistema digital y un sistema analógico así como entre sistemas del mismo dominio. Se presentaron también, conceptos que son de suma importancia no solo para la conversión analógica digital, sino para el procesamiento de señales en general, tales como la frecuencia de muestreo o de Nyquist. Se expusieron los tipos de ADC y DAC más comunes y se mencionaron algunas ventajas y desventajas que se tiene al usarlos. A su vez, se describieron los módulos del DSP TMS320F28377s que engloban estas herramientas, se enlistaron algunas de sus características y se dejó en claro por que serán utilizados a lo largo de este documento. Por otra parte, se habló de las comunicaciones seriales y las partes fundamentales que la componen, como la trama o la velocidad de transmisión.

El módulo analógico es utilizado en este proyecto para muestrear una señal de audio, ser procesado en el DSP y después transmitido a una bocina. El módulo HC05 visto en el capítulo tres utiliza la comunicación serial SCI para transmitir información. De esta manera, el usuario puede cambiar parámetros de la señal al conectar con el HC05 con su teléfono inteligente.

# Capítulo 5

## Diseño e implementación del sistema

Un sistema electrónico con la capacidad de modificar señales de audio de manera digital está compuesto por dos componentes principales: el *Hardware* dedicado a identificar la entrada del sistema y reproducir a la salida diferentes señales de audio, y un subsistema que te permita generar, almacenar y realizar modificaciones en la señal digital a través de un *Software* dedicado (subsistemas embebidos tales como microcontroladores, DSP, FPGA, etc). El diseño del sistema dependerá, en cuestión de *Hardware*, de muchos factores tales como las características de la señal a tratar, de la precisión a la que se desee muestrear, del costo y disponibilidad de los componentes electrónicos que componen el sistema entre muchas otras cosas, así como también los elementos añadidos que se quieran tomar en cuenta que puedan mejorar la interacción entre el usuario y el sistema. En cuestión de *Software*, el diseño se encuentra ligado al tratamiento que se le quiere dar a la señal de audio (la forma en que se quiere modificar la señal de entrada para obtener una señal de salida con otras características), siempre tomando en cuenta las restricciones del mismo Hardware tal como la velocidad de procesamiento, velocidad de muestreo, tamaño de memoria para almacenar datos, etc.

En este capítulo se parte del análisis de los requerimientos planteados desde el inicio de este proyecto para proceder a mostrar la propuesta de diseño tanto para el Hardware y Software que complementan nuestro sistema de audio y su posterior implementación en torno al DSP TMS320F28377s.

## 5.1. Requerimientos generales

Como el propósito de esta tesis es la de desarrollar un sistema en tiempo real empleando un DSP, se utilizó el audio como señal de entrada debido a su presencia en una gran cantidad de sistemas digitales de uso común, como televisores, computadoras y teléfonos inteligentes. La señal es procesada con diferentes efectos de audio los cuáles se pueden elegir utilizando un dispositivo inteligente, lo que le añade accesibilidad al sistema y la posibilidad de personalizar la señal de audio de salida de acuerdo a cada usuario.

## 5.2. Descripción general

El sistema general, se puede dividir en tres subsistemas, tal como se puede ver en la Figura 5.1:

- El acondicionamiento de la señal de entrada y de salida.
- La adquisición y procesamiento del audio mediante el DSP
- La interfaz del sistema con la aplicación en el teléfono inteligente.

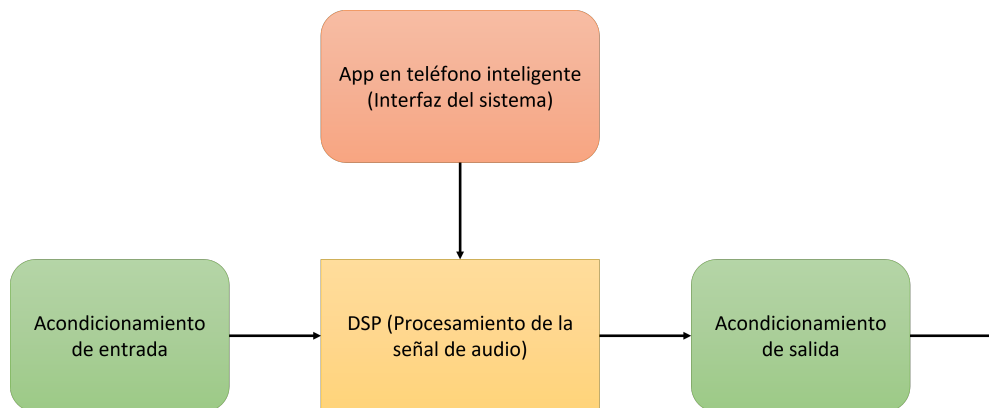


Figura 5.1: Diagrama general del sistema.

Tal como se puede observar en la Figura 5.2, en la etapa inicial del acondicionamiento, se agrega un offset a la señal dado que la misma cuenta con amplitudes positivas y negativas, por lo que es necesario mantenerla en únicamente en la parte positiva para que pueda ser convertida correctamente por el ADC.

La etapa intermedia inicia cuando el DSP adquiere la información de la señal de audio estéreo empleando el ADCB del mismo, en los canales B2 y B4 y empleando un EPWM como inicializador (SOC) del ADC. Se está considerando el espectro en el intervalo de frecuencias de  $20[Hz] - 20,000[Hz]$ , por lo que la frecuencia de muestreo del ADC es de  $44,000[Hz]$  para cumplir con el teorema de Nyquist.

EL módulo bluetooth HC05 se conecta en el puerto SCIB del DSP a una tasa de Bauds de 112,500, cuya función es la de servir como interfaz de comunicación entre el teléfono inteligente y el procesador digital de señales. Permite elegir entres cuatro efectos: Distor-

sión, Retardo de la señal, Reverberación y un Banco de Filtros, con variantes para los tres primeros.

La etapa intermedia termina cuando la señal procesada es transmitida por los convertidores digital-analógico DACA y DACB y es acondicionada, mediante un filtro pasa bajas Sallen-Key, para poder ser conectada a un dispositivo capaz de aprovecharla, como pueden ser bocinas o audífonos alámbricos.

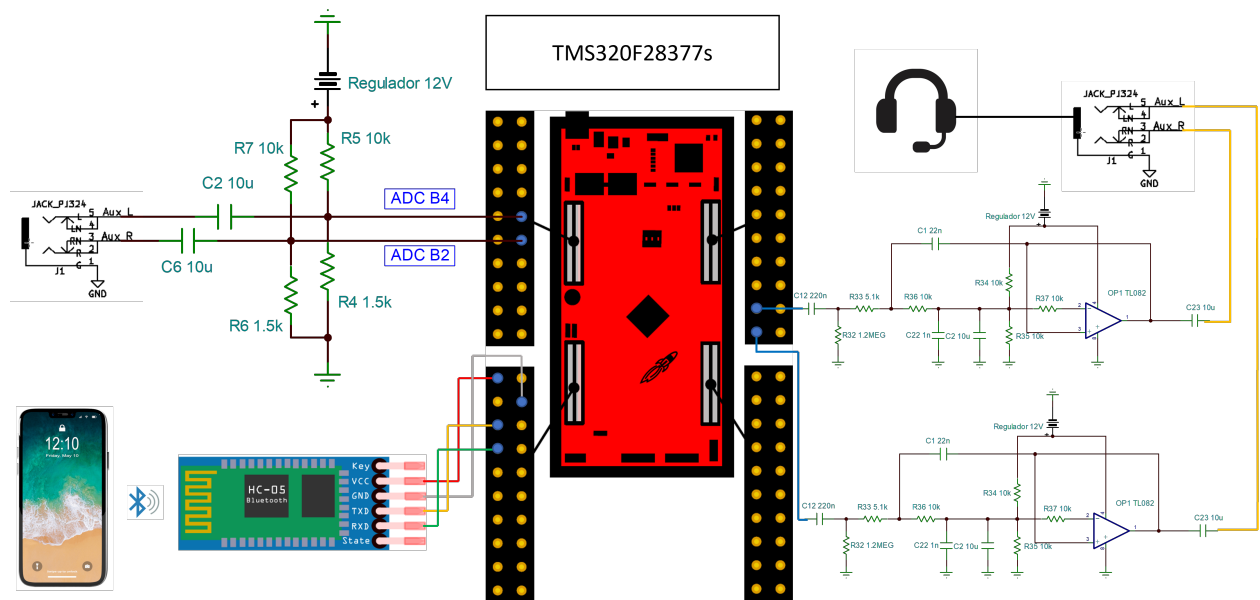


Figura 5.2: Diagrama de conexiones con la tarjeta.

### 5.3. Acondicionamiento de señales Tarjeta de expansión

La tarjeta de expansión, desarrollada en el Laboratorio de Procesamiento Digital de Señales de la UNAM (Figura 5.3), es una serie de circuitos que permiten realizar aplicaciones con audio y voz que se conectan directamente a la tarjeta TMS320F28377s. A grandes rasgos, cuenta con:

- Tres entradas: dos de las cuáles son monocal para la utilización de micrófonos y la restante es una entrada estéreo para el procesamiento de audio.
- Dos interruptores para realizar pruebas de interrupciones.
- Un LED RGB conectado a los EPWM 7, 8 y 9.
- Dos salidas de audio: una salida estéreo y una salida monoaural.

En la realización de esta tesis se utilizaron los siguientes módulos de la tarjeta de expansión: la entrada y salida estéreo para el procesamiento del audio y el LED RGB para identificar cuando se recibe información en el módulo SCI, por lo que a continuación se describirán más a detalle los componentes y el funcionamiento de éstos módulos.

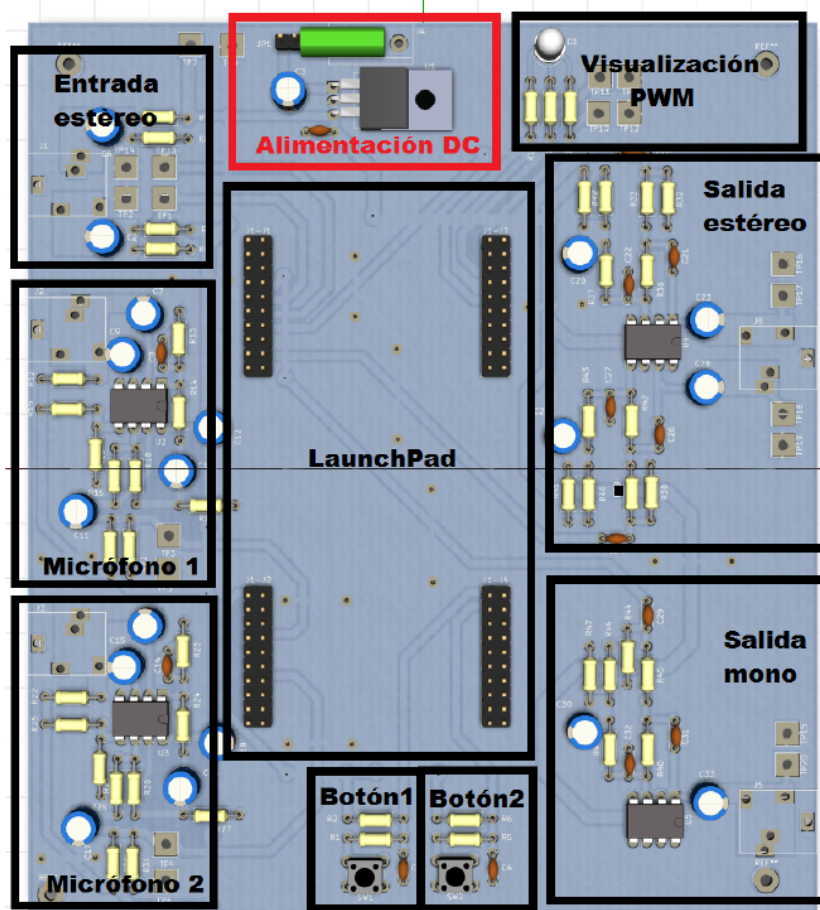


Figura 5.3: Tarjeta de Expansión, tomada de [51].

Para la alimentación de la tarjeta de expansión se recomienda el uso de una fuente de poder de 15[V] debido a que se conecta a un regulador de voltaje de 12[V] que es utilizado para energizar los circuitos analógicos, como se puede ver en la Figura 5.4.

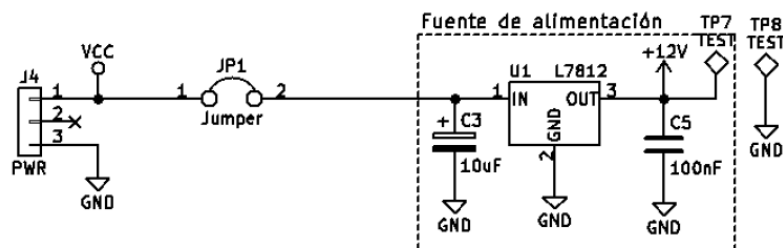


Figura 5.4: Circuito eléctrico de la alimentación de la tarjeta de expansión, tomado de [51].

Como se mencionó anteriormente, el LED RGB está conectado a los EPWM 7, 8 y 9 (Figura 5.5), y dependiendo del ciclo de trabajo, cambiará de color. Esto es utilizado para detectar cuando se recibe información proveniente del módulo SCI, de acuerdo con la tabla 5.1.

Una señal de audio estéreo parte de la idea de simular la audición humana, por lo que utiliza

Tabla 5.1: Asignación de colores para las instrucciones del SCI

EPWM	Color	Significado	Duración
7A	Azul	Cambio de Efecto	1s
8A	Verde	Cambio en la ganancia de algún filtro	0.5s
9A	Rojo	Información del sensor de orientación	0.5s

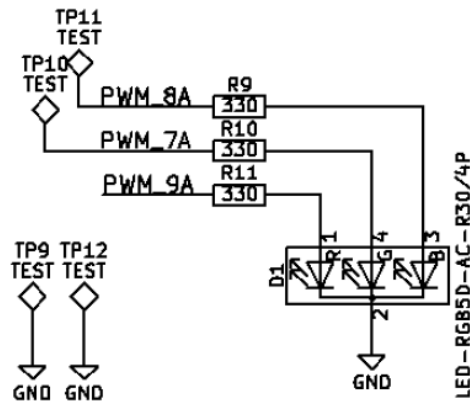


Figura 5.5: Conexión del LED RGB a los EPWM, tomado de [51].

dos canales que son comúnmente identificados como canal izquierdo y derecho. La tarjeta de expansión realiza el acondicionamiento de la señal, que consiste en utilizar un capacitor de acoplamiento para eliminar cualquier posible señal de DC y un divisor de tensión para dar un offset conocido (1.56[V] aproximadamente) a la señal. El offset cumple un papel importante puesto que el módulo ADC del DSP solo puede convertir voltajes positivos y una señal de audio tiene magnitudes tanto positivas como negativas. El circuito eléctrico para el acondicionamiento de cada canal de audio se muestra en la Figura 5.6. La tarjeta de expansión conecta la señal estéreo con el ADCB canal 2 y 4 del DSP.

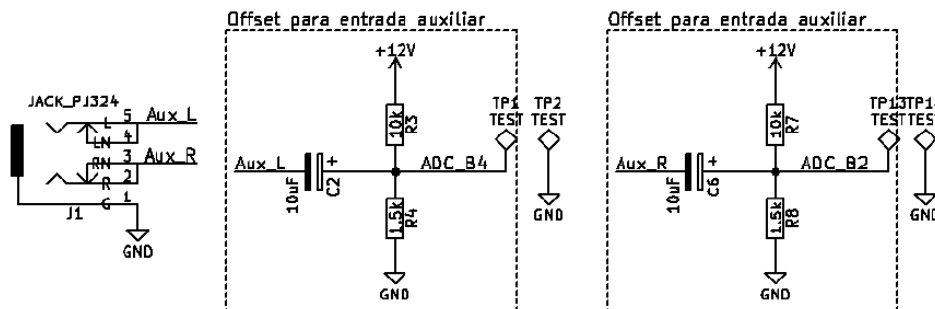


Figura 5.6: Circuito de acondicionamiento de la señal de entrada de audio estéreo, tomado de [51].

Finalmente, el DSP transmite la señal de audio procesada a la salida estéreo a través del DACA y el DACB. El circuito de acondicionamiento consiste un filtro pasa bajas Sallen-Key de segundo orden con una frecuencia de corte de 15[kHz] y un capacitor de acoplamiento. El circuito se muestra en la Figura 5.7

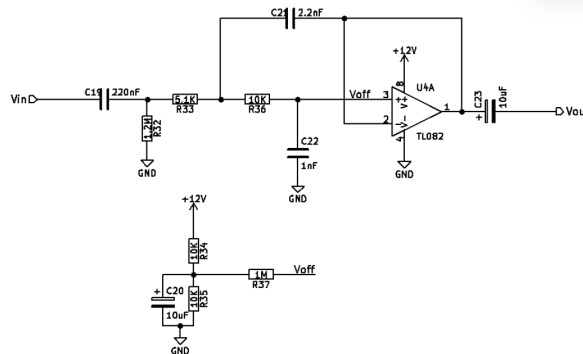


Figura 5.7: Circuito de acondicionamiento de la señal de salida audio estéreo, tomado de [51].

## 5.4. Interfaz del sistema y aplicación móvil

Para la interfaz humano-máquina se desarrolló una aplicación para sistemas operativos Android en la herramienta en la nube *MIT App Inventor*. La aplicación emplea diversos elementos gráficos como botones, barras deslizantes o listas desplegables, además de sensores y módulos del teléfono inteligente para interpretar los efectos que desea el usuario, para finalmente enviarlos a través de una comunicación Bluetooth al módulo HC-05.

De manera general, la aplicación cuenta con dos pantallas: una pantalla de bienvenida donde se muestran los créditos de la elaboración de ésta tesis y la segunda que cuenta con los efectos, sus variantes y etiquetas que indican al usuario la inclinación del teléfono.

### 5.4.1. Pantalla de bienvenida y créditos

Gráficamente, esta pantalla tiene como objetivo el presentar los créditos de la elaboración de la aplicación móvil, además de darle la instrucción al usuario de ocupar su teléfono de manera horizontal. Cuenta con los siguientes elementos:

- Título de la tesis.
- Nombre de los alumnos.
- Instrucción.

Lo que puede ver el usuario se muestra en la Figura 5.8, en el entorno de programación de bloques (Figura 5.9), de ésta primera pantalla cuenta con elementos simples como etiquetas de texto y un temporizador para cambiar el contenido visible pasado un intervalo de tiempo. El botón permite avanzar a la pantalla que contiene los efectos.



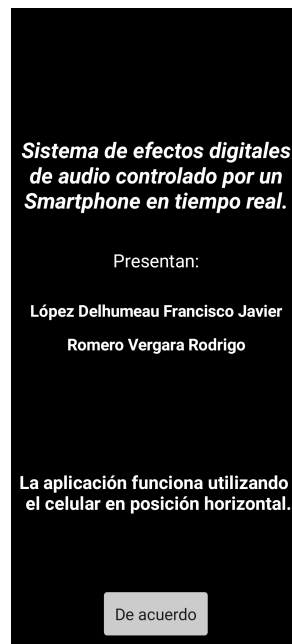


Figura 5.8: Mensaje de bienvenida en la aplicación.

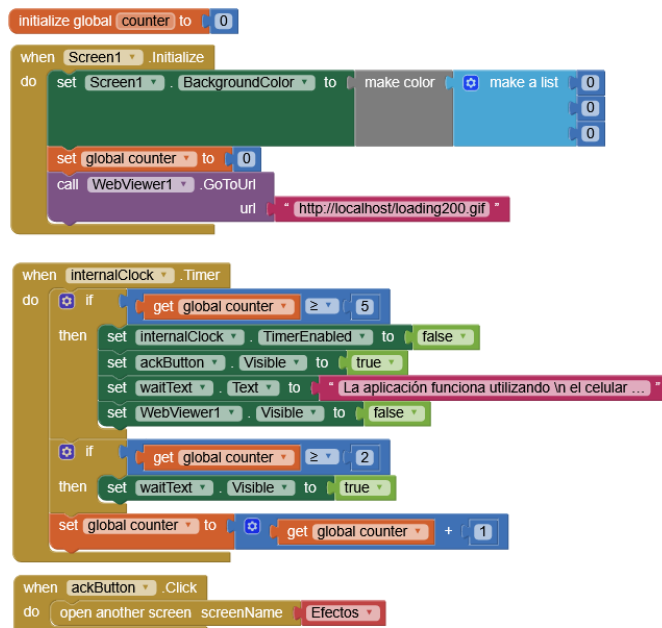


Figura 5.9: Entorno de programación de bloques de la pantalla de Bienvenida.

### 5.4.2. Pantalla de efectos

La pantalla de efectos, que también puede denominarse como la principal, contiene los valores de entrada que permiten al usuario modificar el audio a la salida del sistema. Las diferentes partes de la pantalla serán divididas por secciones para explicar su funcionamiento detalladamente.

## Sección inicial

Ésta sección que puede verse en la Figura 5.10, cuenta con el estado de la conexión con el módulo Bluetooth, que por defecto es igual a "Desconectado" un indicador que sirve para informarle al usuario si el Bluetooth de su teléfono inteligente está activado.

Como paso siguiente, al presionar el botón de "Dispositivos", aparecerán todos los aparatos conectados previamente al teléfono inteligente, aquí se seleccionará el dispositivo "DSPTesis" para poder avanzar a la siguiente sección.

El entorno de bloques cuenta con subrutinas que realizan las siguientes acciones:

- La primera subrutina inicializa los valores de la pantalla: Color de fondo, apariencia de los interruptores (encendido o apagado) y elementos gráficos que solo muestran información, además de ocultar los elementos de otras secciones.
- Genera una lista de los elementos conectados al dispositivo mediante Bluetooth para desplegarla al presionar el botón de "Dispositivos".
- Cuenta con un temporizador que verifica que se esté realizando la conexión con el dispositivo cada segundo, y dependiendo de esto, mostrar los efectos en la pantalla al usuario.



Figura 5.10: Pantalla principal de la aplicación, mostrando los elementos cuando no se ha realizado la conexión con el módulo Bluetooth.

## Sección dos

En la segunda sección se hablará del **sensor de orientación**. El valor del sensor de orientación está definido de forma angular en el intervalo  $[-90^{\circ}, 90^{\circ}]$ , cómo se puede observar en la Figura 5.11, cuya resolución está definida en ésta tesis a  $1^{\circ}$ . Cada vez que se detecta un cambio en el sensor de orientación, el teléfono envía el valor al módulo para ser procesado por el DSP y cambiar la salida de audio de acuerdo a lo establecido en la configuración del paneo del sonido estéreo. En la pantalla se muestra una etiqueta de texto que le indica al usuario el valor actual del sensor en grados.

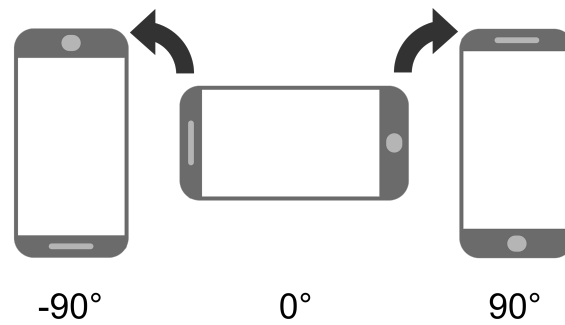


Figura 5.11: Esquema de rotación del sensor de orientación del teléfono.

### Sección tres

La tercera sección comprende la lista desplegable de los efectos de audio y cómo se presentan en pantalla. Todos son mostrados al usuario mediante una etiqueta de texto que permite saber que efecto se está aplicando en el momento (Figura 5.12). Además, para las opciones de **Saturación**, **Reverberación** y **Retardo de la señal** se cuenta con un interruptor capaz de variar las salidas de éstos efectos, como se puede ver en la tabla 5.2:

Tabla 5.2: Tabla de efectos y sus variaciones

Efecto	Inicial	Variación
Saturación	OverDrive	Distorsión
Reverberación	Reverberación 1s	Reverberación 2s
Retardo de la señal	Coro	Eco

El entorno de bloques envía al módulo Bluetooth HC05 la información de que efecto de audio desea aplicar el usuario en el momento que es cambiado en el celular, ya sea un efecto inicial o su variación (ver Tabla 5.3).

### Sección cuatro

La última sección se describe el acomodo de las ganancias del banco de filtros. Se maneja un banco con diez filtros distribuidos en el intervalo de frecuencia de 20 Hz a 20,000 Hz. Se utiliza el elemento gráfico *barras deslizantes* para poder asignar la ganancia, que puede ir de -12dB a 12dB siendo todas independientes (Figura 5.13). Además de contar con un botón de reinicio que asigna todas las ganancias a 0dB.

En el entorno de bloques se envía la información de acuerdo con la tabla 5.3; se eligió un formato de punto fijo  $Q_3$  debido a que se necesitan 4 bits para representar la parte entera

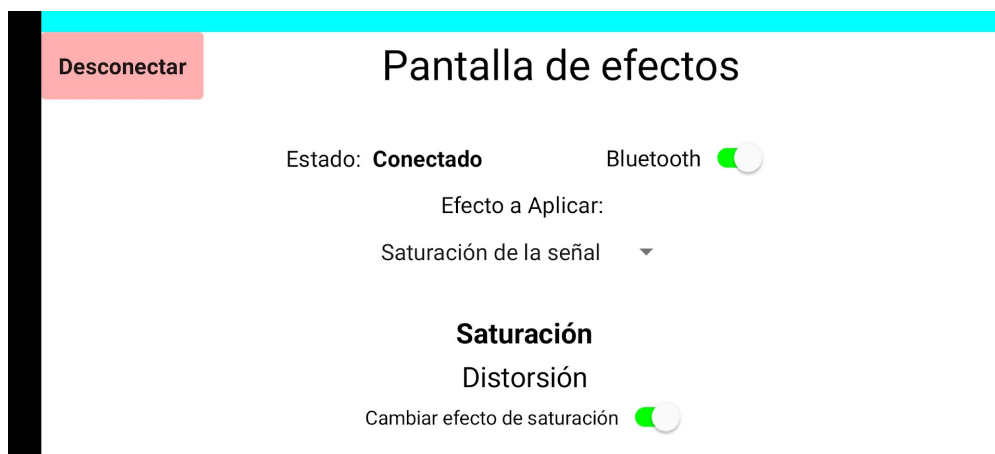


Figura 5.12: Pantalla principal, mostrando los elementos cuando se ha realizado la conexión con el módulo.

más un bit para el signo, lo que nos deja con 3 bits para representar la parte decimal. Cada que ocurre un cambio en cualquiera de las barras, se envía la información a través de la conexión Bluetooth.

