



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA ELECTRÓNICA

Licenciatura en Ingeniería en Mecatrónica

Diseño de un modulador Delta-Sigma para aplicaciones biomédicas

Presenta:

Diana Jessica Gómez Barrientos

TESIS

Tesis presentada para obtener el grado de
Licenciada en Ingeniería en mecatrónica

Asesor

Dr. Víctor Rodolfo González Díaz

Puebla, Puebla a Junio 2025

Agradecimientos:

A mi familia, por el apoyo incondicional y la paciencia. A mi institución y compañeros, donde me nutrí de conocimiento, forje lazos de amistad y descubrí nuevos caminos. Por último, un agradecimiento especial al Consejo Nacional de Humanidades, Ciencia y Tecnología (CONAHCyT) por haberme otorgado una beca mediante el programa de Ciencia de Frontera, proyecto 319601.

Contenido

1	Introducción.	1
1.1	Procesamiento digital en un mundo analógico.	1
1.2	Justificación.	2
1.3	Objetivos.	6
1.3.1	Objetivos específicos.	6
1.4	Biopotenciales y Aplicaciones biomédicas.	6
1.5	Especificaciones.	8
1.6	Señales analógicas y digitales.	8
1.7	Convertidores de datos.	9
2	Características de un ADC y procesamiento de señales.	11
2.1	Muestreo y cuantificación.	11
2.2	Error de cuantificación.	13
2.3	Resolución, relación señal a ruido y número equivalente de bits.	15
2.4	Tipos de convertidores.	17
2.5	OSR y ENOB.	18
2.6	ADC tipo Flash	19
2.7	Análisis de respuesta en frecuencia.	21
2.8	Conceptos clave en la transformada discreta de Fourier.	22
3	Modulador Delta-Sigma.	25
3.1	Filtro integrador.	25
3.2	Modulador Delta-Sigma.	26
3.2.1	STF Y NTF.	30
3.2.2	Moldeado de ruido.	31
3.3	Moduladores de tiempo continuo y tiempo discreto	32
4	Diseño del modulador delta-sigma	37
4.1	Ejemplos de arquitecturas existentes.	37
4.2	Propuesta de diseño ideal.	40
4.2.1	Consideraciones de estabilidad	41
4.3	Implementación del circuito en MATLAB-simulink CT.	43
4.3.1	Diagrama de bloques en simulink.	44
4.3.2	Diagrama a bloques del modulador de tres bits.	47
5	Resultados.	51
5.1	Implementación de macro modelo en Cadence-Virtuoso.	51

5.2	Análisis transitorio.	61
5.3	Análisis en frecuencia.	62
5.4	Desempeño del modulador aplicando una señal ECG	65
6	Conclusiones y trabajos a futuro	71
6.1	Conclusiones.	71
6.2	Trabajo a futuro.	72
	Apéndices	73
A	Script de MATLAB modulador de 1 bit.	73
B	Script de MATLAB modulador de 3 bits.	75
C	Código comparador fully diferencial	77
D	Código decodificador	78
E	Codigo de matlab, relacionando los datos de cadence	80
	Referencias	83

CAPÍTULO 1

Introducción.

1.1. Procesamiento digital en un mundo analógico.

El mundo físico en el que se desarrollan los seres vivos -humanos, animales y plantas- está constituido por una variedad de estímulos, como sonidos, colores, imágenes, temperatura, etc. Estos estímulos conforman el entorno del ser humano. Y a medida que la evolución y el desarrollo avanzan, se han