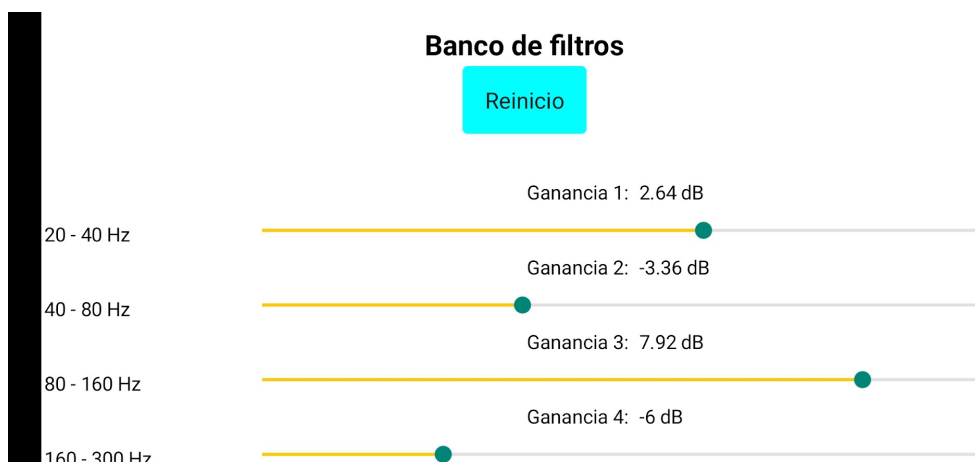


Figura 5.13: Banco de filtros con sus elementos gráficos.

Para finalizar, es importante mencionar la forma en la que se envía cierta información hacia el DSP. Los efectos de audio con sus variaciones, la ganancia de los filtros y los datos del sensor de orientación del teléfono inteligente cuentan con su propia codificación para poder ser aplicadas correctamente. Se envían dos tramas de datos por el protocolo SCI, donde la primera fue denominada como "Instrucción" y la segunda como "Valores". Esto tiene como objetivo el poder brindar al DSP la mayor cantidad de información posible, como la ganancia de los filtros en punto fijo, para tener una sensación más fluida al momento de mover las barras deslizantes. El contenido de cada trama se puede observar en la tabla 5.3.

Tabla 5.3: Codificación de las instrucciones y valores

Instrucción	Identificador	Información en segunda trama
Efecto	32 (00100000 <sub>2</sub> )	Efecto a aplicar + 0 → inicial ; 1 → variación en el bit más significativo (MSb).
Ganancia de filtros	64 (01000000 <sub>2</sub> ) + A Donde A es el índice del filtro seleccionado	Ganancia del filtro seleccionado en decibeles en formato $Q_3$ .
Sensor de orientación	129 (10000001 <sub>2</sub> )	Orientación del teléfono, medido en grados.

## 5.5. Módulo Bluetooth

El módulo Bluetooth HC05 debe ser configurado previamente para cumplir con los requerimientos del proyecto. En la tabla 5.4 se pueden observar los parámetros más importantes del módulo, además de indicar los valores que se modificaron.

Comando	Respuesta
AT+ADDR?	Permite conocer la dirección MAC del módulo Bluetooth, que es usada en la aplicación para acceder a la sección de efectos.
AT+NAME=DSPTesis	Cambiar el nombre del módulo permite identificarlo de manera rápida, además de que permite a la aplicación acceder a la sección de efectos
AT+PSWD?	Es necesario conocer la contraseña para emparejar los dispositivos, por defecto es igual a 1234.

(Continúa en la página siguiente.)

Comando	Respuesta
AT+ROLE=0	Permite modificar el rol del módulo, por defecto está seleccionado como esclavo.
AT+UART=112500,1,0	Se define la tasa de Bauds como 112500, un bit de paro y ningún bit de paridad.

Tabla 5.4: Parámetros específicos del módulo HC05

## 5.6. Programación del DSP

La programación del DSP se realizó utilizando dos lenguajes de programación: C y ensamblador. El lenguaje C es usado para definir los parámetros de los módulos utilizados en el proyecto combinándolo con el C2000Ware (antes controlSUITE), que facilita tanto la comprensión del código como el acceso a los registros de configuración importantes. En el caso del lenguaje ensamblador, se utilizó para optimizar el procesamiento de las operaciones matemáticas y lógicas en la implementación de los efectos de sonido.

### 5.6.1. Convertidor analógico-digital

#### Configuración inicial

Cuando la señal ya fue acondicionada por la tarjeta de expansión, el DSP adquiere la información mediante el módulo ADC. Se utiliza el ADCB en los canales 2 y 4 debido a su posición en la tarjeta de expansión. Su configuración inicial define el preescalador que divide el reloj del módulo entre cuatro, se utiliza la función *AdcSetMode()* que acepta tres valores como parámetros: el convertidor a utilizar, la resolución y el modo en el que se empleará. En el caso de esta tesis, los parámetros son: *AdcSetMode(ADCB, 12 bits, single mode)*. El siguiente paso consiste en precisar cuando se disparará la interrupción en el registro *INTPULSEPOS*, donde la opción elegida fue después de la conversión de la información. Finalmente, para la configuración inicial, se debe encender el módulo del convertidor elegido habilitando el bit del registro *ADCPWDNZ* y esperar aproximadamente 1ms para que el módulo inicie correctamente [48].

#### Configuración del EPWM

Como se mencionó en el capítulo anterior, se permiten utilizar diferentes métodos para disparar un ADC, en este caso, se hablará del EPWM. Si se desea utilizar este módulo como SOC, es necesario definir los siguientes cuatro parámetros, como mínimo:

- Establecer el comparador que disparará el inicio de la conversión, así como la dirección del conteo (ascendente o descendente), esto en el registro *ETSEL.bit.SOCASEL*.

- Determinar la cantidad de eventos necesarios para habilitar el SOC en el registro *ETPS.bit.SOCAPRD*.
- Utilizar los preescaladores del módulo EPWM para ajustar la frecuencia de muestreo, en este caso se usan ambos preescaladores para dividir entre dieciséis el reloj del EPWM y el registro *TBPRD* con una cuenta de 142 para obtener una frecuencia de 44[kHz].

$$f_s = \frac{EPWMCLK}{HSPCLKDIV * CLKDIV * TBPRD} = \frac{100[MHz]}{4 * 4 * 142} = 44.014[kHz]$$

- Habilitar el módulo y el SOC.

### Configuración del SOC

El último paso antes de poder utilizar el módulo es definir los parámetros del SOC y el convertidor. El registro que lleva esta información es el *ADCSOCxCTL* donde la “x” significa el número del SOC y se deben definir: el canal, el disparador y la ventana de adquisición de datos. Dado que se emplean dos canales y que el audio requiere la más alta prioridad, se utilizaron el SOC0 y el SOC1. El canal es definido en el conjunto de bits denominado *CHSEL*. La ventana de adquisición se determina en el conjunto *ACQPS* del mismo registro y su duración se obtiene al multiplicar este valor sumando una unidad por el reloj del sistema; su valor mínimo debe ser igual a 75ns para la resolución de 12 bits y 320ns para la resolución de 16 bits[48].

Por último, se habilitan las banderas de interrupción en el registro *ADCINTSEL1N2.bit.INT1E* y se limpian las mismas para poder generar nuevamente una interrupción.

Cuando el módulo está configurado correctamente, la interrupción del ADCB será disparada cuando se termine de realizar la conversión y el valor resultante se almacenará en el registro *AdcbResultRegs.ADCRESULT0* para el caso del SOC0-canal 2 y *AdcbResultRegs.ADCRESULT1* para el SOC1-canal 4.

### 5.6.2. Convertidor digital-analógico

Para configurar el DAC del DSP es necesario seguir tres instrucciones:

- Establecer el voltaje de referencia, ya sea interno o externo, en el registro *DACCTL.bit.DACREFSEL*. En el caso de este proyecto, se eligió el voltaje interno de referencia igual a 3[V].
- Habilitar la salida del DAC escribiendo el registro *DACOUTEN.bit.DACOUTEN*.
- Inicializar el valor por defecto del DAC en el registro *DACVALS.all*.

Para poder tener una salida en el DAC basta con escribir el valor de 0 a 4096 en el registro *DACVALS.all* de acuerdo con la ecuación 4.8.

### 5.6.3. SCI

La configuración de la comunicación serial asíncrona consta de dos partes fundamentales: la configuración del protocolo y la configuración de los buffer de memoria (FIFOs). Dado que

es necesario conectar externamente el módulo HC05 para poder intercambiar información, se utiliza el SCIB.

El registro que almacena la información de como funciona el protocolo es el *SCICCR*. Para este proyecto se utilizó: un bit de paro, ningún bit de paridad, que no exista retorno de la información y ocho bits de longitud de palabra. La tasa de bauds utiliza dos registros para determinar su valor, el *SCIHBAUD* y *SCILBAUD* y este depende de un reloj derivado de *SYSCLK* llamado *LSPCLK* que, por defecto, es cuatro veces menor. Utilizando la ecuación 4.9, sabiendo que la tasa de bauds requerida es de 115200[Bauds]:

$$SCIBAUD = \frac{25 * 10^6}{115200 * 8} - 1 = 26.12 \quad (5.1)$$

Los registros FIFOS de transmisión y recepción son configuradas en los registros *SCIFFTX* y *SCIFFRX*, respectivamente. Se ajustan parámetros como: habilitación de las interrupciones, nivel de prioridad y algunas mejoras cuando se recibe información.

Cuando se recibe información enviada por Bluetooth y recibida en el modulo HC05, se activa la interrupción del receptor del SCI y se permite leer la información en el registro *SCIRXBUF.all*.

## 5.7. Efectos de sonido

### 5.7.1. Retardos

Como se trató anteriormente, en la sección de la interfaz del sistema, los efectos por retardo implementados en esta tesis abarcan los efectos de coro y eco. Retomando su definición del capítulo 2, un efecto por retardo se puede representar como la suma entre la señal de audio en tiempo presente y la misma señal retrasada por un número de muestras  $N$  y multiplicada por una ganancia  $g$  siempre menor a uno para conservar la audición del sistema (ecuación 2.2). Por lo tanto, como el principio de funcionamiento es el mismo tanto para el eco como para el coro, y recordando de la tabla 2.2 que estos efectos se diferencian dependiendo del valor del intervalo de retardo que pueden adquirir cada uno (10 a 30 milisegundos en caso del coro y más de 50 milisegundos para el eco), se utilizó el mismo algoritmo para su implementación final en el DSP, siempre considerando el número de muestras  $N$  que se retrasa la señal respectivamente para cada efecto al momento de hacer el cálculo.

Para la programación de estos efectos en el DSP se utilizó un buffer de memoria de longitud  $N$  para almacenar los valores en retardo de la señal original, de tal manera que el último valor del buffer es el valor que se suma a la señal de entrada para crear el efecto y este valor es la muestra  $N$  del vector de memoria.

De esta manera, al inicializar el buffer nos aseguramos que éste se encuentre vacío, es decir, rellenando de ceros todos sus valores, debido a que en el primer instante aún no existe señal de audio que hayamos procesado en el DSP. Posteriormente cada que se tiene una nueva lectura de una muestra de audio, se recorren los valores del buffer un lugar en el vector,



quedando la muestra de audio actual en el primer valor del arreglo y desechando el último valor anterior, esto se puede observar mejor en la figura 5.14. En esta figura apreciamos que la salida se compone de la suma de la muestra actual  $x(n)$  con el valor de la posición  $N$  del vector de retardos, representando  $x(n - N)$ , multiplicado por un escalar  $g$ . Cabe destacar que para las primeras  $N - 1$  muestras de audio la suma a la salida será igual que la señal de entrada porque el valor  $N$  del buffer sigue siendo 0, por lo tanto, podemos decir que para ambos casos en los retardos, el efecto empezará a ser audible hasta la muestra  $N$ , aprox. entre los  $10ms$  a los  $> 50ms$  según sea el caso.

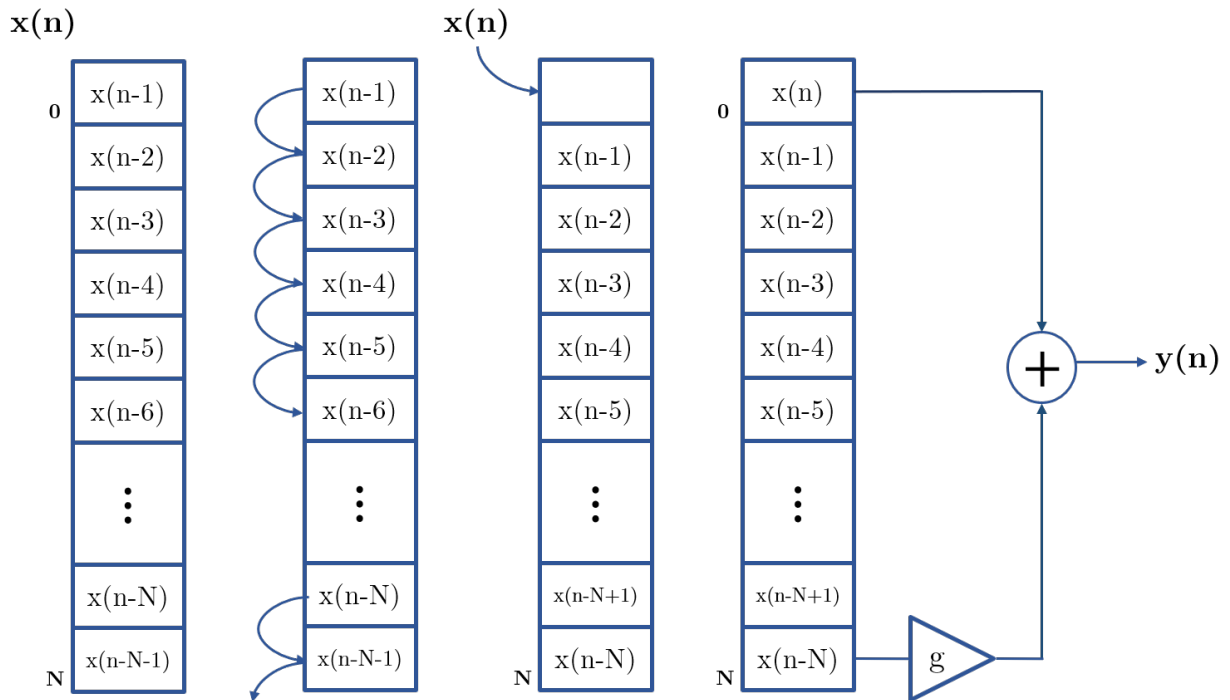


Figura 5.14: Diagrama del funcionamiento de buffer de memoria para los efectos de retardo.

### Cálculo de muestras $N$

Si tomamos en cuenta que la frecuencia es el número de veces que se cambia un fenómeno por unidad de tiempo y tenemos que nuestra frecuencia de muestreo de la señal de audio es igual a  $44,000Hz$ , por lo tanto, tenemos un tratamiento de 44000 muestras por segundo. A partir de esta relación podemos hacer el cálculo del número de muestras necesarias que se necesitan almacenar en el buffer de memoria para los retardos de coro y eco considerando los valores temporales de los retardos acordes a la tabla 2.2. En el caso del efecto de coro, se optó por el mayor valor posible de  $30ms$ , y para el efecto eco el doble del valor mínimo registrado de  $100ms$ , con el objetivo de tener, en ambos casos, la mayor percepción del efecto dentro del rango estándar y propuesto en la industria musical. De tal manera el cálculo del número de muestras  $N$  de presenta en las ecuaciones 5.2 y 5.3

$$\begin{aligned} N_{coros} &= f_s \times \text{segundos} \\ N_{coros} &= 44000Hz \times 0.30s = 1320[\text{muestras}] \end{aligned} \quad (5.2)$$

$$\begin{aligned} N_{eco} &= f_s \times \text{segundos} \\ N_{eco} &= 44000Hz \times 0.100s = 4400[muestras] \end{aligned} \quad (5.3)$$

### Cálculo de ganancia $g$

Podemos considerar que la multiplicación entre retardo y la ganancia  $g$  es equivalente a la división del mismo retardo por una ganancia  $g_2$  tal que ahora  $g_2 > 1$  y  $g_2 = \frac{1}{g}$ . Debido a que la implementación de este efecto se realizó en lenguaje ensamblador, se optó por realizar esta operación de la manera más práctica posible, con las herramientas que el mismo lenguaje de programación y el DSP nos permitieron. De esta manera, para evitar un mayor procesamiento al realizar la multiplicación por una variable definida en lenguaje ensamblador, se utilizó la instrucción del corrimiento aritmético a la derecha realizado y almacenado en el mismo acumulador, el cuál podemos ver como el equivalente a una división dónde el número de corrimientos  $C$  que se realice es equivalente a la división del valor en el acumulador entre  $2^C$ . Acorde a esto, se eligió el número menor de corrimientos posibles (1), con tal de que el efecto fuera perceptible al máximo por el usuario bajo estas circunstancias. En resumen el valor  $g$  es tal que

$$g = \frac{1}{2^1} = 0.5 \quad (5.4)$$

y de esta manera cumplimos con la condición de  $g < 1$  tal que nuestro SLITD es estable.

A partir de estos cálculos y consideraciones, se decidió programar primero los efectos de retardo en *Octave*, para obtener la simulación de nuestras funciones de retardo con respecto a valores de prueba, como pulsos o senoidales, y posteriormente, programarlas en el DSP en lenguaje ensamblador. La lógica de ambos programas siguen a los diagramas de flujo de las figuras 5.15 y 5.16.

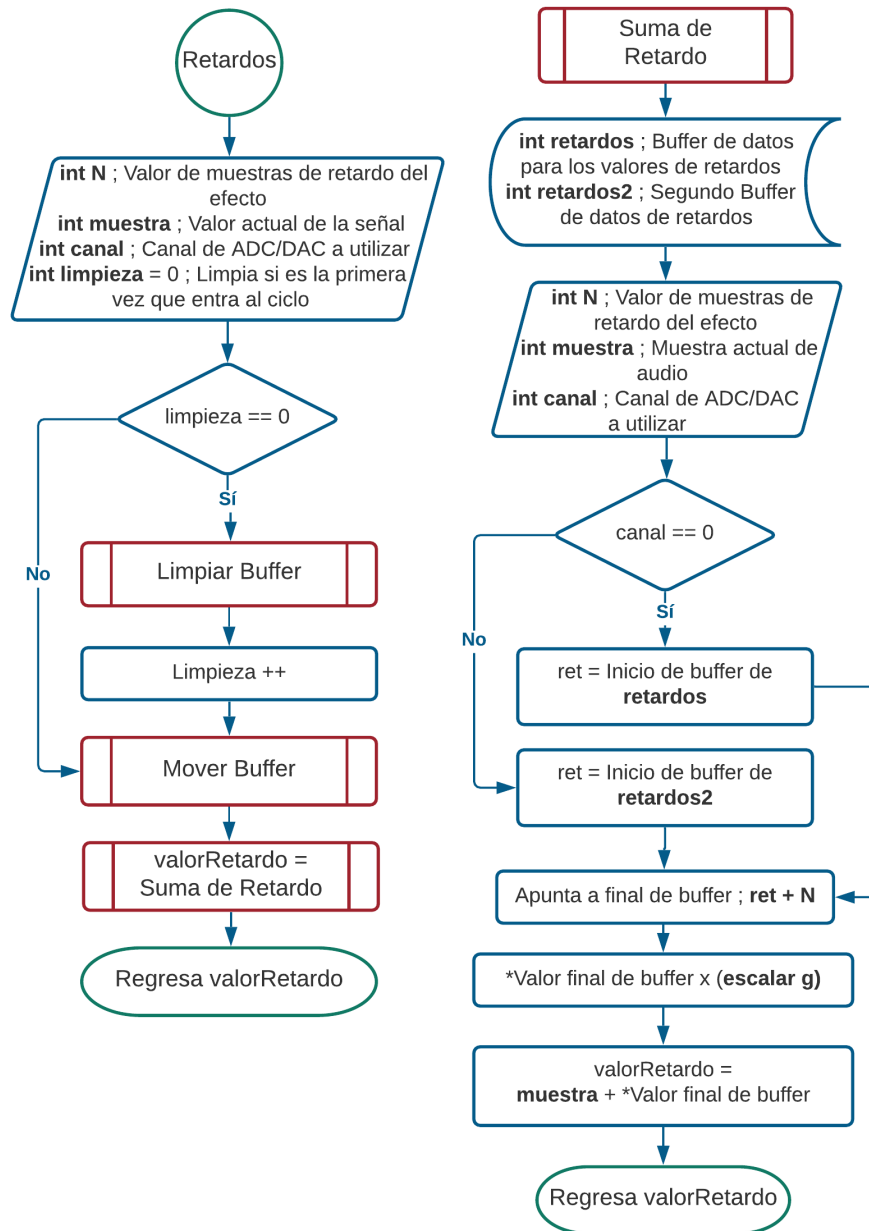


Figura 5.15: Diagrama de flujo de la rutina principal del efecto de retardos y subrutina de suma de retardo.

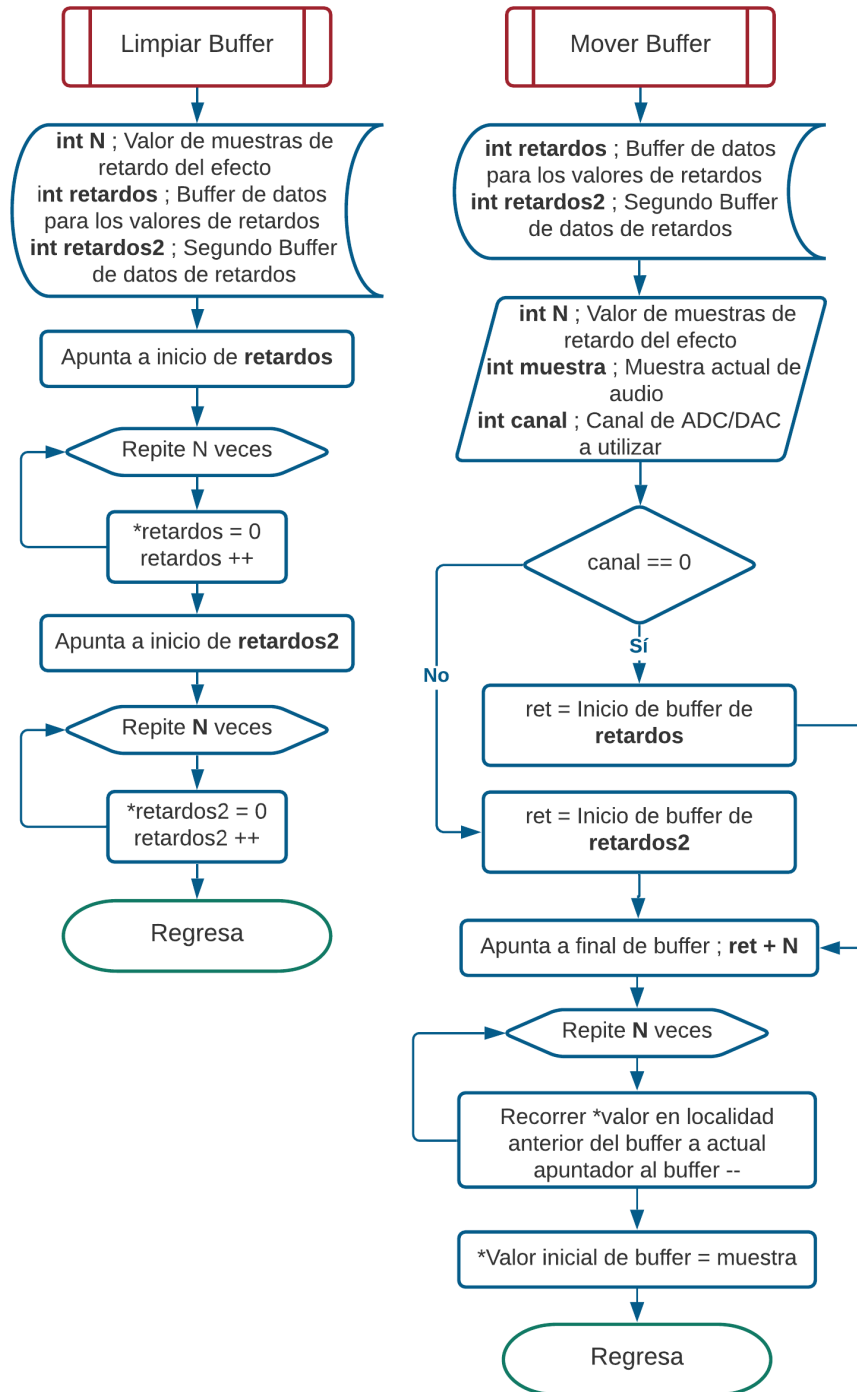


Figura 5.16: Diagrama de flujo de subrutina del efecto de retardos para recorrer valores en buffer y subrutina de limpieza de Buffer.

### 5.7.2. Reverberación

El efecto de reverberación es muy parecido a los efectos por retardo que analizamos anteriormente, puesto que se puede representar como la suma de múltiples señales retardadas en el tiempo, sin embargo, la principal diferencia, es que además de añadir más de un retardo a la señal de salida, cada uno de estos retardos se multiplica por una ganancia propia, la cual está definida por la magnitud de una señal exponencial de base  $a$  en el instante del retardo. Como se explicó en el Capítulo 2, el cálculo de la base  $a$  de la función exponencial está relacionada con el tiempo de reverberación, delimitado por el concepto de piso de ruido y el tiempo de mezcla, y por lo tanto, el número de muestras que se encuentran en este intervalo. Como se observa claramente en la ecuación 2.13, podemos obtener el valor de  $a$  a partir del número de muestras que consideramos para nuestros respectivos efectos dependiendo de su tiempo de reverberación.

Tal como se muestra en la tabla 2.3, el tiempo de reverberación con el que se diseñan las salas de conciertos convencionales oscilan entre 1.4 y 2.2 segundos, -exceptuando el Royal Albert Hall- [10][9], y para locaciones más generales (como salas audiovisuales, salones de clases y cuartos pequeños) el tiempo de reverberación puede ser menor aunque aún perceptible. A partir de esto, se eligieron dos valores aproximados al mínimo y máximo de esos tiempos de reverberación como valores predeterminados en la implementación de estos efectos. Como mínimo se eligió el tiempo de reverberación de un segundo y en el otro extremo el valor de dos segundos.

A diferencia de nuestro diseño en los efectos de retardo dónde tomamos la frecuencia de muestreo de nuestra señal de audio a  $44,000Hz$ , por el gran consumo de espacios de memoria que se utilizan en los efectos de reverberación, decidimos reducir nuestra frecuencia de muestreo tomando ahora como referencia el rango vocal de una persona promedio. Como resultado, para este efecto consideramos una frecuencia de muestreo de  $8,000Hz$ , que es también el número de muestras discretas que obtenemos durante un segundo. Entonces podemos calcular el número máximo dentro de este rango de muestras para nuestros dos casos de efectos de reverberación, tal que:

$$\begin{aligned} N_{max1s} &= 8000Hz \times 1s = 8000[muestras] \\ N_{max2s} &= 8000Hz \times 2s = 16000[muestras] \end{aligned} \tag{5.5}$$

y por consiguiente a partir de los resultados de la ecuación 5.5 utilizamos de nuevo la ecuación 2.13 para calcular la base exponencial relacionada a nuestros dos tiempos de reverberación de los efectos respectivos.

$$\begin{aligned} a_{1s} &= \sqrt[N_{max1s}]{0.001} = \sqrt[8000]{0.001} = 0.999136 \\ a_{2s} &= \sqrt[N_{max2s}]{0.001} = \sqrt[16000]{0.001} = 0.999568 \end{aligned} \tag{5.6}$$

Una vez con estos valores, nos fue posible recrear la función exponencial correspondiente para ambos efectos, sin embargo, al momento de la implementación, este método nos presenta una restricción importante: no se puede realizar la suma de todos los componentes de reverberación debido a que, si tomamos de manera real las distancias diferenciales entre cada pulso de retardo en la señal (magnitud de las reflexiones tardías), el número de retardos

a sumar a la señal de audio de entrada tiende a infinito y por lo tanto, la suma total de retardos también tiende a infinito de manera que nuestro sistema se convierte en un sistema inestable. Es por esto que se buscó la forma de tomar un número finito de retardos para que la suma total con la señal de entrada nos arrojará un resultado viable. Tomando en cuenta esta restricción, se propuso la restricción en el valor del primer retardo a partir de la ecuación 2.15, que dicta que el mínimo valor que puede tomar la primer reflexión tardía para ser percibida por el usuario es del mínimo valor que puede tomar el efecto eco. Por lo tanto, a pesar de que un efecto eco es perceptible desde los  $50ms$ , nosotros decidimos tomar el valor de nuestro efecto eco de  $100ms$  (efecto analizado anteriormente en los efectos de retardo) con el objetivo de poder reutilizar algunas de sus funciones y subrutinas a nivel programación, recordando que cumple con el rango mínimo para pertenecer al rango de reflexiones tardías. Los demás valores de la exponencial, se calcularon en los siguientes múltiplos de nuestro primer retardo ( $200ms$ ,  $300ms$ ,  $400ms$ , ...) hasta llegar a uno o dos segundos dependiendo del efecto.

Cabe destacar que para el caso de la reverberación de dos segundos sólo se sumaron los valores correspondientes a un máximo de 12000 muestras (1.5 segundos de retardo), pero todos los cálculos de los coeficientes se realizaron considerando el piso de reverberación de dos segundos. Esto debido a que a pesar de tener un efecto de reverberación de dos segundos tenemos una restricción por hardware, ya que el número de muestras que podemos guardar en la memoria RAM del DSP es de 45000 valores aproximadamente para ambos canales.

El cálculo de todos los valores de la exponencial por los que se multiplica su respectivo retardo, se obtuvieron aplicando la siguiente ecuación:

$$\begin{aligned} a_{1s}^S &= 0.999136^S \\ a_{2s}^S &= 0.999568^S \end{aligned} \tag{5.7}$$

dónde S es el número de muestras en cada retardo. Además, se obtuvo la suma de estos retardos y el máximo valor de reverberación posible que puede arrojar este efecto (considerando un valor unitario en la muestra de audio de entrada y de todas las muestras retardadas). Los valores obtenidos se muestran en la tabla 5.5.

De esta tabla podemos destacar dos cosas importantes. En *primer instancia*, que el aumento del número de muestras entre cada múltiplo es constante, lo cual desde el punto de vista de programación nos permite hacer ciclos de repetición con el mismo avance entre muestras en cada operación, haciendo más eficiente la implementación en software. En *segundo lugar*, observamos que la suma máxima que obtenemos en ambos efectos no tiene un valor entero mayor a 3. Si recordamos que estos valores los trabajamos en formato de punto fijo, tal que primero normalizamos la señal de entrada para tener un valor entero máximo de la unidad, equivalente al espacio de un bit, y los demás bits se utilizan para la parte decimal con el objetivo de tener la mayor precisión posible en los cálculos, tomar otro bit para la parte entera nos permite tener un valor entero máximo de 3 unidades, lo cual es aún factible. Si la suma total nos arroja un valor mayor a 3, se tendría que tomar otro bit para la parte entera reduciendo así la precisión decimal en las operaciones matemáticas y en el resultado. Por esto toman los valores obtenidos de la tabla 5.5 con ese número de múltiplos para realizar la implementación de ambos efectos nos resulta conveniente en un tema práctico y de eficiencia

Tiempo del retardo	Muestras del retardo ( $S$ )	$a_{1s}^S$	$a_{2s}^S$
100ms	800	0.50119	0.70795
200ms	1600	0.25119	0.50119
300ms	2400	0.12589	0.35481
400ms	3200	0.06310	0.25119
500ms	4000	0.03162	0.17783
600ms	4800	0.01585	0.12589
700ms	5600	0.00794	0.08913
800ms	6400	0.00398	0.06310
900ms	7200	0.00200	0.04467
1s	8000	0.00100	0.03162
1100ms	8800		0.02239
1200ms	9600		0.01585
1300ms	10400		0.01122
1400ms	11200		0.00794
1.5s	12000		0.00562
<b>Suma de amplitud de coeficientes</b>		1.00376	2.41039
<b>Valor máximo posible</b>		2.00376	3.41039

Tabla 5.5: Valores de la función exponencial correspondientes a cada retardo de los efectos de reverberación de uno y dos segundos.

para el tipo de software y hardware que se utiliza en este proyecto. Para la programación de estos efectos en el DSP y de su código en Octave para la simulación, se reutilizó el mismo método que en los efectos de retardos, de la misma manera que se reutilizaron las subrutinas de limpieza del buffer y de corrimiento de datos (figura 5.16). Sin embargo, para la suma de los retardos, se utilizó un segundo buffer en el cual se tienen en forma de vector los valores de la función exponencial respectivos a la posición de las muestras por el cuál se van a multiplicar. Esto se puede observar de mejor manera en la figura 5.17. Las rutinas propias de este efecto se programaron con base en el Diagrama de Flujo de la figura 5.18.

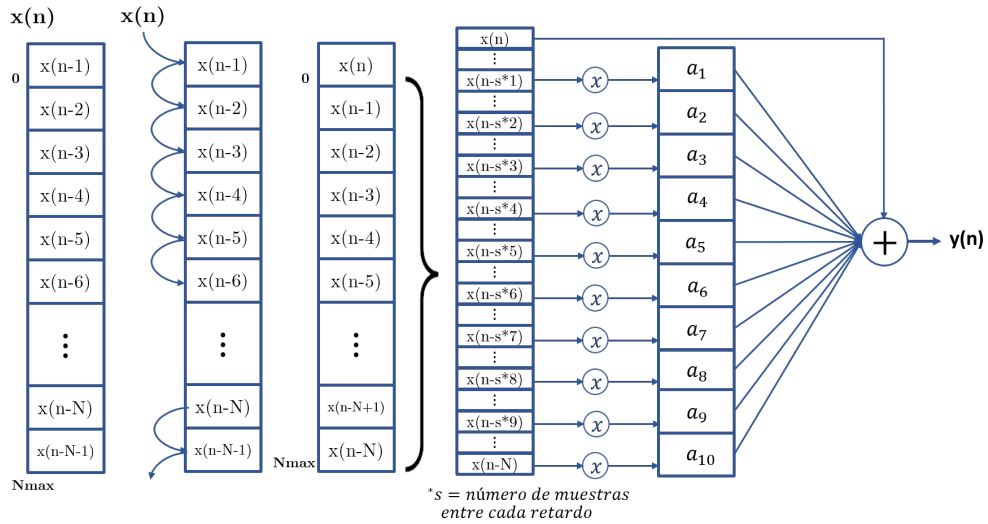


Figura 5.17: Diagrama del funcionamiento de buffer de memoria para los efectos de reverberación.

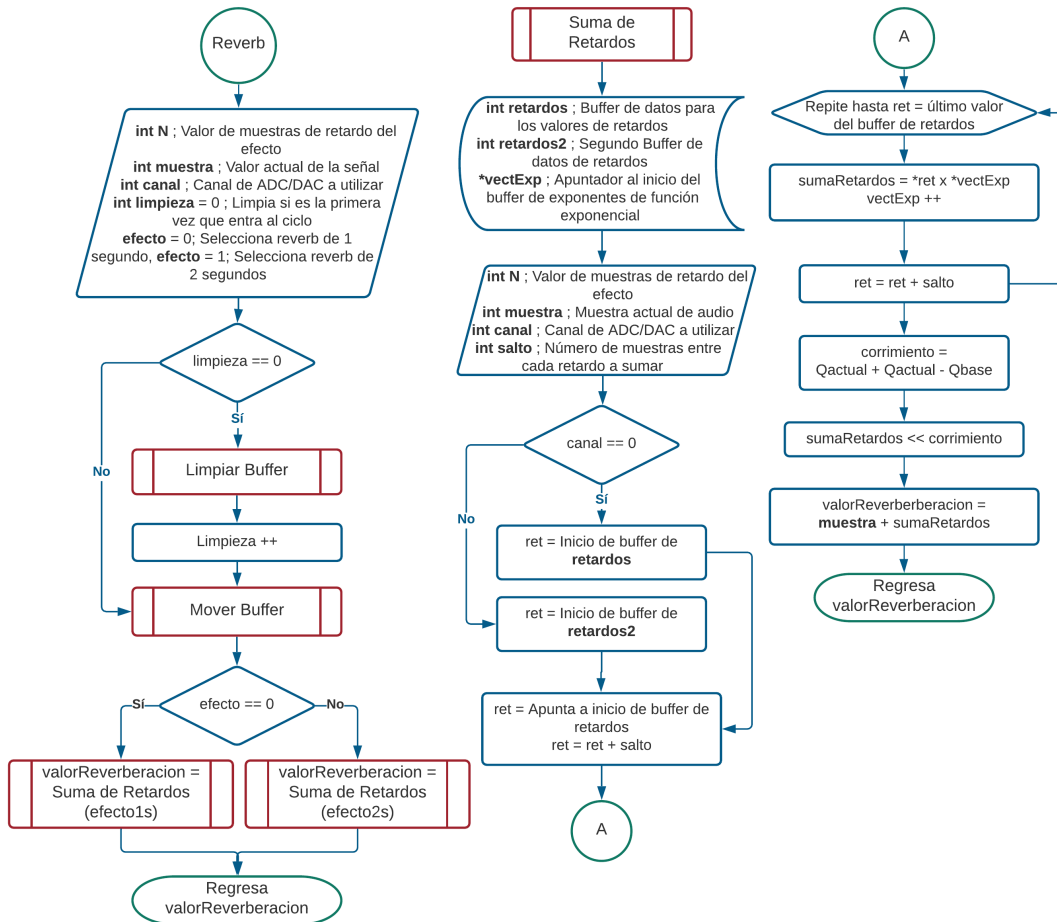


Figura 5.18: Diagrama de flujo de la rutina principal del efecto de reverberación y subrutina de suma de retardos.



### 5.7.3. Distorsión

En el caso de los efectos de distorsión, recordamos que se implementaron dos tipos de efectos: un efecto homónimo (distorsión) y el efecto de saturación. Para ambos casos el diseño partió de sus ecuaciones correspondientes mostradas en el capítulo 2 (ecuación 2.16 y ecuación 2.17 respectivamente). Como podemos observar de las ecuaciones referenciadas, en ambos casos se poseen una serie de operaciones matemáticas de mayor complejidad que las tratadas en efectos anteriores tales como potencias y exponenciales. Es por esto y porque en este tipo de efecto no se requiere la implementación de vectores de memoria para guardar muestras de la señal de audio, que se optó por implementar estos efectos en código C y no en ensamblador, utilizando únicamente la librería "math.h" que nos permite realizar operaciones matemáticas más complejas.

La observación más importante que se tuvo en cuenta durante el diseño de estos efectos, es que en ambas ecuaciones se considera una señal alterna con valores positivos y negativos, lo cual no se cumple al recibir nuestra señal discreta de audio del ADC de nuestro DSP, puesto que como recordamos, el ADC que utilizamos sólo recibe y arroja valores positivos cuyo rango va de 0 a  $2^L$  valores, tal que  $L$  es el número de bits de resolución que posee el ADC (en este caso una resolución de 12 bits). Por lo tanto, para hacer la conversión a una señal alterna, se le aplicó un offset negativo a la entrada del efecto que se calcula como se muestra en la ecuación 5.8

$$\begin{aligned} \text{offset}_{distorsion} &= \frac{ADC_{max}}{2} \\ \text{offset}_{distorsion} &= \frac{2^{12}}{2} = \frac{4096}{2} = 2048 \end{aligned} \tag{5.8}$$

de tal manera que, nuestro nuevo intervalo ahora va de  $-2048$  a  $2048$  y por lo tanto, ya se pudieron aplicar las ecuaciones de manera directa, como se puede observar en los diagramas de flujo para este tipo de efectos. Sin embargo, de igual manera que para la entrada de la señal de audio en el periférico ADC, la salida que se le entrega al DAC sólo abarca valores positivos cuyo intervalo va de 0 a  $2^L$ . Como la resolución del ADC no cambia (sigue siendo de 12 bits), se puede aplicar un offset del mismo valor que en la entrada pero de diferente signo, (ahora positivo) asegurando que nuestro intervalo de valores de la señal a la salida ahora es de 0 a 4096.

En el caso del efecto de distorsión, debido a que durante su procesamiento existen distintos casos de operaciones con respecto al porcentaje del nivel de voltaje que se recibe en la muestra, se decidió normalizar la señal, dividiendo entre el máximo valor absoluto que puede tomar la entrada ( $|[-2048, 2047]| = 2048$ ) de tal manera que el valor máximo de la señal al momento de las comparaciones es unitario, lo que nos permite proponer los intervalos de comparación en números decimales y que resulte en un mejor entendimiento del código sin tener que transformar estos valores al intervalo anterior de la señal. Tanto el programa de simulación, como el archivo en C que se utilizó para la implementación en el DSP se realizaron a partir del diagrama de flujo de la figura 5.19.

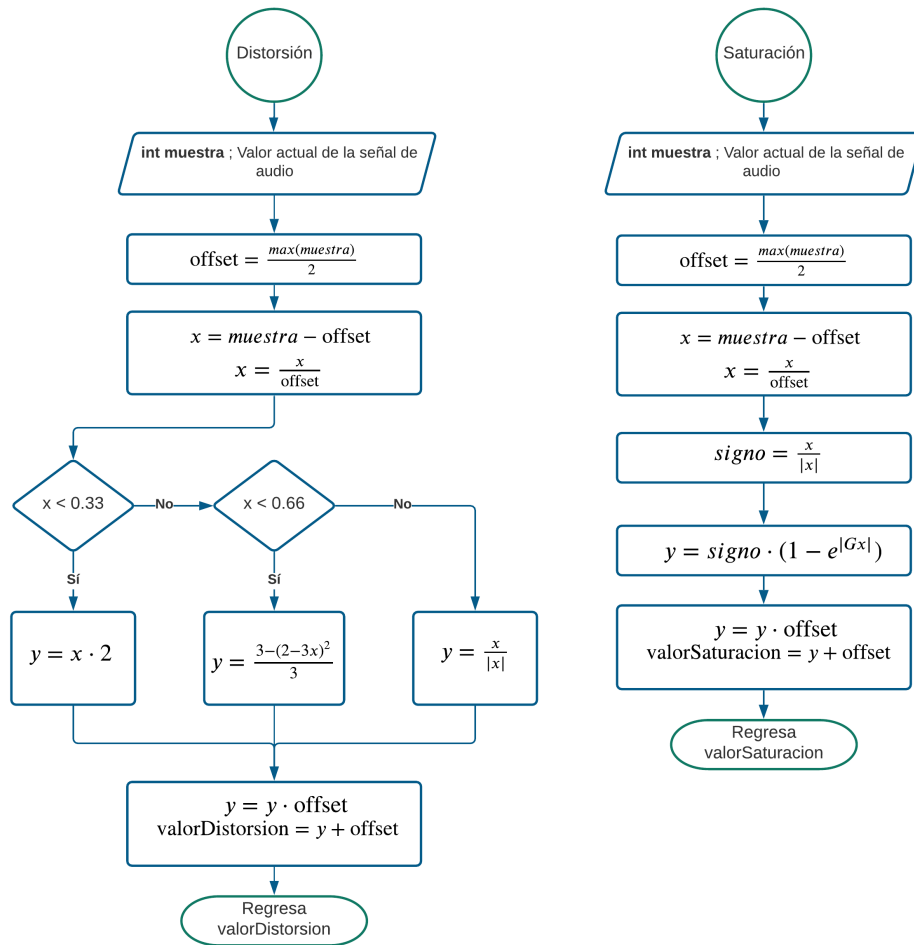


Figura 5.19: Diagrama de flujo de las rutinas de los efectos de Distorsión y Saturación.

### 5.7.4. Filtros digitales

El diseño de los Filtros Digitales (FD) para este proyecto se realizó a partir de la información recopilada en el Capítulo 2 en el apartado de FD, y una metodología de diseño descrita a continuación [22]:

- **Especificación:** se toman las especificaciones para la funcionalidad de nuestro FD a partir de la magnitud y fase que se desea obtener en la respuesta del filtro.
- **Aproximación:** a partir de conceptos matemáticos se selecciona un prototipo donde se aproxime lo más posible al filtro propuesto y se cumplan con las especificaciones del punto anterior.
- **Cálculo:** se generan de manera matemática los coeficientes del FD de acuerdo a las especificaciones y aproximaciones calculadas anteriormente.
- **Simulación:** se toman los coeficientes calculados en el punto anterior y se aplican a un número de muestras con el objetivo de verificar que se cumplan las especificaciones de aproximación dadas.

- **Implementación:** se programa la ecuación en diferencias del tipo de filtro elegido utilizando los coeficientes calculados en los pasos anteriores, ya sea por software o por hardware.

### Especificación y aproximación

Las especificaciones iniciales se toman a partir del objetivo del proyecto descrito en el Capítulo 1. Un banco de filtros de diez bandas que abarca el espectro auditivo que percibe una persona promedio, por lo tanto es coherente que el desarrollo de nuestros FD será a partir de 10 filtros paso banda que abarquen el intervalo de 22Hz a 22kHz en conjunto. Las aproximaciones de las frecuencias de corte para cada una de las bandas se obtienen a partir de las frecuencias estándar para un ecualizador de octava, citadas en la tabla 2.4 y resumidas en la tabla 5.6

Frecuencias de 10 bandas		
Frecuencia Inferior $f_i$ (Hz)	Media geométrica $f_o$ (Hz)	Frecuencia Superior $f_s$ (Hz)
22	31	44
44	63	88
88	125	177
177	250	355
355	500	710
710	1000	1420
1420	2000	2840
2840	4000	5680
5680	8000	11360
11360	16000	22720

Tabla 5.6: Frecuencias utilizadas en el diseño del banco de filtros de 10 bandas a partir del diseño de un ecualizador de una octava.

En cuanto al tipo de FD que se utiliza en este proyecto, podemos recordar los dos tipos de filtros principales que se utilizan para el procesamiento digital: los filtros FIR y los filtros IIR. Retomando los conceptos del Capítulo 2, recordamos que un sistema lineal invariante en el tiempo discreto (SLITD) está definido si se conoce su respuesta al impulso y esta es la principal característica que diferencia a estos dos tipos de filtros, en los filtros FIR la respuesta al impulso  $h(n)$  es finita y definida en un intervalo  $0 \leq n \leq N - 1$  y en los filtros IIR su  $h(n)$  es infinita y definida en el intervalo  $0 \leq n < \infty$ . Aunque esto de manera teórica no nos deja vislumbrar de manera obvia todas las consecuentes diferencias entre estos dos tipos de filtros, se eligió utilizar un filtro tipo IIR por sus ventajas prácticas sobre los filtros FIR que se derivan de sus diferencias teóricas y se citan a continuación:

- Debido a su recursividad, resultante de no poder aplicar una convolución directa a una respuesta de un impulso infinito de manera práctica, se pueden generar filtros IIR de gran pendiente en la banda de transición con pocos coeficientes, a diferencia de un filtro FIR que necesitaría un orden mucho mayor para lograr estas características, ocupando una gran cantidad de memoria en el DSP para el almacenamiento de sus coeficientes.

- Los filtros IIR pueden generar respuestas muy aproximadas al comportamiento de un filtro analógico, lo cual nos permite elegir una aproximación a algún tipo de filtro analógico cuyas características nos son más familiares y hacer uso de herramientas durante el cálculo de sus coeficientes que nos ayuden a implementar métodos de diseño directos o indirectos más complejos con menor dificultad.
- Los filtros IIR al no ser de fase lineal como los filtros FIR, poseen una mejor respuesta de la magnitud en frecuencia.

Una vez que tenemos en cuenta que se eligió implementar nuestro efecto con filtros IIR, y tomando ventaja de que con este tipo de FD podemos obtener una función adecuada de un filtro analógico y transformarlo al dominio discreto, se procedió a elegir el tipo de aproximación que coincidiera con nuestras especificaciones de entre los citados en el Capítulo 2, representados en la figura 3.9.

En este caso en particular se optó por elegir un prototipo Butterworth, debido a que es característico por tener una máxima respuesta plana tanto en la banda de paso como en las bandas suprimidas, sin presencia de lóbulos en ambas bandas como se observan en los otros tipos de filtros analógicos, lo cual podría presentarnos un severo problema si recordamos que nuestra implementación consiste en un gran número de filtros paso banda sumados en cascada. De manera teórica, también se consideraron los filtros tipo Bessel, sin embargo, esta opción se descartó debido a que su pendiente es mucho menos pronunciada que en un filtro Butterworth, tal que en pruebas prácticas no se consiguió reducir la caída a más de  $-6dB$  en las intersecciones de los filtros al utilizar el máximo orden del filtro posible donde el sistema continuaba siendo estable.

Ahora bien, para una aproximación de filtro analógico tipo Butterworth, se requiere regularmente un orden alto para obtener una banda de transición angosta y los diseños que parten con un orden de  $n = 3$  son bastante aceptables, como podemos observar en la figura 5.20. Sin embargo, a consecuencia de una de las mismas características de los filtros IIR que dicta que estos filtros pueden ser inestables, de manera práctica se comprobó que para nuestro sistema, cualquiera de nuestros FD en las bandas propuestas resulta inestable con un orden de valor tres o mayor, por lo tanto, para obtener la mejor caída en la banda de transición, se optó por elegir el máximo orden posible que cumpliera con la estabilidad del sistema. En consecuencia, el orden de nuestro filtro por lo que  $n = 2$  para cada banda de nuestro efecto.

Por último, debemos tomar en cuenta que al implementar un filtro digital paso banda, tenemos como resultado un orden de  $2n$  con respecto al orden  $n$  de diseño elegido anteriormente [52], por lo tanto, tenemos como resultado para nuestros filtros digitales tipo Butterworth un orden de  $2n = 4$ . Con esto verificamos que se cumple con el requerimiento mencionado en el párrafo anterior para un orden aceptable de un filtro analógico.

## Cálculo y Simulación

Como se mencionó anteriormente, existen diferentes tipos de software dedicados al diseño de los coeficientes de los FD. En este proyecto fue utilizado el software *Octave* como ayuda para el diseño de estos filtros tipo Butterworth, utilizando su respectiva función **butter**. Para un mejor entendimiento podemos explicar de manera general las principales características de esta función:

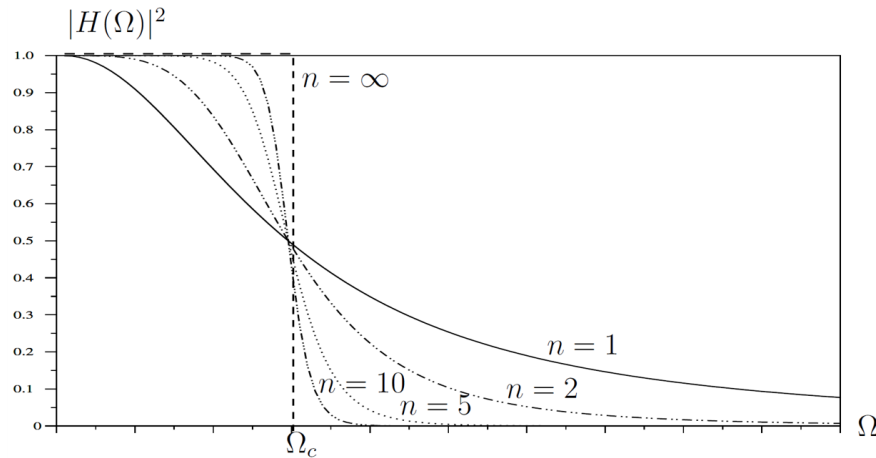


Figura 5.20: Respuesta de función de transferencia de filtro paso bajas tipo Butterworth de distintos tipos de orden  $n$ . Tomado de [22].

### ■ Procesamiento

La función **butter** del software en mención utiliza técnicas y algoritmos para obtener los coeficientes de un FD a partir de la aproximación de una función  $H(s)$  de un filtro analógico aplicando el método de **Transformada Bilineal (TB)** cuidando que los filtros analógicos y su función de transferencia  $H(s)$  y los filtros digitales con FT  $H(z)$ , tengan la misma magnitud de la respuesta en frecuencia.

### ■ Parámetros de entrada

En este caso, una vez que especificamos que nuestros filtros son de tipo paso banda, se proporcionaron dos parámetros de entrada fundamentales:

- $n$ : es el orden analógico del filtro a diseñar, que está relacionado a los límites de ganancia en la banda de paso y la banda suprimida, así como al cálculo de la frecuencia de supresión a partir de la frecuencia de corte.
- $w_n$ : son las frecuencia de corte, y se reciben como un vector de dos elementos, donde el primer elemento corresponde a la frecuencia de corte inferior y el segundo elemento como la frecuencia de corte superior. Estas frecuencias deben estar normalizadas, tomando valores en el intervalo  $[0, 1]$  donde 1 corresponde a la frecuencia de Nyquist.

### ■ Parámetros de salida

La respuesta de esta función consiste en los coeficientes de la función de transferencia del filtro, devueltos como vectores de longitud  $m = 2n + 1$ , siendo  $n$  el orden del FA dado en los parámetros de entrada. La función de transferencia que se entrega se puede expresar como:

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_m z^{-m}}{a_0 + a_1 z^{-1} + \dots + a_m z^{-m}} \quad (5.9)$$

La simulación se realizó en este mismo software, donde a partir de los coeficientes generados para cada filtro, se procedió a obtener y graficar su respuesta en frecuencia, como resulta en la Figura 5.21. En esta figura podemos observar la respuesta para cada uno de nuestros diez filtros, cada uno en el intervalo deseado, cumpliendo de esta manera las especificaciones planteadas en el primer paso del diseño.

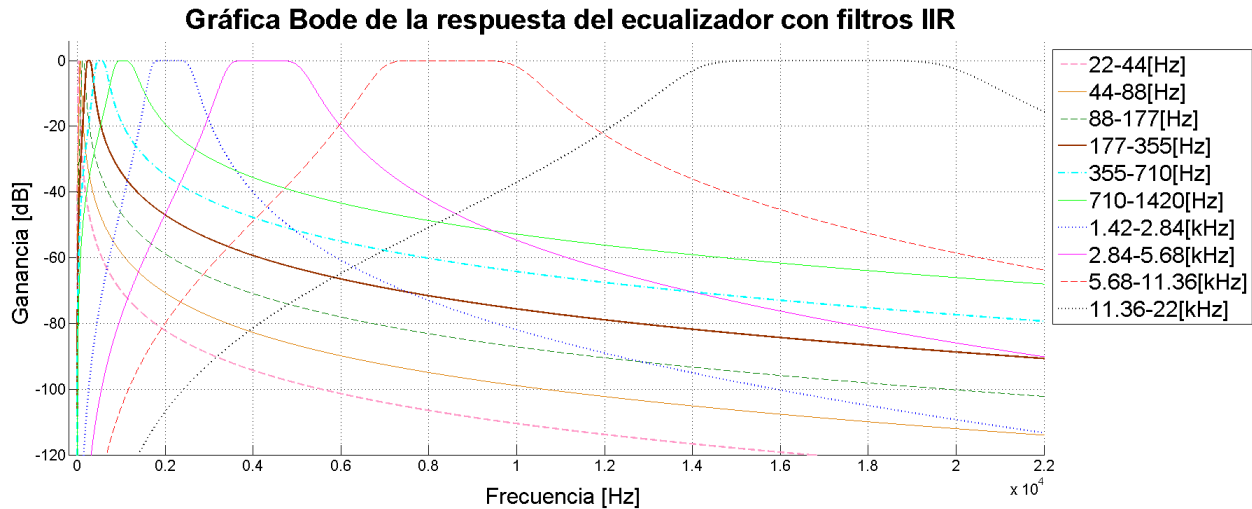


Figura 5.21: Simulación de banco de las 10 bandas de Filtros Digitales con respecto a un ecualizador de una octava.

### Implementación

La forma que se utilizó para hacer la implementación de los filtros es la forma Directa I, que consiste en implementar directamente la ecuación en diferencias 3.18, representada en la figura 5.22 como un diagrama de bloques, sin efectuar ningún cambio.

Aunado a esto, para un mejor entendimiento, la ecuación 3.18 se puede desarrollar de la siguiente manera:

$$y(n) = b_0x(n) + b_1x(n-1) + \dots + b_qx(n-q) - a_1y(n-1) - a_2y(n-2) - \dots - a_p y(n-p) \quad (5.10)$$

de aquí podemos observar que existe una línea de retardo de valor  $q$  para la señal de entrada  $x(n)$  y otra línea de retardo  $p$  para la señal de salida  $y(n)$ , por lo que en total, tendremos  $p + q$  bloques de retardo como se confirma en el diagrama de bloques de la Figura 5.22.

Con base en lo anterior, se observa que la implementación de los filtros IIR se puede ver como la convolución de los coeficientes  $b_m$  con la señal de entrada menos la convolución de la señal de salida retardada con los coeficientes  $a_m$ . Esto a nivel programación nos facilita el proceso debido a que, como lo manejamos anteriormente con el efecto de Reverberación, podemos reutilizar algunas funciones implementadas para efectos anteriores, tales como la limpieza y el corrimiento de los buffers de memoria.

A su vez, se reutiliza el concepto del manejo de diversos buffers como en el efecto de Reverberación, donde ahora tendremos dos buffers para almacenar los valores de retardos de la

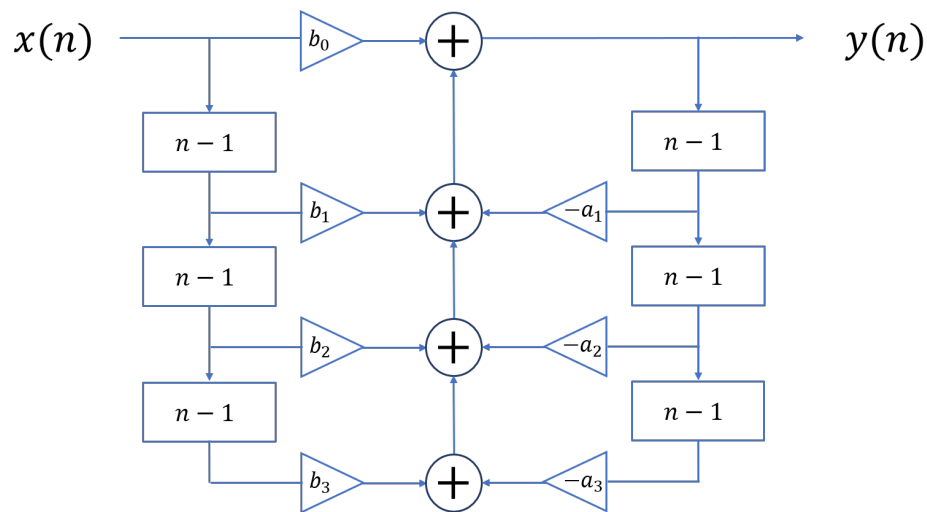


Figura 5.22: Diagrama de bloques de implementación de filtros tipo IIR en forma directa I.

entrada y la salida respectivamente, así como otros dos vectores de memoria para guardar los coeficientes  $b_m$  y  $a_m$ . Esta operación a nivel de localidades de memoria se puede observar mejor en la figura 5.23.

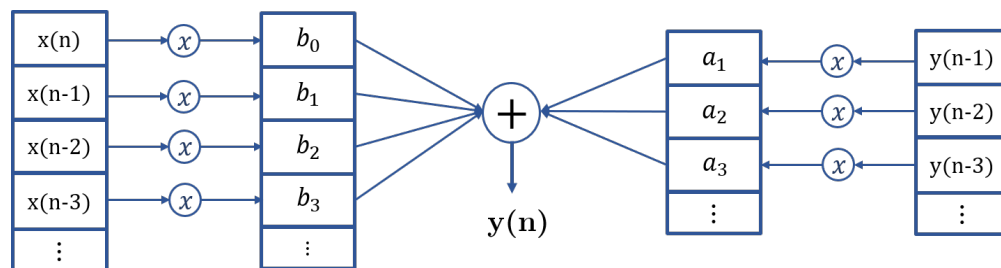


Figura 5.23: Diagrama del funcionamiento de buffer de memoria para la implementación de filtros IIR.

En cuestión de la programación de este efecto en el DSP, como se mencionó anteriormente se tomaron las mismas estructuras de los efectos anteriores para aplicar la convolución de los retardos con los coeficientes así como para recorrer los valores de los vectores de memoria para cada muestra nueva. La única diferencia significativa, fue el uso de los dos buffers de memoria para los coeficientes y dos para los retardos de las entradas y salidas respectivamente. Esto se puede observar de mejor manera en la figura 5.24.

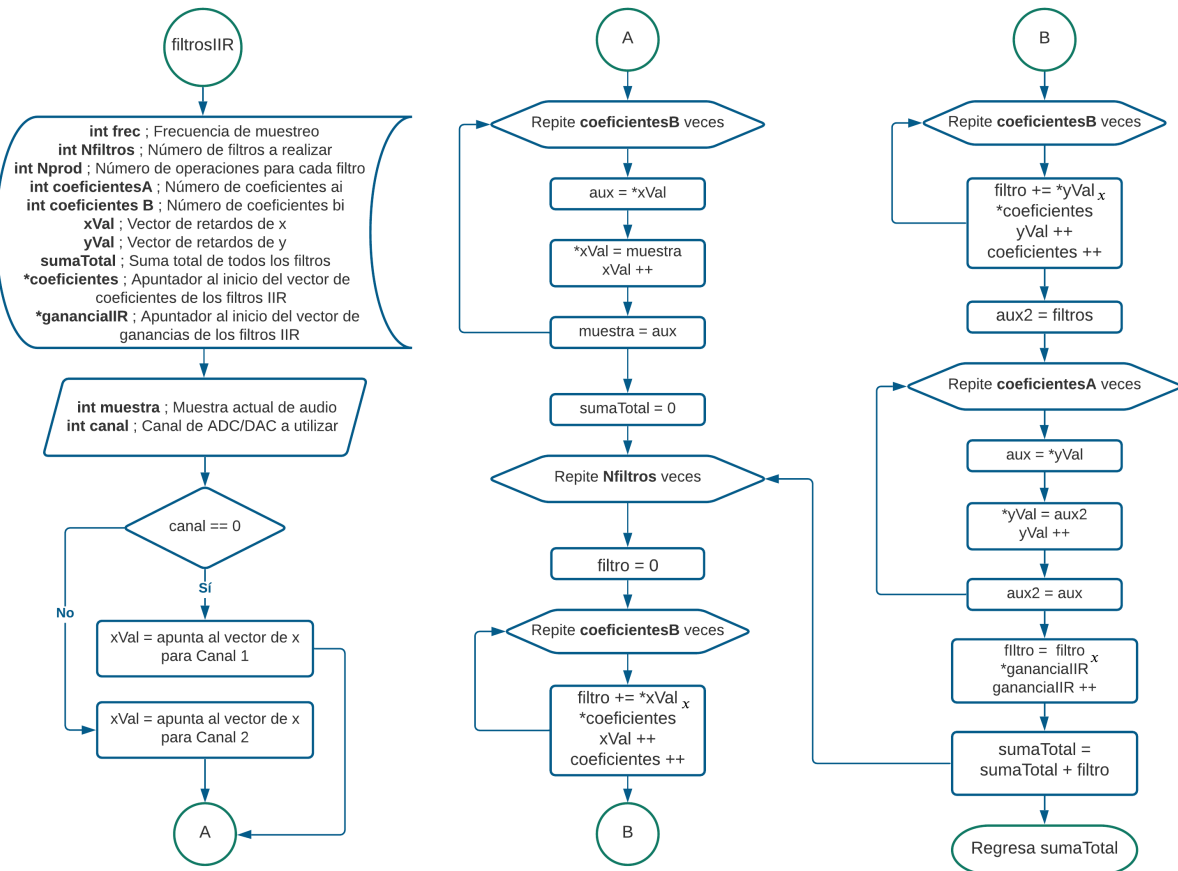


Figura 5.24: Diagrama de flujo de la rutina principal del efecto de filtrado.

### 5.7.5. Paneo de amplitud

Como última aplicación, se generó un paneo de amplitud a la señal de audio en los respectivos canales estéreo antes de proporcionarse los valores al DAC del DSP. Este cambio de amplitud se aplicó en la señal de salida sin importar el efecto seleccionado.

A partir de la teoría revisada en el Capítulo 2, en la sección de las técnicas de paneo de sonido estéreo, se decidió aplicar la técnica del paneo de amplitud con respecto al ángulo de una fuente virtual de sonido. Para la representación de la dirección de la fuente se tomó como referencia el ángulo del giroscopio del celular, que se nos proporciona como dato en grados por medio de nuestra aplicación al DSP, como el ángulo de dirección en la que se encuentra nuestra fuente virtual. Sin embargo, si retomamos las características de la configuración estereofónica estándar, tenemos un límite de  $60^\circ$  entre la mínima y la máxima posición que puede tomar la fuente ( $-30^\circ$  y  $30^\circ$  grados con respecto al eje medio vertical), lo cual no coincide con los grados de libertad que nos genera la rotación del sensor de orientación del teléfono, que va de los  $-90^\circ$  a los  $90^\circ$  dando como resultado un valor de  $180^\circ$  de intervalo total. La manera más sencilla para solucionar este problema, sería limitando el valor en la aplicación del teléfono de  $-30^\circ$  a  $30^\circ$ , sin embargo, al hacer pruebas prácticas, cualquier movimiento brusco dentro de ese intervalo, producía un cambio significativo en los grados



medidos, por lo tanto, el usuario tendría que ser muy cuidadoso al girar el dispositivo para percibir los cambios de manera gradual. Por lo tanto, como la configuración de  $180^\circ$  es mucho más amigable en el manejo del sensor de rotación para el usuario, se resolvió este inconveniente mapeando los grados medidos en el celular con su respectivo valor del intervalo de los  $60^\circ$  requeridos por el paneo, tal que cada unidad de grados medida por el sensor tiene la siguiente equivalencia:

$$1^\circ(\text{sensor de rotación}) = \frac{60^\circ}{180^\circ}(\text{grados de paneo}) \quad (5.11)$$

por lo tanto la equivalencia de los grados se calcula como:

$$\theta = \frac{60^\circ}{180^\circ} \times \phi \quad (5.12)$$

tal que  $\theta$  son los grados de paneo y tienen un intervalo de  $[-60, 60]$  y  $\phi$  son los grados del sensor con un intervalo de  $[-90, 90]$ .

Una vez que tenemos esto en cuenta, se calcularon las ganancias para los dos canales para todos los valores en grados que pudiera arrojar el sensor de rotación. Para ello, apoyándonos en la herramienta de Octave, se tomaron los 180 valores que puede tomar el sensor y una vez mapeados a su respectivo valor como ángulo de paneo, para cada uno se realizó lo siguiente:

- Primero se tomó un ángulo base  $\theta_0$  con valor de  $30^\circ$  (suponiendo que nuestra fuente virtual inicia en el máximo ángulo posible).
- De igual manera, suponemos que el valor inicial de ganancia correspondiente al ángulo base  $\theta_0$  es igual a la unidad.
- Después se realizó la conversión de los ángulos de grados a radianes.
- Posteriormente utilizando el teorema de las tangentes que se representa en la ecuación 2.28, podemos hacer el despeje de la segunda ganancia y proceder a obtener su valor tal como se muestra en la siguiente ecuación:

$$g_2 = \frac{-(g_1 \tan \theta - g_1 \tan \theta_0)}{\tan \theta + \tan \theta_0} \quad (5.13)$$

- Por último, a partir de las leyes de paneo, normalizamos las dos ganancias obtenidas utilizando la suma de los cuadrados de cada ganancia tal como en la ecuación 2.29, de tal manera que obtenemos:

$$g_1 = \frac{g_1}{\sqrt{g_1^2 + g_2^2}} \quad (5.14)$$

$$g_2 = \frac{g_2}{\sqrt{g_1^2 + g_2^2}} \quad (5.15)$$

De esta manera, obtenemos las ganancias para los dos canales que tenemos en nuestro sistema de audio, correspondientes de manera respectiva a los planos de audición izquierdo y derecho respecto al escucha.

Con respecto a la implementación en el DSP, todos los valores de las ganancias para cada ángulo que puede tomar el sensor de rotación, se guardaron en un archivo de cabecera y se importó en el código como un vector de valores con su respectivo espacio en la memoria del DSP, así, cada que se va a retornar un valor de la señal de audio al DAC, primero se lee el valor del ángulo que nos arroja en ese momento el sensor de rotación del celular, guardado en una variable en el DSP, y con respecto a esto, se multiplica el valor de la señal de los canales 1 y 2 -izquierda y derecha-, con sus respectivas ganancias de la posición que corresponde del vector de memoria de las ganancias de paneo (los índices de los vectores de memoria son directamente proporcionales a los valores del rango del sensor). De esta manera, se evita rehacer el cálculo de las ganancias con respecto a los ángulos de entrada (cuyo rango siempre será constante) cada que se procese un valor de nuestra señal de audio, ahorrando tiempo en el procesamiento y sobretodo memoria de código en el DSP.

## 5.8. Resumen

En este capítulo se presentó un breve análisis de cada especificación planteada en el capítulo 1 del presente documento y de la documentación e información recopilada en los capítulos 2, 3 y 4. A partir de ello se realizó una propuesta de diseño acorde a las características de la señal de audio con la que se planteó para trabajar y del DSP que se desempeña como la herramienta en donde se procesa nuestra señal de manera digital.

En el caso del software, el diseño se planteó desde el análisis realizado en los capítulos 2 y 3, acerca de los efectos de audio, siempre tomando en cuenta el objetivo del proyecto -procesamiento de audio en tiempo real-, por lo que se realizaron propuestas enfocadas en poseer una mayor eficiencia en el tiempo de procesamiento de la señal a partir de las características intrínsecas del DSP tales como el funcionamiento de sus periféricos, el número de localidades de memoria disponibles para utilizar o los lenguajes de programación permitidos por el procesador de señales.



# Capítulo 6

## Pruebas y Resultados

El análisis de los resultados que se muestran en la presente tesis se realizaron observando la señal eléctrica a la salida del circuito de acondicionamiento conectado al DAC del DSP, con ayuda de un osciloscopio e ingresando al sistema señales cuyo comportamiento es conocido, como lo son las senoidales o un tren de pulsos.

Para el análisis de señales eléctricas se utilizan dos dominios: el tiempo y la frecuencia. En el dominio del tiempo se analizan las señales correspondientes a los efectos de **reverberación y retardo** de la señal, puesto que idealmente éstos no alteran las componentes espectrales de la señal de entrada. Por otro lado, en el dominio de la frecuencia se observaron los efectos de **distorsión y el banco de filtros** utilizando la Espectro de la señal de salida (FFT) para observar su comportamiento en el sistema. Todas las pruebas fueron simuladas en el programa Octave y con el mismo se diseñaron las señales de entrada para ingresarlas al sistema a través de la tarjeta de expansión para después ser procesadas por el DSP. Se analizó solo un canal a la salida sin bocinas observando la respuesta con un osciloscopio.

## 6.1. Análisis de la señal en el dominio del Tiempo

### 6.1.1. Análisis de efectos de coro y eco.

#### Resultados de la simulación

Para la simulación del efecto coro y eco, se realizó un archivo en Octave, donde se programó una señal de un segundo de duración con dos pulsos de un ancho de 10ms cada uno y con una separación de 0.5 segundos entre sí. Posteriormente se aplicaron los efectos de coro y eco los cuales fueron programados conforme al mismo diagrama de flujo de las figuras 5.15 y 5.16, utilizando también un buffer en forma de vector de memoria y realizando el recorrido de los datos del vector en cada ciclo para la posterior suma del último valor y la señal de entrada en cada valor de salida. El resultado de esta simulación se presenta en la figura 6.1.

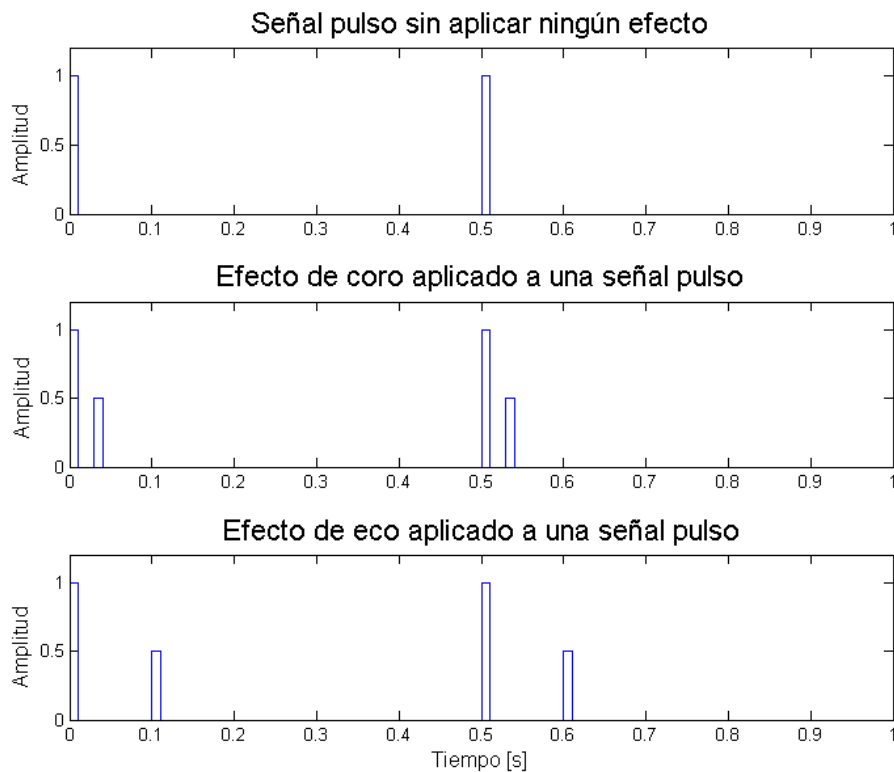


Figura 6.1: Simulación en Octave de los efectos eco y coro aplicados a señal de pulsos.

#### Resultados experimentales

Experimentalmente, se utilizó la señal tren de pulsos a la que se hace referencia en la sección anterior y fue ingresada al sistema para poder observar su comportamiento sin bocinas o audífonos. Los resultados de la aplicación del eco y reverberación se presentan en la figura 6.2.

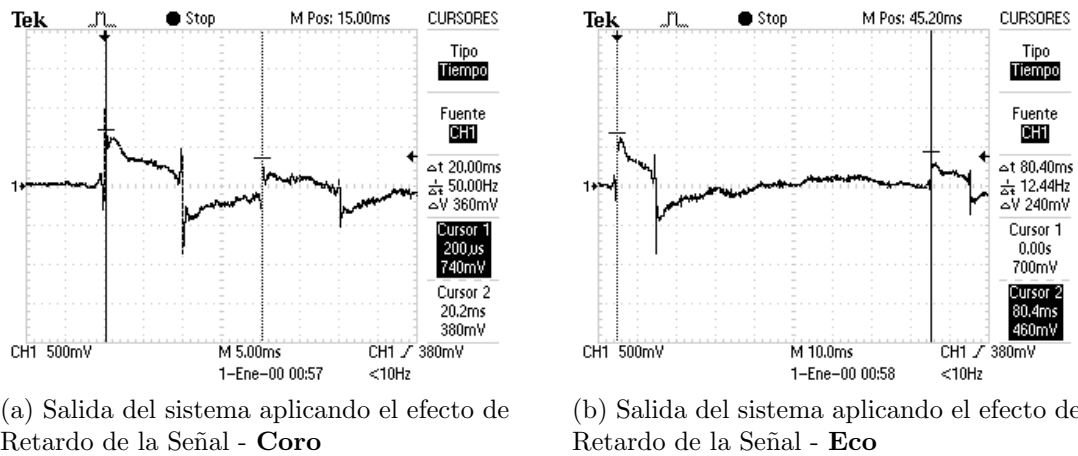


Figura 6.2: Aplicación del efecto de Retardo de la Señal

### Análisis de resultados

Por la simulación se puede confirmar que el algoritmo desarrollado en esta tesis funciona correctamente. Sin embargo, experimentalmente se observa que es necesaria una mayor velocidad del DSP para tener una mejor precisión. Se hicieron pruebas utilizando una velocidad base de 100[Mhz] y de 200[MHz] (la máxima posible en este dispositivo). Es importante mencionar que el movimiento del buffer depende de las muestras entrantes al efecto. Considerando esto, para el efecto de eco se perdían la mitad de las muestras y, por ende, la mitad de los movimientos, por lo que la frecuencia de muestreo real era de 22[Khz] y el retardo real de 60[ms] aproximadamente. Para el *efecto de coro*, la cantidad de muestras que se perdían es de tres por cada muestra entrante, dando una frecuencia de muestreo de 8800[Hz] y un retardo de 0.5[s]. Es por esto que se modificaron los tamaños de los buffer de memoria para cada efecto, con tal de mejorar el tiempo de ejecución en el efecto con respecto al tiempo entre la toma de muestras.

En el caso del **efecto de coro**, el nuevo buffer que se utilizó fue de 800 muestras. El tiempo que toma en procesar una muestra de este efecto es ahora de aproximadamente 20[ $\mu$ s] lo cual se encuentra justo por debajo del intervalo de los 22.72[ $\mu$ s] que existen entre cada toma de muestra. Esto nos da un resultado de un retardo de exactamente 20[ms] como se muestra en la figura 6.2(a) manteniendo la frecuencia de muestreo a 44[kHz], con lo que aún se cumple con el intervalo de tiempos que se define de manera teórica para el efecto coro que es de 10ms hasta 20ms como se hizo referencia en la tabla 2.2.

En el caso del **efecto de eco**, el buffer que se utilizó fue de 1600 muestras. A diferencia de el efecto coro, aquí hubo un sobrepaso del tiempo en el que tarda en procesar el efecto con respecto a la toma de una nueva muestra, por lo que ahora se perdía una muestra de cada dos, y por tanto se trabajó a una frecuencia de muestreo real de 22[kHz]. En este caso, el retardo con 1600 muestras en lugar de ser de 40ms es de aproximadamente el doble: 80.4ms tal como se aprecia en la figura 6.2(b), de dónde podemos comprobar que a partir de la referencia de la tabla 2.2 también se cumple con la condición teórica para un efecto de coro que estipula que el retardo del eco debe ser mayor a 50ms.

### 6.1.2. Análisis de efectos de reverberación.

#### Resultados de la simulación

De igual manera se realizó un archivo de simulación en Octave para el *efecto de reverberación* de *un segundo* y de *dos segundos*, utilizando la misma función para crear la señal de tren de pulsos que se utilizó en los efectos de eco y coro para la prueba. En este caso, el ancho del pulso es de 10ms, pero el periodo en estas señales es ahora de un segundo y dos segundos respectivamente. Para esta simulación también se utilizó un buffer de memoria en el que se recorrieron los datos en cada ciclo y se realizó la suma respectiva de cada valor y su multiplicación por la ganancia de la función exponencial de reverberación con el valor de entrada. Los valores de las funciones exponenciales se calcularon directamente en el archivo de Octave como se describió en el Capítulo 5, y fueron los mismos valores que se guardaron en un archivo como encabezados para su importación en el código del DSP. Las funciones dónde se realizan los cálculos para obtener los valores de las funciones de reverberación están basados en los diagramas de flujo de la figura 5.18 y el resultado de estas funciones se presenta en la figura 6.3.

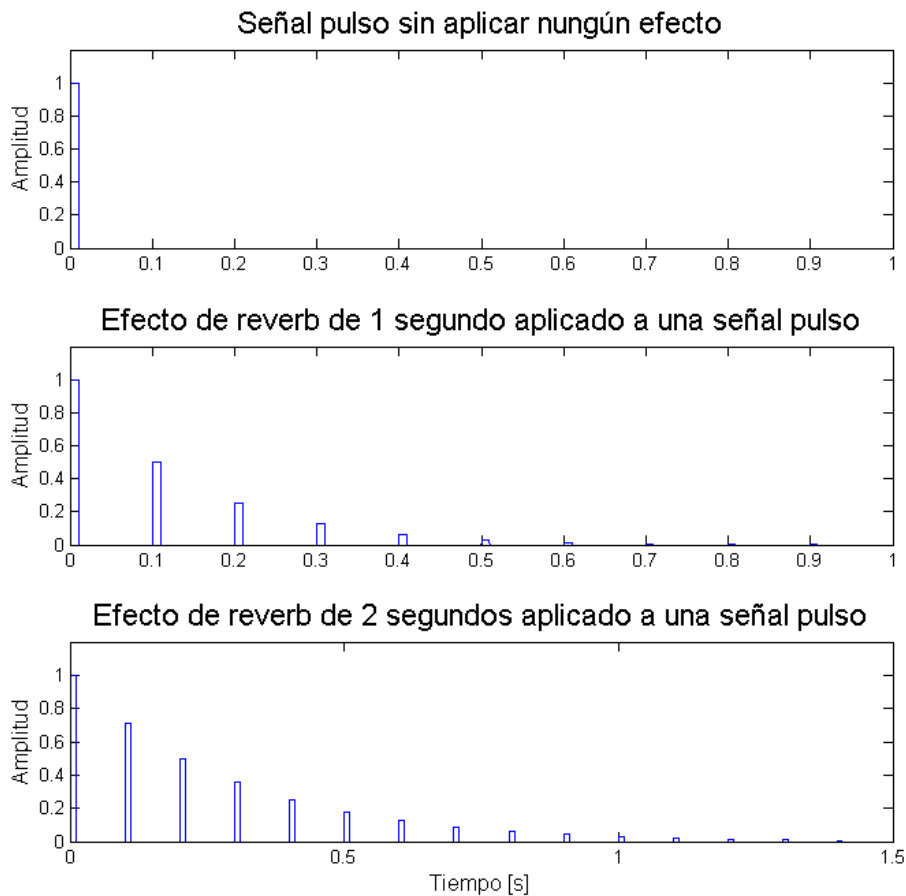
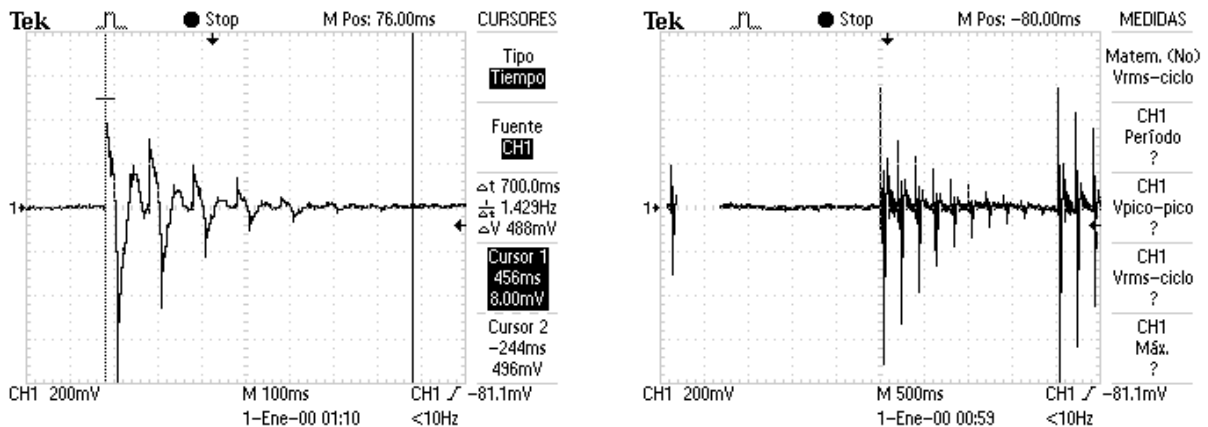


Figura 6.3: Simulación en Octave de los efectos de reverberación aplicados a señal de pulso.



## Resultados experimentales

Para analizar la respuesta del **efecto de reverberación** se generó un tren de pulsos en Octave que después fue ingresado a manera de audio al sistema, para poder observar su respuesta. Se eligió la frecuencia de muestreo a 8000[Hz] únicamente para este efecto debido a la cantidad de muestras que es necesario mover en los buffers; por lo que la reverberación, en cualquiera de sus variantes, funciona mejor para procesamiento de voz. Los resultados experimentales se pueden observar en la figura 6.4.



(a) Salida del sistema aplicando el efecto de reverberación a un segundo.

(b) Salida del sistema aplicando el efecto de reverberación a dos segundos.

Figura 6.4: Aplicación del efecto de Reverberación.

## Análisis de resultados

El efecto de reverberación presenta las mismas implicaciones que el efecto de retardo de la señal pero a una escala mayor, debido a que se necesitan más muestras durante la aplicación de los efectos. Se logró identificar el intervalo en donde la duración del efecto se duplicaba, debido a los tiempos de procesamiento, por lo que se decidió utilizar la variante de reverberación de un segundo por debajo de este límite y la reverberación a dos segundos por encima de éste (al igual que en los efectos de retardo), realizando los cálculos pertinentes. De esta manera, la cantidad de muestras que se usan en el efecto de menor duración, es de 5600 muestras que, a una frecuencia de muestreo de 8000[Hz], se convierte en una reverberación de duración de 0.7[s]. Para el efecto de mayor duración se usan 8000 muestras para su aplicación; debido a que el tiempo que le toma al sistema procesar esta cantidad de información, la frecuencia se ve reducida a la mitad, por lo tanto, la duración del efecto se ve duplicada, obteniendo 2[s] de reverberación. A pesar de que en ambos casos no se suman el número de retardos teóricamente considerados, esto no afecta la percepción de estos efectos en el momento de que el usuario escucha el resultado.

Como se observa en la figura 6.4 tanto para los casos (a) y (b), tenemos que nuestros resultados experimentales cumplen con las dos características principales de la reverberación: la presencia de los retardos tardíos y su amplitud proporcional a una señal exponencial. Cabe mencionar, que si bien es cierto que los cambios del tamaño de los buffer en cada caso de los dos efectos repercuten en el número de retardos que se suman a la señal de entrada, o

en el caso de la reverberación de un segundo, que se trunca el efecto y el último retardo que se suma es a los 0.7 segundos, esto no quiere decir que este efecto es ahora de 0.7 segundos de reverberación, puesto que los valores escalares por los que se multiplican dichos retardos siguen siendo los valores calculados de manera teórica con respecto a las características de la exponencial y el piso de reverberación de uno y dos segundos, por lo que en ambos casos, teniendo esto en cuenta, siguen cumpliendo con las características de un efecto de reverberación de uno y dos segundos respectivamente.

## 6.2. Análisis de la señal en el dominio de la Frecuencia

### 6.2.1. Análisis de efectos de distorsión.

#### Resultados de la simulación

Para el caso de estos efectos se realizó un archivo en Octave donde primero se programó una señal senoidal discreta de  $1\text{KHz}$  de frecuencia. Después de esto, se procedió a iterar para cada uno de sus valores y se realizaron las mismas operaciones que se realizan en tiempo real en el DSP y que están representadas en el diagrama de flujo de la figura 5.19, tanto para el efecto de distorsión como el de saturación. Luego se obtuvieron dos señales resultantes, una del efecto de saturación y otra del efecto de distorsión, las cuales se muestran en la figura 6.5.

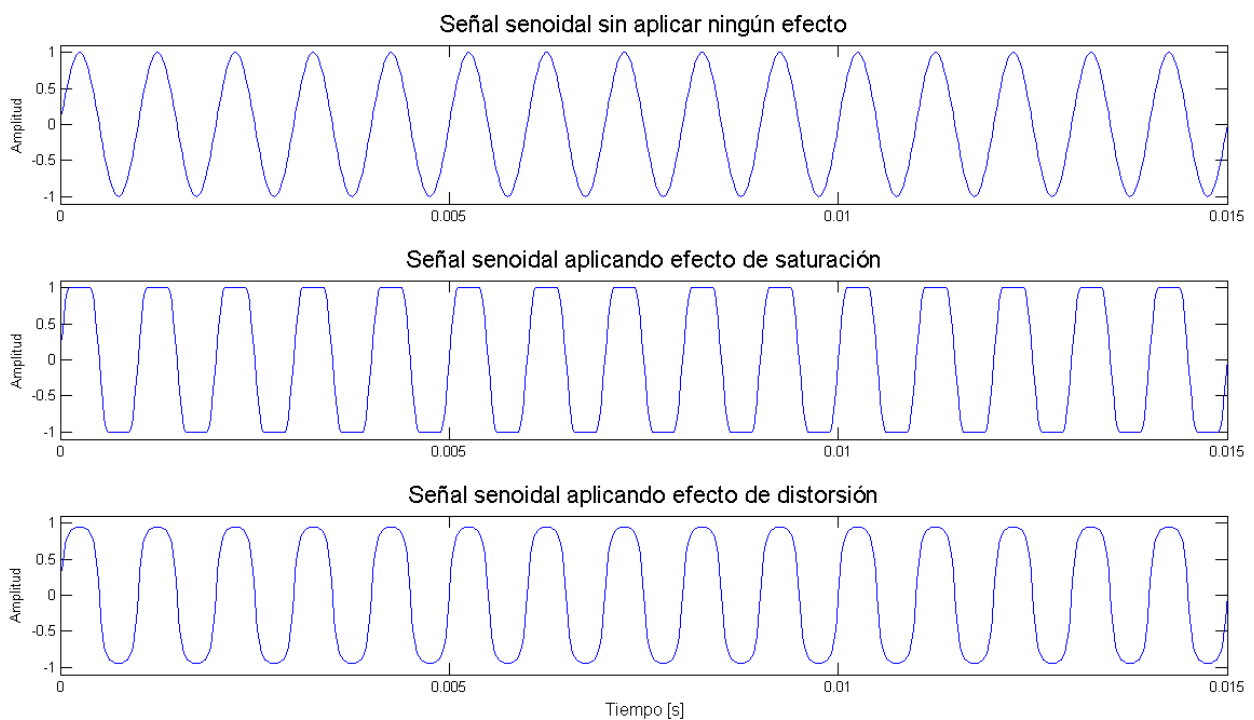


Figura 6.5: Simulación en Octave de los efectos de distorsión aplicados a señal senoidal de  $1\text{[KHz]}$ .

Sin embargo, para obtener el resultado de los efectos aplicados en la señal senoidal en el dominio de la frecuencia, se aplicó a todas las señales mostradas en la figura 6.5 la FFT con un comando propio en Octave, y para poder observarlas mejor, al momento de graficar dichas señales, sólo se graficó el valor absoluto de sus respuestas, por lo que no se muestran las espigas repetidas en cada caso. Es así que nuestro resultado en el dominio de la frecuencia se representa en la figura 6.6. De igual manera se obtuvo por medio de un comando de Octave la distorsión armónica (THD) respectivo para los resultados de la simulación de estos efectos de distorsión, los cuales se presentan en la tabla 6.1.

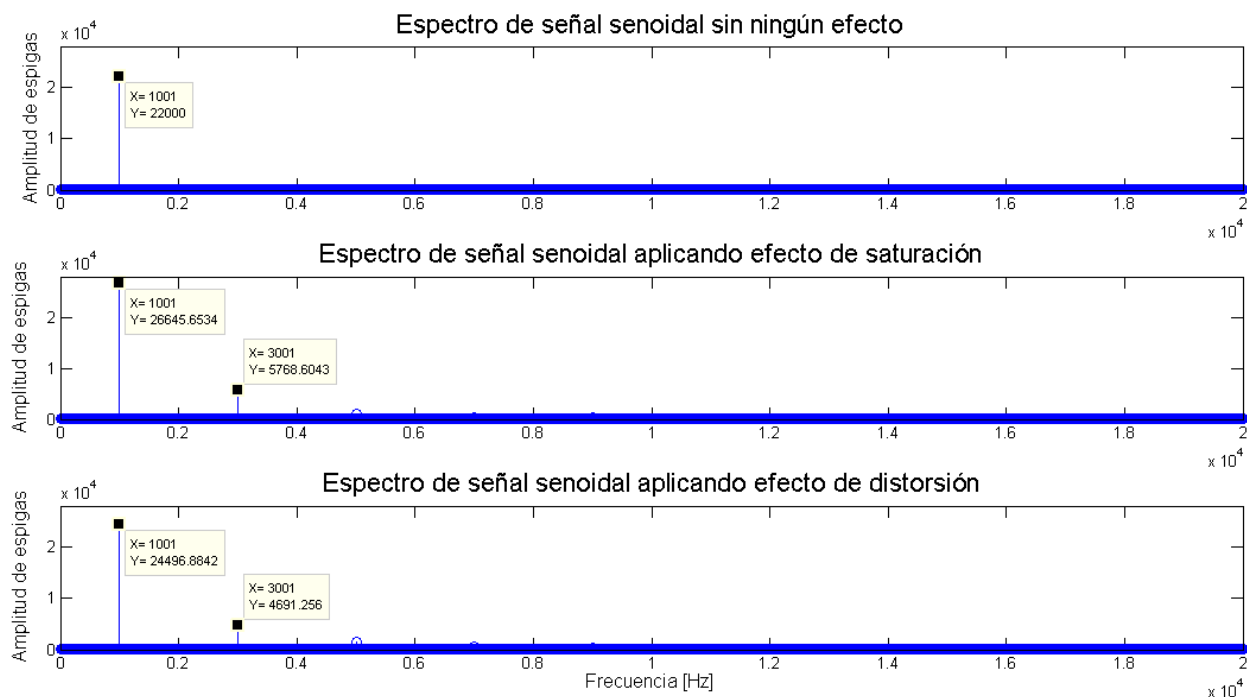


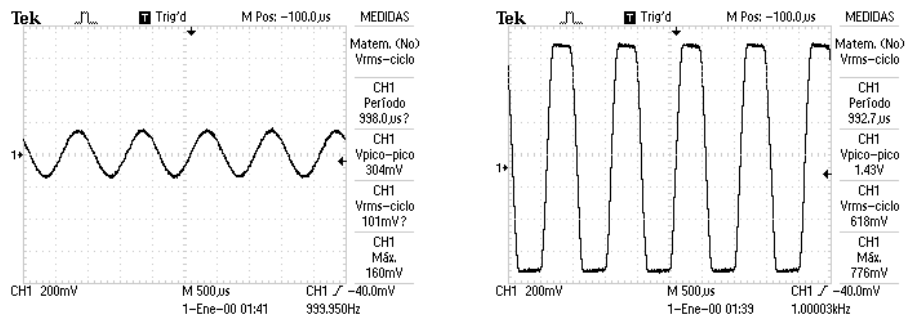
Figura 6.6: Simulación en Octave de los efectos de distorsión aplicados a señal senoidal de 20Hz en el dominio de la Frecuencia.

Efecto aplicado	Distorsión Armónica (THD)
Saturación	35.445 %
Distorsión	27.446 %

Tabla 6.1: Distorsión armónica de las simulaciones de los efectos de distorsión calculadas en Octave.

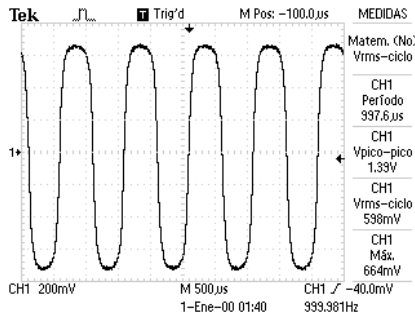
### Resultados experimentales

Para el análisis de estos efectos, se utilizó una señal senoidal de 1[KHz] proveniente de una computadora portátil y se ingresó al sistema conectando un cable de audio estéreo al circuito de acondicionamiento de entrada anterior al ADC del DSP. Posteriormente se observó la salida del sistema ya procesada por el DSP con el osciloscopio. Se utilizó una computadora portátil ya que resulta más sencillo poder observar una distorsión al tener una mayor amplitud en la señal de entrada en comparación con el uso de otros dispositivos electrónicos como teléfonos inteligentes. Los resultados se presentan en la figura 6.7 para el dominio del tiempo y en la figura 6.8 para el dominio de la frecuencia.



(a) Salida del sistema sin aplicar ningún efecto.

(b) Salida del sistema aplicando efecto de saturación



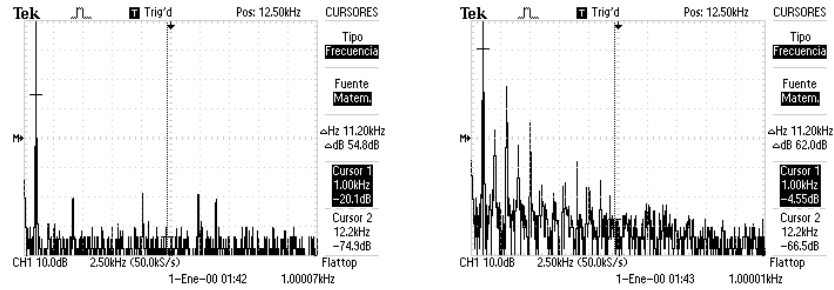
(c) Salida del sistema aplicando efecto de Distorsión

Figura 6.7: Aplicación de los efectos de distorsión.

### Análisis de resultados

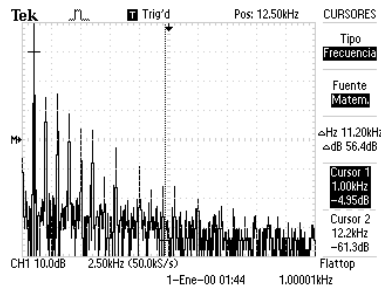
Los efectos de saturación resultan más sencillos de implementar en el DSP debido a que son efectos que no requieren de recolección de muestras anteriores, por lo que en este caso no se tienen problemas en el ajuste de los tiempos de procesamiento.

En el caso del efecto de saturación, se utilizó un valor escalar para amplificar la señal de entrada y que el efecto fuera más perceptible, debido a que la máxima amplitud que se obtuvo de una señal de audio utilizando una computadora portátil fue de aproximadamente 376[mV] y considerando los valores máximos de amplitud que podemos leer en el ADC (1.74 [V]), este valor representa un 21 % de la señal considerada, haciendo que el efecto siempre



(a) Espectro de la señal de salida del sistema sin aplicar ningún efecto.

(b) Espectro de la señal de salida del efecto de saturación.



(c) Espectro de la señal de salida del efecto de distorsión.

Figura 6.8: Espectro de la señal de salida para los efectos de Distorsión.

aplicaba el caso donde la amplitud absoluta de la señal era menor a 33% y no se percibía ningún cambio.

El valor escalar aplicado fue de 5, para incluir un algunos dispositivos cuya salida máxima rondaban entre los 321[mV] y los 376[mV], trabajando así con una señal supuesta de entre 1.605[V] y 1.88[V]. La distorsión armónica de la señal de salida una vez aplicado el valor escalar fue registrada en la tabla 6.2.

Frecuencia [Hz]	Ganancia en DB
$f_0 = 1000$	-04.55
2000	-14.50
3000	-16.90
4000	-32.10
5000	-28.50
6000	-43.70
7000	-54.50
8000	-65.70
9000	-44.9
10000	-45.3

Tabla 6.2: Distorsión armónica del efecto de saturación en una señal senoidal con  $f_0 = 1$ [kHz].

Un problema que se presentó en la implementación del efecto de saturación fue que la simetría de la señal resultante se perdía si el offset no estaba correctamente calculado. El offset base o el agregado por la tarjeta de expansión es de 1.56[V]; si una señal agrega un offset, del orden de 10 [mV], la señal se saturará más rápido en el flanco positivo que el negativo, ocasionando una distorsión no deseada. Con ayuda del osciloscopio se determinó la componente de directa y se ajustó en el código para que la señal se saturara a la misma amplitud en ambos flancos.

Por lo que la distorsión armónica para el efecto de **saturación** es calculada en la ecuación 6.1:

$$THD_{\%} = \frac{\sqrt{\sum_{n=2}^{\infty} V_n^2}}{V_1} = 40.67\% \quad (6.1)$$

donde  $V_n$  es el voltaje de las armónicas de orden  $n$  y  $V_1$  es el voltaje de la fundamental  $f_0$ .

El efecto de distorsión fue modificado de igual manera en el escalar de la operación exponencial por la misma razón descrita en el efecto de saturación. La distorsión armónica esta registrada en la tabla 6.3.

Frecuencia [Hz]	Ganancia en DB
$f_0 = 1000$	-04.95
2000	-18.50
3000	-20.50
4000	-24.10
5000	-36.90
6000	-30.90
7000	-48.90
8000	-38.10
9000	-45.70
10000	-47.30

Tabla 6.3: Distorsión armónica del efecto de distorsión en una señal senoidal con  $f_0 = 1$ [kHz].

Por lo que la distorsión armónica para el efecto de **distorsión** es calculada en la ecuación 6.2:

$$THD_{\%} = \frac{\sqrt{\sum_{n=2}^{\infty} V_n^2}}{V_1} = 29.66\% \quad (6.2)$$

donde  $V_n$  es el voltaje de las armónicas de orden  $n$  y  $V_1$  es el voltaje de la fundamental  $f_0$ .

Como pudimos observar en la implementación de estos efectos, la distorsión debido a la saturación de la señal genera un cambio más abrupto de la señal de entrada en los puntos de máxima amplitud, asemejando la señal senoidal de entrada a una señal cuadrada. Por otro lado, el efecto de distorsión modifica la señal de manera más suave y aún en los puntos de máxima amplitud la señal puede verse ligeramente más redondeada. A sí mismo se puede observar la similitud de la forma de las señales en los resultados de la simulación en la figura 6.5 y los resultados experimentales de la figura 6.7, existiendo una característica de las señales experimentales que difieren con el primer caso: el aumento de amplitud considerable con respecto a la señal de entrada, esto debido a la modificación que se mencionó anteriormente

al tener que multiplicar la señal de entrada por un escalar de manera digital en el código del DSP.

En el caso de sus componentes frecuenciales, comprobamos el aumento de armónicos en ambos efectos, donde para el efecto de saturación se obtuvo una mayor distorsión armónica que en el efecto de distorsión, resultado esperado si recordamos que el recorte de la señal de distorsión es más suave que en el de saturación. También se observa que los valores de THD que se obtuvieron de manera experimental son factibles si los comparamos con los obtenidos en la simulación, sin embargo, en ambos casos el valor experimental es mayor. Esto puede ser, debido a que como se observa en la figura 6.7(a) la señal de entrada ya cuenta con algunas espigas en el eje de la frecuencia debido a que nuestra señal senoidal, a diferencia de la simulada, no es pura y existen ciertas fuentes de ruido tanto en la obtención de la entrada como a la salida del sistema que después del procesamiento añaden distorsión armónica.

### 6.2.2. Análisis de banco de Filtros.

#### Resultados de la simulación

En el caso de este efecto, se realizó un archivo de Octave en el que se sumaron diversas señales senoidales con lo cual se pudieron realizar las pruebas con una sola señal que cuenta con diversos tonos de frecuencias perfectamente identificables. Las frecuencias que se eligieron para la suma total de la señal de prueba fueron las frecuencias centrales de cada uno de nuestros filtros implementados en el efecto (las cuales se muestran en la tabla 6.4), con el objetivo de poder observar mejor y comparar con con más precisión el resultado de los filtros de las simulaciones y los valores experimentales en estas señales. Cabe destacar que esta misma señal de la suma de senoidales se exportó en el mismo código a una archivo de audio, tal que se pudo utilizar esta misma grabación en las pruebas experimentales.

Número de Señal Senoidal	Frecuencia [Hz]
Señal senoidal 1	33
Señal senoidal 2	66
Señal senoidal 3	133
Señal senoidal 4	266
Señal senoidal 5	532
Señal senoidal 6	1065
Señal senoidal 7	2130
Señal senoidal 8	4260
Señal senoidal 9	8520
Señal senoidal 10	17040

Tabla 6.4: Señales senoidales que conforman la suma de la señal resultante utilizada en las pruebas del banco de Filtros.

Sin embargo, para poder evaluar estos resultados tanto simulados como experimentales debemos tomar en cuenta la figura 5.21 de donde nos podemos dar cuenta, y retomando el

diseño de los filtros en el capítulo 5, que debido a la limitación de orden de nuestros filtros, éstos no son filtros ideales y por ende su caída no es vertical, lo que provoca que existan incidencias entre ellos. Para observar mejor estas zonas donde inciden los filtros dentro del mismo archivo de simulación que se utilizó para crear los filtros, se realizó la suma de cada uno. El resultado se presenta a en las figuras 6.9 y 6.10.

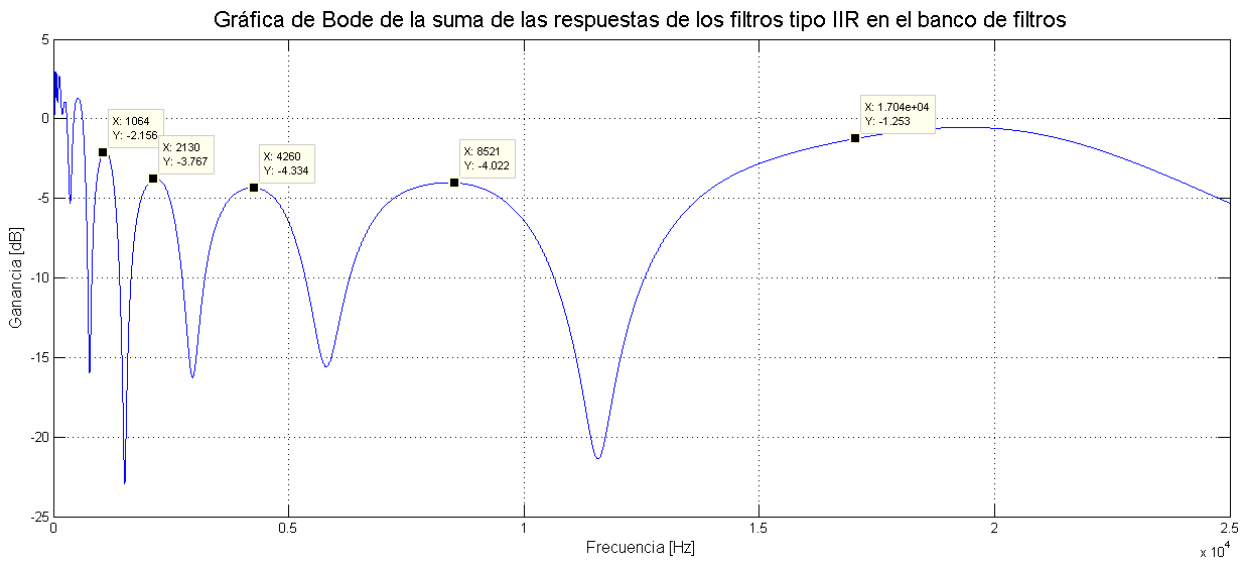


Figura 6.9: Simulación del resultado de la suma de los filtros IIR del efecto del banco filtros con ganancia unitaria de 0 a 25[kHz] en frecuencia.

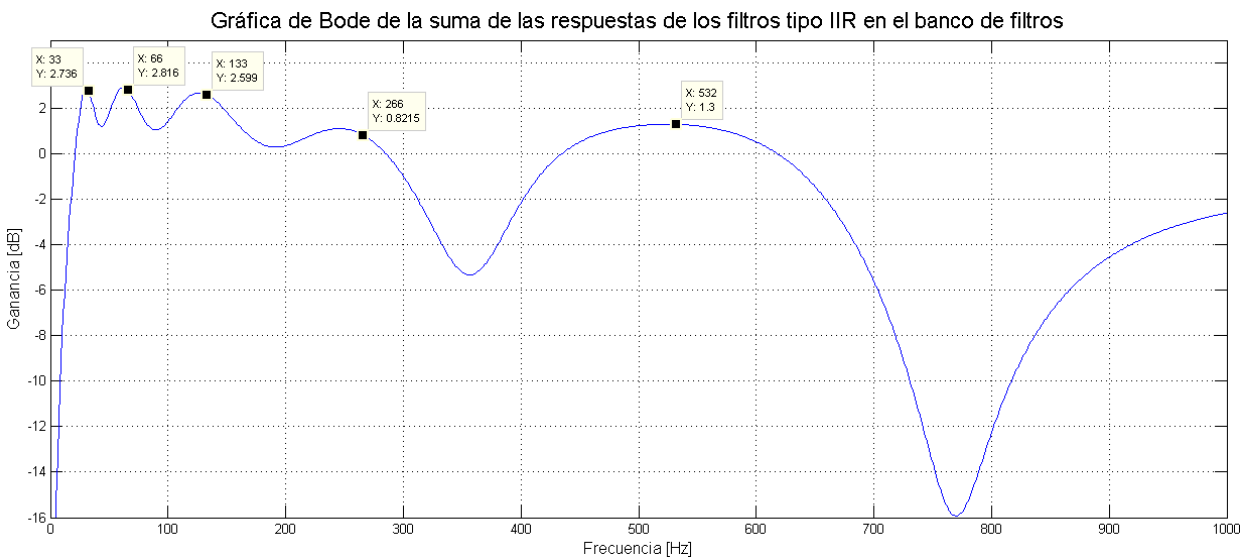


Figura 6.10: Simulación del resultado de la suma de los filtros IIR del efecto del banco filtros con ganancia unitaria ampliado de 0 a 1[KHz] en frecuencia.

Se realizaron dos casos en las simulaciones. En el *primer* caso se removieron las bandas impares y en el *segundo* caso se removieron las bandas pares. Esto se logró multiplicando el resultado de los filtros paso banda pares e impares respectivamente por una ganancia de



0.063 con el fin de reducir la ganancia de los filtros a aproximadamente  $-12dB$ . El resultado del primer caso se puede observar en la figura 6.11 y el segundo caso se observa en la figura 6.12.

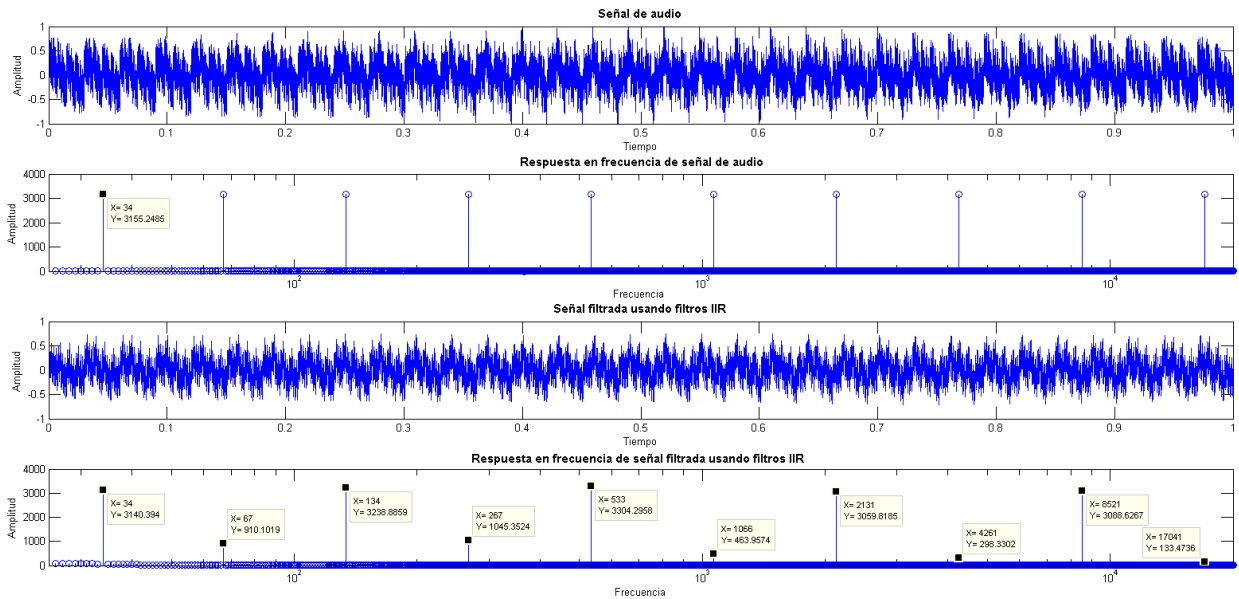


Figura 6.11: Simulación del resultado de la suma de los filtros IIR con atenuación de  $12dB$  en filtros pares.

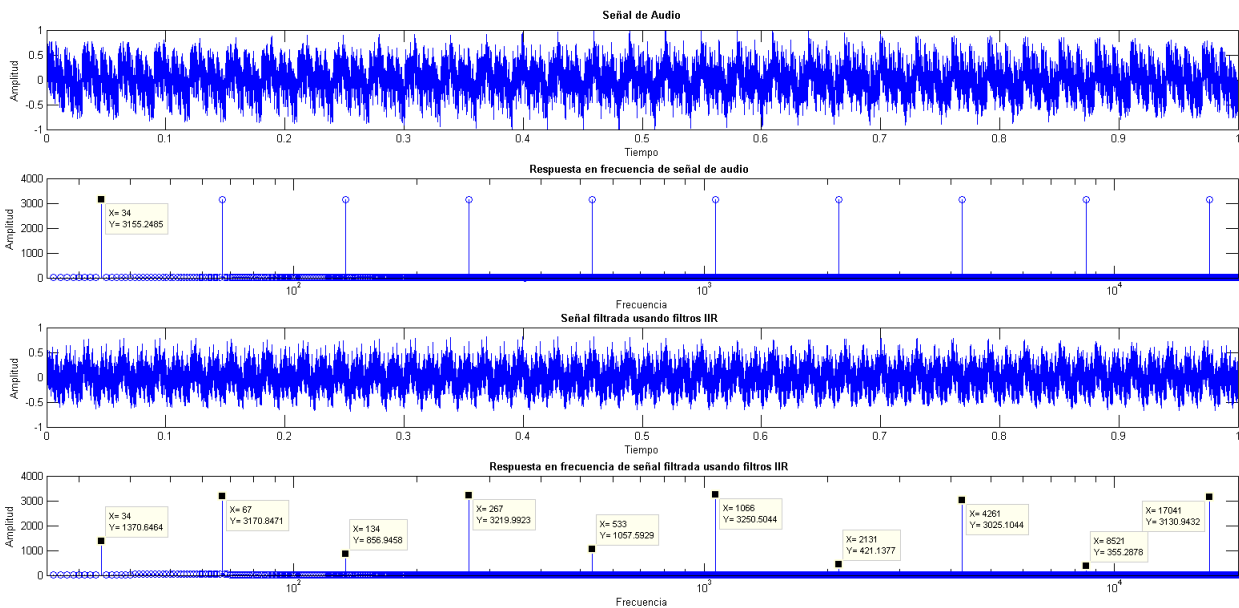
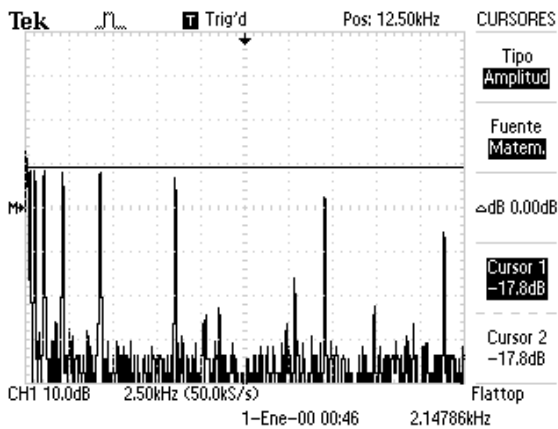


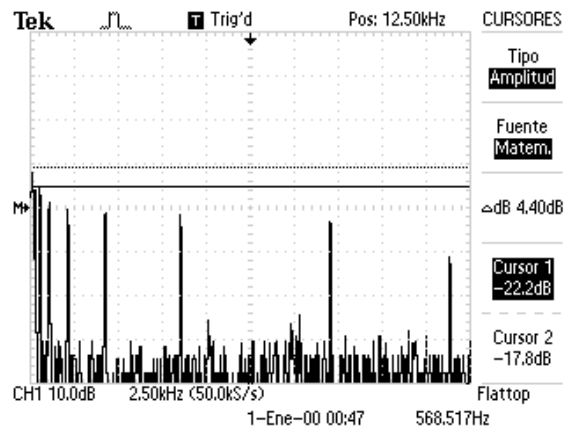
Figura 6.12: Simulación del resultado de la suma de los filtros IIR con atenuación de  $12dB$  en filtros impares.

### Resultados experimentales

Para los resultados experimentales se utilizó una señal senoidal de entrada con las mismas frecuencias que en el caso de las simulaciones y se observó el comportamiento de los filtros analizando la respuesta en el dominio de la frecuencia como se observa en las figuras 6.13, 6.14 y 6.15.

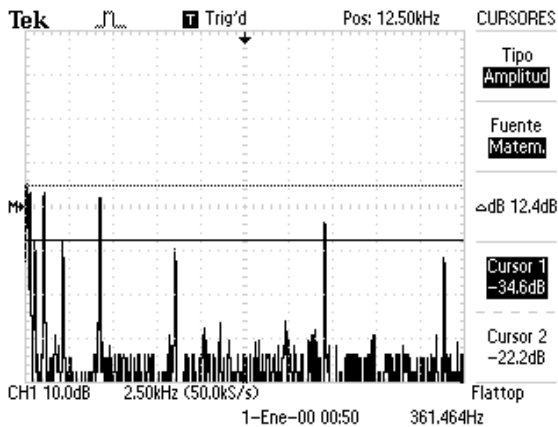


(a) Espectro de la señal de salida sin filtros aplicados.

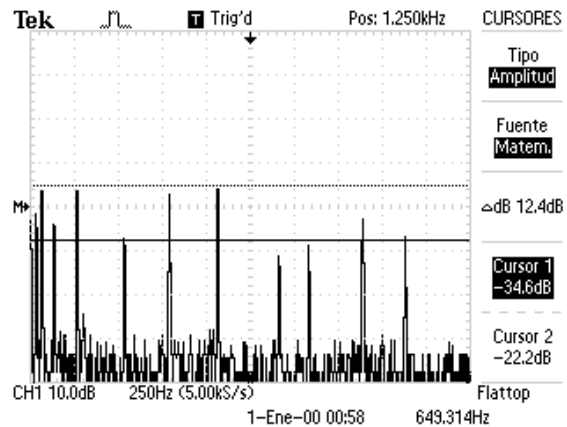


(b) Espectro de la señal de salida con aplicación de filtros con ganancias igual a cero.

Figura 6.13: Espectro de la señal de salida, comparación entre ningún efecto y aplicación de filtros.

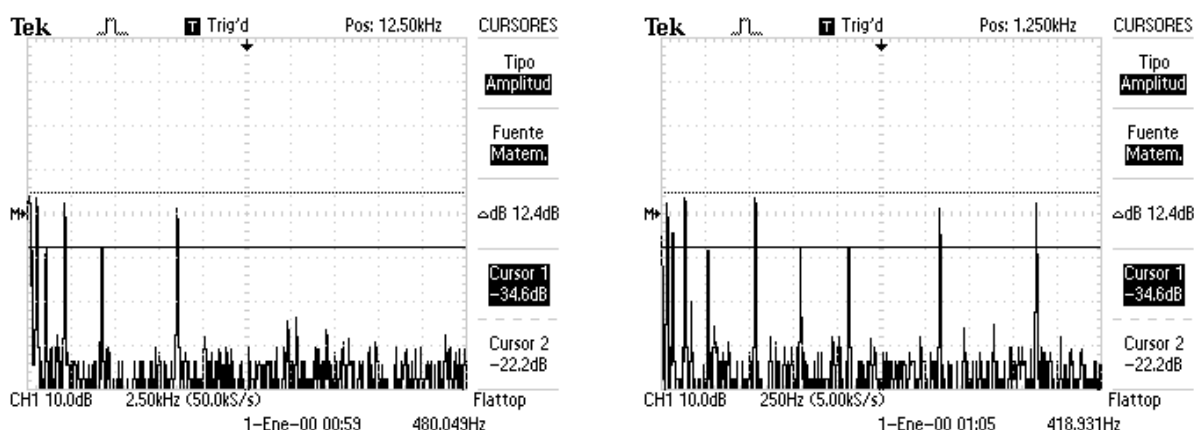


(a) Espectro de la señal de salida con aplicación de los filtros en intervalo de frecuencia de 0 a 50kHz.



(b) Espectro de la señal de salida con aplicación de los filtros ampliada en el intervalo de frecuencia de 0 a 5kHz.

Figura 6.14: Espectro de la señal de salida donde se suprimen las frecuencias de los filtros impares.



(a) Espectro de la señal de salida con aplicación de los filtros en intervalo de frecuencia de 0 a 50kHz.

(b) Espectro de la señal de salida con aplicación de los filtros ampliada en el intervalo de frecuencia de 0 a 5kHz.

Figura 6.15: Espectro de la señal de salida donde se suprimen las frecuencias de los filtros pares.

### Análisis de resultados

En las primeras dos imágenes (figura 6.13) se presentan los espectros de la señal de salida de los resultados experimentales en los casos donde no se selecciona ningún efecto en el sistema (figura 6.13a) y cuando se aplica el efecto de banco de filtros pero no existe ninguna ganancia o atenuación en ninguna banda (figura 6.13b). Aquí se puede notar que se atenúa hasta  $4.40[dB]$  en las espigas de las frecuencias más altas tan solo por la aplicación del efecto, especialmente en las últimas cinco espigas, sin contar la última espiga en los  $17\text{KHz}$  cuyo caso es especial porque esta es reducida aún más debido a un filtro analógico Sallen-Key que se encuentra a la salida de la tarjeta de expansión y cuya frecuencia de corte es de  $15\text{kHz}$ . Si esto se compara con lo que ocurre en la simulación, dónde de la figura 6.11 se espera una atenuación máxima de aproximadamente  $4.33[dB]$  en las frecuencias mayores, podemos decir que es perfectamente factible esta atenuación que obtuvimos de manera experimental.

En las frecuencias menores, al tener filtros IIR de orden uno, la respuesta de los filtros no es tan pronunciada y da como resultado que se traslapen las frecuencias resultantes con otras, generando a su vez, una amplificación en este intervalo de frecuencias. Se puede observar este fenómeno en la espiga de la frecuencia de  $33\text{Hz}$  de la figura 6.13(b) que se encuentra por encima del cursor de referencia. De igual manera esto se puede observar a partir de la figura 6.12 dónde se esperaba un aumento de máximo  $2.88[dB]$  en las primeras cinco frecuencias.

En las figuras 6.14 y 6.15 se puede observar la respuesta en frecuencia de los filtros, en dos casos específicos donde se aplica una atenuación de  $-12[dB]$  a las bandas pares e impares respectivamente (pares e impares de las citadas en la tabla 6.4). De estos resultados podemos observar que tanto para los casos de atenuación de las frecuencias pares como impares se obtuvo una atenuación de aproximadamente  $-12.4[dB]$  de los  $-12[dB]$  esperados en las frecuencias en las que se aplicó el efecto y la ganancia negativa.

El único caso dónde no se cumplió la atenuación ni aproximada de  $-12[dB]$  fue en la espi-

ga en la frecuencia de 17[kHz] debido a que como se comentó anteriormente presenta dos atenuaciones: una por el filtro analógico Sallen-Key de la tarjeta de expansión y otra por los filtros, es por eso que se atenúa más que otras espigas aún sin ningún efecto aplicado y es completamente suprimida al quitar las frecuencias de los filtros pares, como se puede observar en la figura 6.15a.

### 6.3. Análisis del paneo.

#### Análisis experimental

Para el análisis experimental del paneo se ingresó una señal senoidal de 1[kHz] sin aplicar ningún efecto, variando el valor recibido del sensor de orientación para observar su respuesta. Se realizó el análisis de un canal de la salida estéreo (izquierdo), debido a que el coeficiente entre canales varía dependiendo del valor recibido del teléfono celular de forma complementaria. Los datos fueron registrados en la tabla 6.5. Para calcular el error porcentual se utilizó la ecuación 6.3:

$$Error \% = \left| \frac{V_{exp} - V_{teo}}{V_{teo}} \right| * 100 \quad (6.3)$$

donde  $V_{exp}$  es el valor experimental y  $V_{teo}$  es el valor teórico.

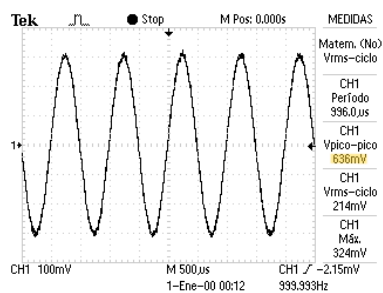
Valor en grados del sensor de orientación	Voltaje pico-pico de la señal [mV]	Atenuación experimental [dB]	Atenuación teórica [dB]	Error [%]
-90[°]	636	0	0	0
-60[°]	612	-0.3341	-0.2176	53.53
-45[°]	588	-0.6815	-0.5460	24.81
-30[°]	552	-1.2304	-1.0827	13.64
-15[°]	508	-1.9587	-1.8834	3.99
0[°]	448	-3.0436	-3.0103	1.10
15[°]	384	-4.3825	-4.5362	3.38
30[°]	308	-6.2981	-6.5630	4.03
45[°]	232	-8.7594	-9.2758	6.22
60[°]	160	-11.9867	-13.1093	5.56
75[°]	92	-16.7933	-19.5103	13.52
90[°]	32	-25.9661	-43.4264	40.20

Tabla 6.5: Valores obtenidos experimentalmente de la aplicación del paneo.

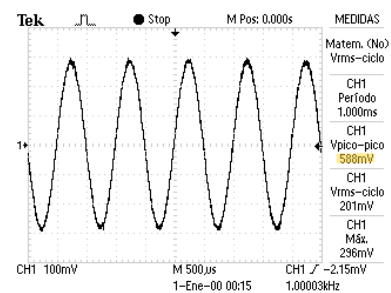
La tabla 6.5 nos permite observar que el efecto fue aplicado de manera correcta, dado que la amplitud de la señal disminuye conforme se invierte el sentido en el sensor de orientación; se esperan valores similares en el canal de audio derecho debido a que la implementación es inversamente proporcional, como se describió en capítulos anteriores. Analizando los resultados obtenidos, podemos observar que en los extremos se presenta el mayor error porcentual; tomando como base el último valor de la tabla, en el que el sensor de orientación registra una lectura de 180°, podemos observar que la señal no es completamente atenuada y que existe

ruido añadido a la señal de entrada, afectando los resultados obtenidos experimentalmente. Podemos obtener este ruido al restar el valor obtenido experimentalmente de voltaje menos el valor esperado de la señal cuando la atenuación es máxima. Al convertir el valor esperado obtenemos 4.2868[mVpp] que son restados de los 32[mVpp] obtenidos, por lo que tenemos un ruido de, aproximadamente, 27.7131 [mVpp] que afecta a todos los resultados obtenidos.

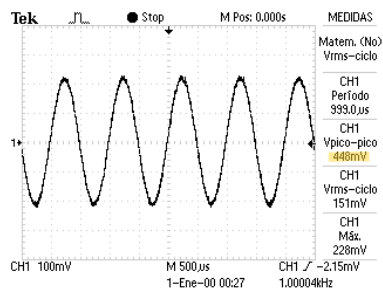
Se optó por calcular los coeficientes utilizando Octave e ingresándolos al sistema utilizando un vector de tipo flotante; la principal ventaja de este método es que el tiempo de procesamiento se ve incrementado dado que no se tiene que utilizar la unidad trigonométrica del DSP para realizar los cálculos. Se utilizó de igual manera el tipo flotante para obtener una respuesta más precisa en comparación con punto fijo a 16 bits. La principal ventaja de esta implementación fue la cantidad de memoria utilizada, ya que se usaron 636 bloques de 32 bits.



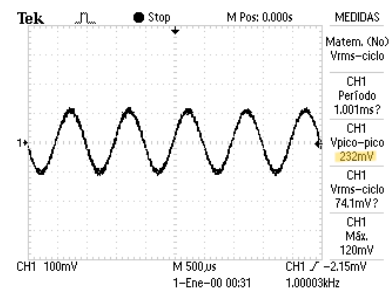
(a) Paneo en canal izquierdo a  $-90^\circ$  de inclinación.



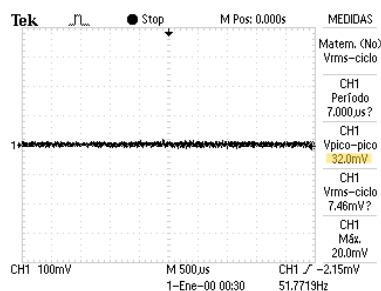
(b) Paneo en canal izquierdo a  $-45^\circ$  de inclinación.



(c) Paneo en canal izquierdo a  $0^\circ$  de inclinación.



(d) Paneo en canal izquierdo a  $45^\circ$  de inclinación.



(e) Paneo en canal izquierdo a  $90^\circ$  de inclinación.

Figura 6.16: Resultados experimentales de la aplicación del paneo.

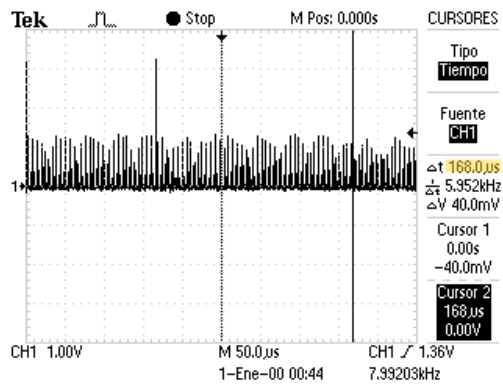
## 6.4. Análisis en tiempo real

De manera práctica, se midió el tiempo que le toma al DSP realizar cada uno de los efectos y los resultados fueron registrados en la tabla 6.6. Utilizando la herramienta del Code Composer Studio "Profile Clock" se midieron la cantidad de ciclos de reloj que le lleva al DSP llegar de un proceso a otro, para el caso de esta Tesis fue desde que la muestra era guardada en el DSP hasta que se aplicaba el Paneo y se registraron en la columna de *Número de ciclos de reloj*. En la columna de *Tiempo medido en el DSP* se tomó como base la frecuencia de 200[MHz] para calcular el tiempo que le toma al DSP la aplicación de cada proceso. Finalmente, se midieron los tiempos de manera experimental con la ayuda de un osciloscopio: se programó el EPWM6 del DSP para generar un pulso de amplitud aproximada de 400[ns] al inicio y al final de la aplicación de los efectos y los resultados se registraron en la columna de *Tiempo medido en osciloscopio*.

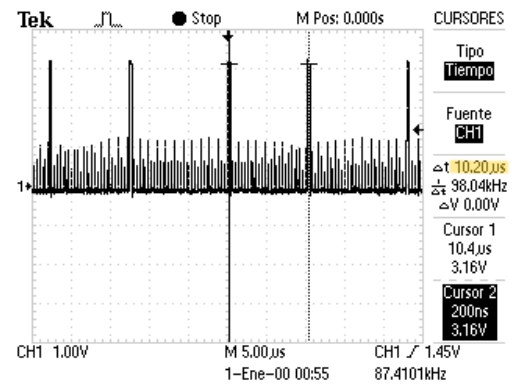
A su vez se presentan dos imágenes de los resultados experimentales de la medición del tiempo que toma en procesar los efectos de reverberación a dos segundos y de filtrado en el DSP, que coinciden con los valores de la tabla 6.6. Los oscilogramas de los resultados experimentales para todos los efectos se pueden encontrar en el anexo C.

<b>Efecto</b>	Número de ciclos de reloj	Tiempo medido en el DSP [ $\mu s$ ]	<b>Tiempo medido en osciloscopio [<math>\mu s</math>]</b>
Ningún efecto aplicado	402	2.01	2.08
Saturación	525	2.625	2.72
Distorsión	1267	6.335	6.40
Reverberación (1 segundo)	23,592	117.96	118.00
Reverberación (2 segundos)	33,298	166.49	168.00 (Figura 6.17a)
Retardo (eco)	4,129	20.645	20.80
Retardo (coro)	7942	39.71	39.60
Banco de filtros	2,070	10.35	10.20 (Figura 6.17b)

Tabla 6.6: Tiempo de la aplicación de los efectos del sistema.



(a) Tiempo de procesamiento del efecto de reverberación a dos segundos.



(b) Tiempo de procesamiento del banco de filtros.

Figura 6.17: Tiempo que el DSP tarda en procesar los efectos de Reverberación a dos segundos y Banco de filtros.

## 6.5. Resumen

En este capítulo se presentaron los resultados tanto de las simulaciones como de las pruebas experimentales para cada efecto. En general se dividió el análisis en los efectos cuya modificación a la señal original de audio se puede observar en el dominio del tiempo (efectos de retardo y reverberación) y en los que también presentan modificaciones en el dominio de la frecuencia (efectos de distorsión y el banco de filtros).

Al inicio del capítulo se desarrollo el análisis en tiempo real que indica cuánto tiempo toma el procesamiento de cada efecto en realizarse en el DSP, con el que pudimos comparar las diferencias entre el tiempo esperado del procesamiento de cada efecto contra el tiempo de procesamiento real, lo cuál posteriormente nos ayudó a realizar modificaciones a los efectos analizados en el dominio del tiempo que presentaron problemas por tomar un tiempo de procesamiento muy largo y por lo tanto también nos permitió analizar de mejor manera las repercusiones de las modificaciones mencionadas en estos efectos.

Tanto en el análisis de resultados en el dominio del tiempo como en el de la frecuencia, se comparó el comportamiento esperado a partir de los resultados de las simulaciones con respecto a los resultados experimentales, y se analizó si el comportamiento de los resultados fueron factibles así como también, se sugirieron algunas causas del por qué había elementos que no coincidían con los datos esperados.



# Capítulo 7

## Conclusiones

La ingeniería de audio ha tomado un papel muy relevante desde hace un par de décadas en la mezcla, edición y modificación del sonido, tanto en proyectos musicales como en archivos multimedia. Sin embargo, gracias al auge de los sistemas de realidad virtual, la implementación de nuevos efectos se ha expandido con el fin de añadir sensaciones más naturales e inmersivas. Esto lo podemos presenciar en los videojuegos, películas o inclusive archivos de audio donde los efectos especiales permiten al escucha recrear las características del audio modificado por el recinto o locación donde se pretende estar sin la necesidad de encontrarse ahí.

Es por esta razón que en este trabajo se investigaron las propiedades de los efectos especiales en audio, sus características, los métodos de implementación y su principal utilidad en la industria musical, cinematográfica o de entretenimiento. Con ello se eligieron y analizaron dos clasificaciones que nos parecieron las más representativas en el campo de la producción musical: *distorsión y ecualización*, y dos clasificaciones representativas de los efectos de inmersión espacial: *reverberación y retardo*. Posteriormente se diseñaron, simularon y programaron dos variantes de cada clasificación; de retardo: *coro y eco*, de reverberación: *reverberación de uno y dos segundos*, de distorsión: *saturación y distorsión* y de filtrado: *banco de 10 filtros*, con el objetivo de tener distintas percepciones de cada tipo de efecto. También se agregó un paneo estéreo el cuál proporciona otro grado de inmersión dependiente del movimiento del usuario, tal como sucede en los sistemas de realidad virtual.

A nivel *hardware*, se logró la implementación del sistema haciendo una división en tres bloques: entrada, procesamiento y salida. Para la entrada se utilizaron solo dispositivos pasivos donde se observó que la principal ventaja de ésta implementación es que se puede reproducir fácilmente a partir de los diagramas eléctricos presentados en esta tesis.

En cuanto a la *etapa de procesamiento*, como se resume en la tabla 6.6 se lograron tiempos de procesos que permiten a las aplicaciones ejecutarse en tiempo real. Adicionalmente se aprovecharon tres módulos en el DSP: el módulo SCI, el convertidor analógico-digital y el convertidor digital-analógico. El escoger una resolución de 16 bits en el ADC permitió que las capturas fueran más precisas, además de disminuir el ruido presente al adquirir la señal. La decisión de utilizar EPWMs resultó en una programación de los periféricos más sencilla y reutilizable, ya que también se usan en diferentes aplicaciones, como en el manejo de

los LEDs contenidos en la tarjeta de expansión o para realizar el monitoreo del tiempo de procesamiento de los efectos.

En el caso del *SCI* se implementó una codificación de las tramas de datos tal que la longitud de la trama fue suficiente para transmitir todos los datos requeridos y que además permite que se puedan agregar más efectos de audio y/o variantes en caso de ser necesario. Finalmente, la *etapa de salida tipo estéreo* compuesta por un filtro y por un offset fue correctamente implementada y probada en instrumentos como audífonos o bocinas. El filtro Sallen-Key proporciona una salida más limpia del audio debido a que elimina las señales de alta frecuencia; sin embargo, su frecuencia de corte presenta inconvenientes al eliminar un intervalo del espectro auditivo teórico, como pudo ser observado en el análisis de resultados de los filtros. El filtrado ideal debería ser aquél que tenga una frecuencia de corte igual tanto en el filtro analógico como en el digital.

A nivel *software* se combinaron los lenguajes ensamblador y C en la programación del sistema lo cual permitió que se cumpliera el objetivo de procesar el audio y los efectos en tiempo real. De manera más específica, programando en lenguaje ensamblador se optimizó la aplicación de los efectos al utilizar la ALU y la FPU así como el procesamiento de ciclos y tareas repetitivas como limpiar y recorrer los buffers de memoria. Por otro lado, el lenguaje C nos permitió la reutilización de código legible en la configuración de los periféricos del sistema.

En los *efectos de retardo y reverberación* se identificó un aumento en el tiempo real de procesamiento con respecto al esperado debido al hecho de no contemplar los procesos paralelos en el DSP durante el diseño de los efectos, que se lograron implementar de manera exitosa después de hacer una corrección, reduciendo el tamaño del buffer de memoria de los retardos, obteniendo en ambos casos una frecuencia de muestreo factible y cumpliendo con los valores de retardo y de ganancia establecidos dentro del rango teórico estipulado para estos efectos.

En los *efectos de distorsión y ecualización* se obtuvieron resultados muy parecidos a los esperados en las simulaciones en cuanto a la atenuación y el cambio de forma deseados en las señales de salida, tanto en el dominio del tiempo como en el de la frecuencia. El mayor inconveniente en estos casos fue la presencia de ruido no deseado causado por ruido externo en los módulos analógicos en las etapas de entrada y salida de audio. Este ruido se aprecia en los oscilogramas obtenidos en el capítulo 6, sin embargo de manera práctica no afecta ni en la distinción ni en la percepción de las modificaciones causadas por los efectos.

La *incorporación del teléfono celular y la aplicación* al sistema, consiguió dar una mayor libertad al usuario y un enfoque más intuitivo gracias a la representación gráfica del cambio de efectos y sus derivados, así como del manejo de las barras deslizantes para el control del ecualizador. Este mismo factor se ve representado en la incorporación de la respuesta al panel directamente proporcional al movimiento que realiza el usuario con el teléfono inteligente.

## 7.1. Trabajo a futuro

A continuación se mencionan algunas propuestas que pueden mejorar el rendimiento del sistema o aumentar su utilidad.

- Realizar un análisis del ruido en el sistema y mejorar la atenuación de ruido en las etapas de acondicionamiento de entrada y salida con métodos de supresión de ruido más avanzados.
- Reutilizar el código de los efectos implementados para ampliar el número de posibilidades en cuanto a la percepción de los efectos (en especial los de retardo y reverberación) y no solo tener dos valores estáticos para cada uno; sino un intervalo donde el usuario pueda elegir más opciones dentro del mínimo y el máximo valor de retardos especificados de manera teórica para cada efecto.
- Aprovechar la codificación de las tramas de datos del SCI que permite transmitir más opciones de efectos de audio con sus respectivas características para elegir de algunos otros efectos presentados en el capítulo 2, en la tabla 5.2 e implementarlos en el sistema.
- Mejorar el rendimiento en los efectos de retardo y reverberación utilizando más módulos del DSP como la CLA y el DMA para transferir datos entre buffers o realizar procesos en paralelo. Esto aumentaría la precisión en los tiempos de procesamiento para los efectos implementados o nuevos efectos que se requieran implementar en los que esta condición de precisión sea indispensable.
- Utilizar el DSP modelo TMS320F28379D que cuenta con los mismos periféricos que el utilizado en esta tesis pero con dos núcleos, por lo que podríamos dividir los procesos estéreo para cada núcleo y mejorar la memoria para los buffers de retardo. Esto permitiría implementar efectos de retardo con una mayor duración como una reverberación de más de dos segundos, o a una frecuencia de muestreo mayor a 8[kHz].
- Diseñar un dispositivo con un acelerómetro - giroscopio independiente anexo a en los auriculares para que los grados de acción que se utilizan en el paneo, sean adquiridos de aquí, y no del giroscopio integrado en el teléfono celular. A su vez, se puede cambiar la configuración estereofónica del paneo para dar más grados de libertad, o incluso realizar un paneo en tres dimensiones. Esto permitiría un nuevo enfoque al usuario aun más relacionado con las nuevas tecnologías de realidad virtual, donde el paneo es directamente proporcional al movimiento de su cabeza.
- Añadir una batería externa de alimentación y buscar la integración de todo el sistema en una PCB para tener una portabilidad en el diseño del dispositivo y una mejor experiencia de usuario.
- Robustecer el sistema al incorporar elementos de detección de errores, como la detección de tramas inexactas en el SCI o sobre voltajes en el ADC. Además de poder enviar información al celular como la temperatura del procesador u otra información relevante del sistema.



# Bibliografía

- [1] Pablo García Valenzuela. «Sistemas de audio multicanal: bases tecnológicas y revisión de la terminología». En: *Perspectiva Interdisciplinaria de Música 2.3-4* (2009), págs. 99-113.
- [2] Joshua D. Reiss y Andrew P. McPherson. *Audio Effects: Theory, Implementation and Application*. EUA: Taylor & Francis Group, 2015.
- [3] Massachusetts institute of technology. *APP inventor*. Código abierto. URL: <https://appinventor.mit.edu..>
- [4] Doug Coulter. *Digital Audio Processing*. 1ra edición. EUA: Cmp Books, 2000.
- [5] Larry H. Escobar. *Conceptos básicos de Procesamiento Digital de Señales*. Facultad de Ingeniería, UNAM, sep. de 2009.
- [6] Thomas L. Floyd. *Dispositivos Electrónicos*. 8va edición. México: Pearson Educación, 2008, págs. 461, 648.
- [7] Udo. Zolzer. *DAFX : digital audio effects*. 2da edición. Reino Unido: John Wiley & Sons Ltd., 2011.
- [8] Carl E. Seashore. *Psychology of Music*. Dover Publications, jun. de 1967.
- [9] Thomas D. Rossing. *The Science of Sound*. 2da edición. Addison-Wesley, 1990, pág. 536.
- [10] Harvey E. White y Donald H. White. *Physics and Music*. Saunders College, 1980, pág. 375.
- [11] Higini Arau-Puchades. «Revisión del tiempo de reverberación.» En: *Arau Acústica* (2010).
- [12] Federico Miyara. «Ecuallizadores». En: *Universidad Nacional de Rosario* (2004).
- [13] *ISO 266*. ISO. 1975.
- [14] *IEC 61260*. IEC.
- [15] Jens Blauert. *Spatial Hearing: The Psychophysics of Human Sound Localization*. Ed. por MIT Press. 1997.
- [16] John Bennett, Keith Barker y Frederick Edeko. «A New Approach to the Assessment of Stereophonic Sound System Performance». En: *Department of Electronic and Electrical Engineering, University of Sheffield JAES 33* (mayo de 1985).
- [17] Ville Pulkki. «Spatial Sound Generation and Perception by Amplitude Panning Techniques.» Tesis doct. Helsinki University of Technology, Laboratory of Acoustics y Audio Signal Processing, 2001.

- [18] Apurba Das. *Signal Conditioning An Introduction to Continuous Wave Communication and Signal Processing*. Alemania: Springer, 2012.
- [19] Ramón Pallás Areny. *Sensores y acondicionadores de señal*. 4ta edición. México: Alfa Omega-Marcombo, 2001.
- [20] Richard C. Jaeger & Travis N. Blalock. *Microelectronic Circuit Design*. 4ta edición. EUA: McGraw Hill, 2011.
- [21] Kees Schouhamer Immink. «The dc-balanced or dc-free codes, as they are often called, have a long history and their application is certainly not confined to recording practice.» En: *EEE Transactions on Communications* (1997).
- [22] Larry H. Escobar. *Diseño de Filtros Digitales*. Facultad de Ingeniería, UNAM, mar. de 2006.
- [23] Alessio Damato. *Electronic linear filters.svg*. Wikimedia Commons, abr. de 2006. URL: [https://commons.wikimedia.org/wiki/File:Electronic\\_linear\\_filters.svg](https://commons.wikimedia.org/wiki/File:Electronic_linear_filters.svg).
- [24] Sutariya Hardik Jayantilal. «Standard for terminology and test methods for analog-to-digital converters.» En: *International Journal for Scientific Research & Development* (2014).
- [25] Kalyan Pathapati Subbu Christopher Barthold y Ram Dantu. *Evaluation of Gyroscope-embedded Mobile Phones*. EUA, 2011.
- [26] Raymond A. Serway y John W. Jr. Jewett. *Physics for scientists and engineers*. 6ta edición. EUA: Thomson BrooksCole, 2004.
- [27] Wei Zhang Fuxue Zhang y Guosheng Wang. *Non-driven Micromechanical Gyroscopes and Their Applications*. Alemania: Springer, 2018.
- [28] Inc. The MathWorks. *Accelerometer*. <https://la.mathworks.com/help/supportpkg/android/ref/accelerometer.html>.
- [29] Fernando de Castro Junqueira y Ettore Apolônio de Barros. «DEVELOPMENT OF A DYNAMICALLY TUNED GYROSCOPE – DTG.» En: *ABCMSymposium Series in Mechatronics - Vol. 1 - pp.470-478*. (2004).
- [30] Roger T. Howe Stephen D. Senturia y Antonio J. Ricco. *MEMS Vibratory Gyroscopes. Structural Approaches to Improve Robustness*. EUA: Springer, 2009.
- [31] F. Dell’Olio M. N. Armenise C. Ciminelli y V. M. N. Passaro. *Advances in Gyroscope Technologies*. Alemania: Springer, 2010.
- [32] Alan Alasdair. *Basic Sensors in IOS*. EUA: O’Reilly, 2011.
- [33] Michael J. Haji-Sheikh Asaf Grosz y Subhas C. Mukhopadhyay. *High Sensitivity Magnetometers*. Nueva Zelanda: Springer, 2017.
- [34] Real Sociedad Española de Física. *Efecto Hall*. <http://rsefalicante.umh.es/TemasElectromagnetismo/Electromagnetismo07.htm>.
- [35] Lenovo. *¿Qué es un smartphone?* <https://www.lenovo.com/mx/es/faqs/pc-vida-faqs/que-es-un-smartphone/>. 2019.
- [36] Lenovo. *How Many Smartphones Are In The World?* <https://www.bankmycell.com/blog/how-many-phones-are-in-the-world>. 2021.

- 
- [37] Andoid - Google. *Sensores*. <https://developer.android.com/guide/topics/sensors?hl=es>. 2021.
- [38] Jesús Maturana - Computer Hoy. *Benchmarking Snapdragon 845: mejora un 30 % el rendimiento y baja un 23 % el consumo*. <https://computerhoy.com/noticias/moviles/benchmarking-snapdragon-845-mejora-30-rendimiento-baja-23-consumo-75685>. 2018.
- [39] Andoid - Google. *Acerca del proyecto de código abierto de Android*. <https://source.android.com/>. 2021.
- [40] Massachusetts Institute of Technology. *Getting Started with MIT App Inventor*. <http://appinventor.mit.edu/explore/get-started>. 2021.
- [41] electronics datasheets. *LAUNCHXL-F28377S*. <https://www.electronicdatasheets.com/manufacturers/texas-instruments/parts/launchxl-f28377s>.
- [42] James M. Fiore. *Operational Amplifiers and Linear Integrated Circuits Theory and Application / 3E*. 3ra edición. EUA: dissidents, 2018.
- [43] Marcel Pelgrom. *Analog-to-Digital Conversion*. 3ra edición. Suiza: Springer, 2017.
- [44] *Standard for terminology and test methods for analog-to digital converters*. IEEE. 2000.
- [45] Bernard M. Gordon. «Linear Electronic Analog/Digital Conversion Architectures, Their Origins, Parameters, Limitations, and Applications». En: *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, VOL. CAS-25, NO. 7* (1978).
- [46] George Clayton y Steve Winder. *Operational amplifiers*. 5ta edición. Reino Unido: Butterworths, 2003.
- [47] Shree Krishna Khadka. *Count Up Type ADC/Stair Case Ramp Type ADC*. [https://www.researchgate.net/figure/fig-count-up-type-adc-stair-case-ramp-type-adc\\_fig18\\_299514235](https://www.researchgate.net/figure/fig-count-up-type-adc-stair-case-ramp-type-adc_fig18_299514235). 2012.
- [48] *TMS320F2837xS Microcontrollers Technical Reference Manual*. Texas Instruments. 2014.
- [49] Larry H. Escobar. *Arquitecturas de DSP TMS320F28xxx y aplicaciones*. Facultad de Ingeniería, UNAM, 2014.
- [50] David R. Smith. *Digital Transmission Systems*. 3ra edición. EUA: Springer Science & Business Media, 2004.
- [51] Larry H. Escobar y otros. *Manual de Algoritmos y Aplicaciones de Procesamiento Digital de Señales Empleando la Familia TMS320F2837xS*. Facultad de Ingeniería, UNAM, oct. de 2019.
- [52] Forge. Octave. *Butterworth*. <https://octave.sourceforge.io/signal/function/butter.html>.





# Glosario de abreviaturas

**AC:** Corriente alterna.  
**ADC:** Convertidor analógico a digital.  
**AHF:** Salto de frecuencia adaptable.  
**AO:** Amplificador operacional.  
**APP:** Aplicación móvil.  
**ARM:** Magnetómetro anisótropo.  
**AT:** Comandos de atención Hayes.  
**CMRR:** Factor de rechazo al modo común.  
**CPU:** Unidad central de procesamiento.  
**DAC:** Convertidor digital a analógico.  
**dB:** Decibeles.  
**DC:** Corriente directa.  
**DSP:** Procesador digital de señales.  
**DTG:** Giroscopio de ajuste dinámico.  
**EPWM:** Modulación por ancho de pulsos mejorado.  
**FD:** Filtros Digitales.  
**FFT:** Transformada rápida de Fourier.  
**FIFO:** Primero en entrar, primero en salir.  
**FIR:** Respuesta finita al impulso.  
**FPGA:** Matriz de puertas lógicas programable en campo.  
**GPIO:** Entrada/Salida de propósito general.  
**GPU:** Unidad de procesamiento de gráficos.  
**HRTF:** Funciones de transferencia relacionadas con la cabeza.  
**Hz:** Hertz.  
**IDE:** Entorno de desarrollo integrado.  
**IEC:** Comisión Electrotécnica Internacional.  
**IIR:** Respuesta infinita al impulso.  
**ILD:** Diferencias temporales interaurales.  
**ISM:** Bandas de radio industriales, científicas y médicas.  
**ITD:** Diferencias interaurales de nivel.  
**ISO:** Organización Internacional de Normalización.  
**ISP:** Unidad de procesamiento de imágenes.  
**LED:** Diodo emisor de luz.  
**MAC:** Dirección de control de acceso.  
**MEMS:** Sistemas Microelectromecánicos.

**MIT:** Instituto de Tecnología de Massachusetts.  
**MR:** Magnetómetro magneto-resistivo.  
**MSB:** Bit más significativo.  
**NC:** Sistema no causal.  
**NPU:** Unidad de procesamiento neuronal.  
**NRZ:** No retorno a cero.  
**PCM:** Modulación por codificación de pulsos.  
**PSRR:** Factor de rechazo a fuente de alimentación.  
**PWM:** Modulación por ancho de pulsos.  
**QR:** Respuesta rápida.  
**RGB:** Estándar *Red, Green, Blue*.  
**S/H:** *Sample & Hold*.  
**SAR:** Convertidor de aproximaciones sucesivas.  
**SCI:** Interfaz de comunicación serial.  
**SIG:** Grupo de especial interés.  
**SLIDT:** Sistema lineal e invariante en el tiempo discreto.  
**SOC:** Inicio de conversiones.  
**SoC:** Sistema en chip.  
**SO:** Sistema operativo.  
**SR:** *Slew Rate*.  
**TB:** Transformada Bilineal.  
**THD:** *Distorsión armónica*.  
**T/H:** *Track & Hold*.  
**UART:** Transmisor-Receptor asíncrono universal.  
**WEB:** Red informática mundial.

# Apéndice A

## Convención de notación eléctrica

La convención de notación eléctrica permite diferenciar entre variables que tienen una componente directa, alterna o ambas mediante una combinación de letras mayúsculas y minúsculas. Cuando se manejan cantidades totales, se utiliza una letra minúscula con subíndice en mayúsculas  $v_{T}o i_{O}$ . Cuando una variable solo cuenta con componente de directa se utiliza tanto letra como subíndice como mayúsculas. Finalmente, las variables solo de alterna manejan letras minúsculas. Un ejemplo de un voltaje de entrada en un sistema puede ser escrito como:

$$\begin{aligned}v_{ID} &= V_{DC} + v_{ac} \\ i_{ID} &= I_{DC} + i_{ac}\end{aligned}\tag{A.1}$$

Las variables con el subíndice  $ID$  cuentan con ambas componentes, de alterna y de directa; aquellas que tienen el subíndice  $DC$  solo tienen componente directa y las que tienen  $ac$  son de alterna.



# Apéndice B

## Propiedades función impulso

Algunas de las propiedades de la función impulso son las siguientes:

- Área bajo la función:

$$\int_{-\infty}^{\infty} \delta(t) dt = 1 \quad (\text{B.1})$$

- Propiedad de muestreo:

$$\int_{-\infty}^{\infty} f(t) \delta(t - t_0) dt = f(t_0) \quad \forall t_0 \in t \quad (\text{B.2})$$

- Idempotencia de funciones: la convolución de una función con la delta de Dirac da como resultado la misma función.

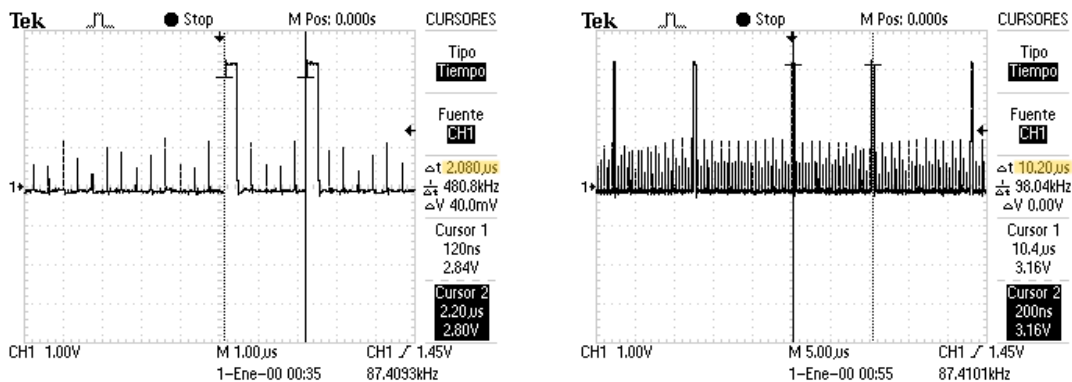
$$\int_{-\infty}^{\infty} x(\tau) \delta(t - \tau) d\tau = x(t) \quad (\text{B.3})$$



# Apéndice C

## Resultados experimentales de los tiempos de procesamiento de los efectos

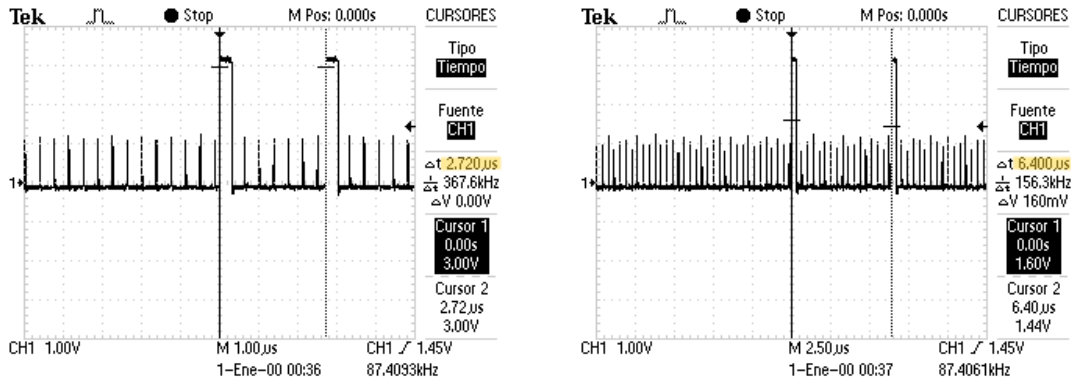
A continuación se muestran los oscilogramas obtenidos del tiempo que le toma al DSP aplicar cada uno de los efectos. Como se mencionó en el Capítulo 6, se programó el EPWM6 del DSP para generar un único pulso al inicio y al final de cada uno de los procesos a una velocidad de 400[ns] donde la respuesta fue observada con un osciloscopio. Los resultados correspondientes a cada uno de los efectos se muestran a continuación.



(a) Tiempo de procesamiento del sistema cuando ningún efecto es aplicado.

(b) Tiempo de procesamiento del sistema al aplicar el banco de filtros.

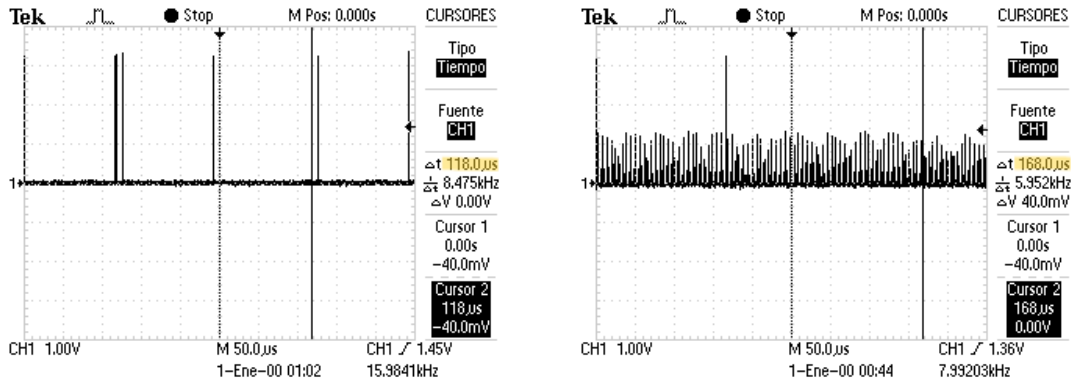
Figura C.1: Tiempo de procesamiento del sistema cuando se aplica el banco de filtros.



(a) Tiempo de procesamiento del sistema cuando se aplica el efecto de saturación.

(b) Tiempo de procesamiento del sistema cuando se aplica el efecto de distorsión.

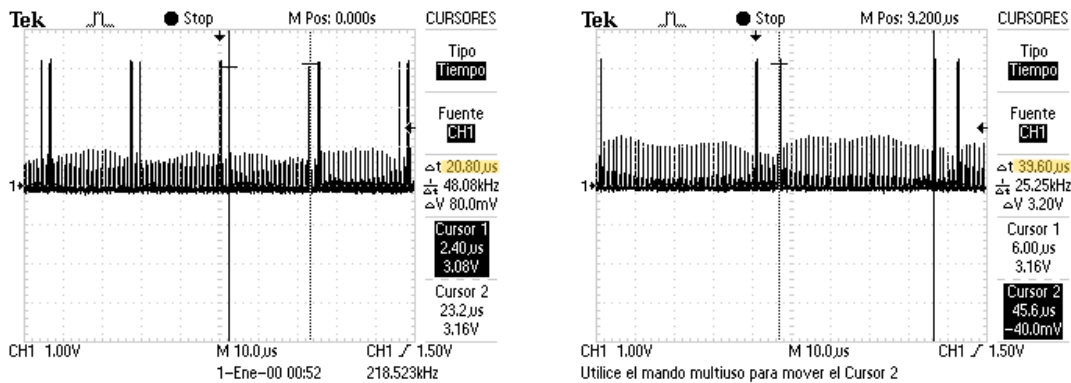
Figura C.2: Tiempo de procesamiento del sistema cuando se aplica el efecto de distorsión.



(a) Tiempo de procesamiento del sistema cuando se aplica el efecto de Reverberación a un segundo.

(b) Tiempo de procesamiento del sistema cuando se aplica el efecto de Reverberación a dos segundos.

Figura C.3: Tiempo de procesamiento del sistema cuando se aplica el efecto de Reverberación.



(a) Tiempo de procesamiento del sistema cuando se aplica el efecto de Coro.

(b) Tiempo de procesamiento del sistema cuando se aplica el efecto de Eco.

Figura C.4: Tiempo de procesamiento del sistema cuando se aplica el efecto de Retardo de la señal.



# Apéndice D

## Código principal

En este anexo, se presenta el código principal implementado en el DSP. Consta de las siguientes partes:

- Declaración de las librerías usadas.
- Definición de macros para valores constantes.
- Prototipos de las funciones usadas de otros archivos.
- Prototipos de las interrupciones.
- Declaración de las variables globales.
- Código principal, empezando con la inicialización de relojes y periféricos.
- Asignación de las interrupciones a sus respectivas rutinas.
- Configuración de los periféricos.
- Rutina principal del programa.
- Subrutinas de interrupción, pasando por los ADC, EPWMs y SCI.

A continuación se anexa la liga y un código QR con los cuales se puede ingresar a un repositorio que contiene todo el código utilizado en esta tesis, tanto del proyecto en el DSP como los archivos de simulación en Octave, para su consulta o libre descarga.



[https://github.com/PacoDelhumeau/Tesis\\_Delhumeau\\_Romero\\_2022](https://github.com/PacoDelhumeau/Tesis_Delhumeau_Romero_2022)

```

teissFinalMain2.c
1 // Archivos que siempre se incluyen
2 #include "F28x_Project.h"
3
4 #include <time.h>
5 #include <math.h>
6 #include <stdlib.h>
7 #include <HeadersC/configPerifericos.h>
8 #include <HeadersC/coeficientesIIR.h>
9 #include <HeadersC/efectosAudioMain.h>
10 #include <HeadersC/coeficientesPaneoEstereo.h>
11
12 /* *****
13 *                               Definición de constantes
14 * *****
15 */
16 #define vref 1
17 #define M_PI 3.14159265358979323846
18 #define FS1 8000
19 #define FS2 44000
20 #define FS3 88000
21 #define Q11 2048
22
23 // Definición de EPWMs
24 #define EPWM1 0
25 #define EPWM2 1
26 #define EPWM6 5
27 #define EPWM7 6
28 #define EPWM8 7
29 #define EPWM9 8
30 // s2 -> dos segundos
31 #define s2 0.5
32 #define s1 1
33
34 // Definición de ADC
35 #define ADCA 0
36 #define ADCB 1
37 #define SOC 0
38 #define EPWM_N 1
39 #define vdirec 30981
40
41 // Retardo de eco en ms
42 #define Eco 20
43 #define Coro 80
44
45 // Definición de DACs
46 #define DACA 0
47 #define DACB 1
48 #define DACC 2
49
50 // Definición de SCI
51 #define SCIB 0
52 #define Tasa_Bauds 115200
53
54
55 /* *****
56 *                               Prototipos de funciones
57 * *****
58 */
59 // Funciones generales
60 int freqOSC(void);
61
62 // Funciones para EL SCI
63 void SCI_Config(Uint16, Uint32);
64
65 // Funciones para el ADC
66 void ADC_Config(Uint16, Uint16, Uint16);
67 void EPWM_Config(Uint16, float, Uint16, float);
68 void configTimer(void);
69 void ADCSOC_Config(Uint16, Uint16, Uint16, Uint16);
70 // Funciones para el DAC
71 void DAC_Config(Uint16);
72
73 // Funciones en lenguaje ensamblador
74 float IIRFloat(int, float, float*, float*);
75 void cleanFloat();
76
77 /* *****
78 *                               Interrupciones
79 * *****
80 */
81 interrupt void ADCA1_TEMP_isr(void);
82 interrupt void ADCB1_AUDIO_isr(void);
83 interrupt void SCIB_inter(void);
84 interrupt void EPWM6_isr(void);
85 interrupt void EPWM7_isr(void);
86 interrupt void EPWM8_isr(void);
87 interrupt void EPWM9_isr(void);
88

```

```

89 /* *****
90 *                               Variables globales
91 * *****
92 */
93 // Variable que recibe el valor transmitido por el SCI
94 char charRecibido = 0;
95 // Decodifica la instrucción recibida (primer valor en el SCI)
96 int instruccion = 0;
97 // Obtiene el valor y modifica las banderas para realizar los efectos (segundo valor en SCI)
98 int valor = 0;
99 // Bandera de los efectos a aplicar. El primero es el efecto [1,4] y la otra es la variante [0,1]
100 int efecto = 1, banderaEfecto = 0;
101 // Matriz que guarda signo y magnitud, respectivamente, de cada filtro
102 float filtros[10] = {1,1,1,1,1,1,1,1,1,1};
103 // Matriz que guarda signo y magnitud del sensor de orientación [0] -> plano X ; [1] -> Plano Y
104 Uint16 sensorOrientacion = 90;
105 // Valores del ADC
106 int32 ADCBcanal2 = 0, ADCBcanal4 = 0;
107 // Multiplicador de prueba
108 float escalarADC = 1;
109 // Bandera de habilitación de EPWM
110 short banderaEpwm = 0;
111 //Bandera para delimitar la llegada de una muestra
112 short banderaMuestreo = 0;
113
114 /* *****
115 *                               Código principal
116 * *****
117 */
118 void main(void){
119 // Variables que almacenan temporalmente el valor de los efectos
120 float canal1 = 0, canal2 = 0;
121 // Numero de muestras
122 int numeroMuestras = 0;
123
124 // Configuración inicial del DSP. Inicialización de relojes y módulos necesarios
125 InitSysCtrl();
126 // Configuración de inicio de los GPIOs
127 InitGpio();
128 InitEPwmGpio();
129
130 /* *****
131 *                               Configuración de las Interrupciones
132 * *****
133 */
134 // Deshabilita las interrupciones para su configuración
135 DINT;
136
137 // Inicia los registros de configuración de PIE
138 InitPieCtrl();
139 // Borra banderas de interrupción
140 IER = 0x0000;
141 IFR = 0x0000;
142 // Inicia el vector de PIE a un valor conocido
143 InitPieVectTable();
144
145 // Direccionamiento de las interrupciones
146 EALLOW;
147 PieVectTable.ADCA1_INT = &ADCA1_TEMP_isr;
148 PieVectTable.ADCB1_INT = &ADCB1_AUDIO_isr;
149 PieVectTable.EPWM6_INT = &EPWM6_isr;
150 PieVectTable.EPWM7_INT = &EPWM7_isr;
151 PieVectTable.EPWM8_INT = &EPWM8_isr;
152 PieVectTable.EPWM9_INT = &EPWM9_isr;
153 PieVectTable.SCIB_RX_INT = &SCIB_inter;
154 EDIS;
155
156
157 /* *****
158 *                               Declaración de funciones necesarias
159 * *****
160 */
161 // Configuración de los ADCs
162 // _asm("\tESTOP0");
163 ADC_Config(ADC_ADCA, ADC_RESOLUTION_12BIT, ADC_SIGNALMODE_SINGLE);
164 ADC_Config(ADC_ADCB, ADC_RESOLUTION_16BIT, ADC_SIGNALMODE_SINGLE);
165
166 // Inicialización del sensor de Temperatura con su offset inicial
167 InitTempSensor(3.0);
168
169 // Configuración de EPWMs
170 // Configura EPWM1 para disparar ADCA
171 EPWM_Config(EPWM1, 2, SOC, 100);
172 //Configura EPWM2 para disparar ADCB
173 EPWM_Config(EPWM2, FS2, SOC, 100);
174 //Configura EPWM6 para DAC
175 EPWM_Config(EPWM6, 3000000, EPWM_N, 50);

```

## Apéndice D. Código principal

```
176 //Configura EPWM7,8,9 para LED
177 EPWM_Config(EPWM7, 2, EPWM_N, 50);
178 EPWM_Config(EPWM8, 2, EPWM_N, 50);
179 EPWM_Config(EPWM9, 2, EPWM_N, 50);
180 //configTimer();
181
182 // Configuración de los canales
183 ADCSOC_Config(ADC_ADCA, 0, 5, 13);
184 ADCSOC_Config(ADC_ADCB, 0, 7, 2);
185 ADCSOC_Config(ADC_ADCB, 1, 7, 4);
186 // Configuración del DAC
187 DAC_Config(DACA);
188 DAC_Config(DACB);
189 // Inicializa los valores del SCIB
190 SCI_Config(SCIB, Tasa_Bauds);
191
192
193 /* *****
194 *          Habilitación de interrupciones
195 * *****
196 */
197 // Habilita interrupciones del grupo 1 (ADCA y ADCB)
198 IER |= M_INT1;
199 IER |= M_INT3;
200 IER |= M_INT9;
201 // Limpia banderas de interrupción
202 PieCtrlRegs.PIEIER1.bit.INTx1 = 1;
203 PieCtrlRegs.PIEIER1.bit.INTx2 = 1;
204 PieCtrlRegs.PIEIER3.bit.INTx6 = 1;
205 PieCtrlRegs.PIEIER3.bit.INTx7 = 1;
206 PieCtrlRegs.PIEIER3.bit.INTx8 = 1;
207 PieCtrlRegs.PIEIER3.bit.INTx9 = 1;
208 PieCtrlRegs.PIEIER9.bit.INTx3 = 1;
209 PieCtrlRegs.PIEIER9.bit.INTx4 = 1;
210 // Enable Global interrupt INTM
211 EINT;
212 // Enable Global realtime interrupt DBGM
213 ERTM;
214
215
216 /* *****
217 *          Configuraciones iniciales
218 * *****
219 */
220 // Habilitación del EPWM1
221 EALLOW;
222 CpuSysRegs.PCLKCR0.bit.TBCLKSYNC = 1;
223 // Habilita el SOCA
224 EPwm1Regs.ETSEL.bit.SOCAEN = 1;
225 EPwm2Regs.ETSEL.bit.SOCAEN = 1;
226 // Activa la cuenta del EPWM
227 EPwm1Regs.TBCTL.bit.CTRMODE = 0;
228 EPwm2Regs.TBCTL.bit.CTRMODE = 0;
229 EDIS;
230
231 //Habilitación del TIMER0
232 CpuTimer0Regs.TCR.bit.TSS = 0;
233
234 // Limpia el vector del filtro
235 cleanFloat();
236
237 while(1){
238     if(banderaMuestreo){
239         EPwm6Regs.TBCTL.bit.CTRMODE = 0;
240         switch(efecto){
241             // Efecto de Distorsion
242             case 2:
243                 if(banderaEfecto == 0){
244                     canal1 = saturacion(ADCBcanal2);
245                     canal2 = saturacion(ADCBcanal4);
246                 }
247                 else{
248                     canal1 = overdrive(ADCBcanal2);
249                     canal2 = overdrive(ADCBcanal4);
250                 }
251                 break;
252             // Efecto de Reverberacion
253             case 3:
254                 numeroMuestras = 5600;
255                 if(banderaEfecto == 1)
256                     numeroMuestras = 8000;
257                 canal1 = reverberacion(numeroMuestras-1, banderaEfecto, ADCBcanal2, 0);
258                 canal2 = reverberacion(numeroMuestras-1, banderaEfecto, ADCBcanal4, 1);
259                 break;
260             // Efecto de retardo de la senal
261             case 4:
262                 numeroMuestras = 880;
263                 if(banderaEfecto == 1)
264                     numeroMuestras = 1760;
265                 canal1 = retardo(numeroMuestras-1, ADCBcanal2, 0);
266                 canal2 = retardo(numeroMuestras-1, ADCBcanal4, 1);
267                 break;
```

```

267 // Banco de filtros
268 case 5:
269     canal1 = ((float)3.3*(ADCBcanal2-vdirec)/65536);
270     canal1 = IIRFloat(0, canal1,&coeficientesIIR[0],&filtros[0])*2048/3;
271     canal2 = ((float)3.3*(ADCBcanal2-vdirec)/65536);
272     canal2 = IIRFloat(1, canal2,&coeficientesIIR[0],&filtros[0])*2048/3;
273     break;
274 // Ningun efecto aplicado
275 default:
276 case 1:
277     canal1 = (ADCBcanal2 - 32768) >> 4;
278     canal2 = (ADCBcanal4 - 32768) >> 4;
279     break;
280 }
281 if(sensorOrientacion < 180){
282     DacaRegs.DACVALS.all = round(canal1 * coeficientesIzq[sensorOrientacion]) + 2048;
283     DacbRegs.DACVALS.all = round(canal2 * coeficientesDer[sensorOrientacion]) + 2048;
284 }
285 banderaMuestreo = 0;
286 EPwm6Regs.TBCTL.bit.CTRMODE = 0;
287 }
288 }
289 }
290
291
292 /* *****
293 * Definición de interrupciones
294 * *****
295 */
296
297 // Interrupción para el sensor de temperatura
298 interrupt void ADCA1_TEMP_isr(void)
299 {
300     // Limpia la bandera de interrupción del ADCA
301     AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1;
302
303     // Condicional en caso de existir overflow
304     if(1 == AdcaRegs.ADCINTOVFL.bit.ADCINT1)
305     {
306         AdcaRegs.ADCINTOVFL.bit.ADCINT1 = 1;
307         AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1;
308     }
309
310     // Acknowledge del vector de PIE
311     PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
312     // Obtiene los datos del sensor de temperatura
313     int sensorSample = AdcaResultRegs.ADCRESULT0;
314     // Convierte el valor del sensor de temperatura a Celcius
315     int sensorTemp = GetTemperatureC(sensorSample);
316 }
317
318 // Interrupción del ADCB para muestrear la señal de audio
319 interrupt void ADCB1_AUDIO_isr(void)
320 {
321     // Limpia la bandera de interrupción del ADCB
322     AdcbRegs.ADCINTFLGCLR.bit.ADCINT1 = 1;
323
324     // Condicional en caso de existir overflow
325     if(1 == AdcbRegs.ADCINTOVFL.bit.ADCINT1)
326     {
327         AdcbRegs.ADCINTOVFL.bit.ADCINT1 = 1;
328         AdcbRegs.ADCINTFLGCLR.bit.ADCINT1 = 1;
329     }
330     // Acknowledge del vector de PIE
331     PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
332
333     // Verifica que la frecuencia sea la correcta
334     if(efecto == 3) modifFreqEPWM(EPWM2, FS1, 100);
335     else modifFreqEPWM(EPWM2, FS2, 100);
336
337     ADCBcanal2 = AdcbResultRegs.ADCRESULT0;
338     ADCBcanal4 = AdcbResultRegs.ADCRESULT1;
339     banderaMuestreo = 1;
340 }
341
342 // Muestreo de los efectos
343 interrupt void EPWM6_isr(void){
344     EPwm6Regs.ETCLR.bit.INT = 1;
345     PieCtrlRegs.PIEACK.all = PIEACK_GROUP3;
346     EPwm6Regs.TBCTL.bit.CTRMODE = 3;
347     EPwm6Regs.TBCTR = 0x0000;
348 }
349

```

```

350 // Apaga EPWM7 -> LED Azul
351 interrupt void EPWM7_isr(void){
352     EPwm7Regs.ETCLR.bit.INT = 1;
353     PieCtrlRegs.PIEACK.all = PIEACK_GROUP3;
354     if(banderaEpwm >= 2)
355         banderaEpwm = 0;
356     else{
357         EPwm7Regs.TBCTL.bit.CTRMODE = 3;
358         banderaEpwm++;
359     }
360     EPwm7Regs.TBCTR = 0x0000;
361 }
362
363 // Apaga EPWM8 -> LED Verde
364 interrupt void EPWM8_isr(void){
365     EPwm8Regs.ETCLR.bit.INT = 1;
366     PieCtrlRegs.PIEACK.all = PIEACK_GROUP3;
367     if(banderaEpwm >= 5)
368         banderaEpwm = 0;
369     else{
370         EPwm8Regs.TBCTL.bit.CTRMODE = 3;
371         banderaEpwm++;
372     }
373     EPwm8Regs.TBCTR = 0x0000;
374 }
375
376 // Apaga EPWM9 -> LED Rojo
377 interrupt void EPWM9_isr(void){
378     EPwm9Regs.ETCLR.bit.INT = 1;
379     PieCtrlRegs.PIEACK.all = PIEACK_GROUP3;
380     if(banderaEpwm >= 10)
381         banderaEpwm = 0;
382     else{
383         EPwm9Regs.TBCTL.bit.CTRMODE = 3;
384         banderaEpwm++;
385     }
386     EPwm9Regs.TBCTR = 0x0000;
387 }
388
389 // Interrupcion del SCIB para recibir información del Smartphone
390 interrupt void SCIB_inter(void){
391     // Variables locales para datos
392     int indiceFiltro = 0, signo = 0;
393     float magnitud = 0;
394
395     // Limpia banderas para poder recibir otra interrupción
396     ScibRegs.SCIFFRX.bit.RXFFOVRCLR = 1 ;
397     ScibRegs.SCIFFRX.bit.RXFFINTCLR = 1 ;
398     // Reconoce la interrupción
399     PieCtrlRegs.PIEACK.all = PIEACK_GROUP9;
400
401     // Guarda el valor recibido en charRecibido
402     charRecibido = ScibRegs.SCIRXBUF.all;
403
404     // Identifica si es el primer valor recibido
405     if(instruccion == 0)
406         instruccion = (int)charRecibido;
407     else{
408         // Decodifica la instrucción y el valor a ser aplicados
409         valor = (int)charRecibido;
410         switch(instruccion & 0xF0){
411             case 0x10:
412                 switch(valor){
413                     case 1:
414                         modifFreqEPWM(EPWM2, FS1, 100);
415                         break;
416                     case 2:
417                         modifFreqEPWM(EPWM2, FS2, 100);
418                         break;
419                     case 3:
420                         modifFreqEPWM(EPWM2, FS3, 100);
421                         break;
422                 }
423             break;

```

```

424     case 0x20:
425         efecto = valor & 0x0F;
426         // El bit más significativo tiene la variante del efecto
427         banderaEfecto = (valor & 0x80) >> 7;
428         banderaEpwm = 1;
429         if(efecto == 3)
430             modifFreqEPWM(EPWM2, FS1, 100);
431         else
432             modifFreqEPWM(EPWM2, FS2, 100);
433         EPwm7Regs.TBCTL.bit.CTRMODE = 0;
434         //Limpia el buffer para los efectos
435         cleanFloat();
436         cleanSignal(12000);
437         break;
438     case 0x40:
439         // El numero de filtro está en la parte menos significativa
440         indiceFiltro = (instruccion & 0x0F)-1;
441         // Obtiene el signo
442         signo = (valor & 0x80) >> 7;
443         // Obtiene la magnitud del filtro en DBs en formato Q3
444         if(signo == 1){
445             signo = -1;
446             magnitud = 256-valor;
447         }else{
448             signo = 1;
449             magnitud = valor;
450         }
451         // Esta en formato Q3*base log
452         magnitud = signo*magnitud/(8*10);
453         magnitud = pow(10,magnitud);
454         filtros[indiceFiltro] = magnitud;
455         EPwm8Regs.TBCTL.bit.CTRMODE = 0;
456         break;
457     case 0x80:
458         //Instrucción = 128 + x -> Giroscopio
459         // Obtiene el signo del Sensor de Orientación
460         signo = (valor & 0x80) >> 7;
461         // Los grados de giro del sensor de orientación
462         if(signo == 1){
463             signo = -1;
464             magnitud = 256-valor;
465         }else{
466             signo = 1;
467             magnitud = valor;
468         }
469         sensorOrientacion = signo*magnitud + 90;
470         EPwm9Regs.TBCTL.bit.CTRMODE = 0;
471         break;
472     }
473     //Reinicia los valores para recibir
474     valor = 0;
475     instruccion = 0;
476 }
477 }
478
479 // Fin del archivo
480

```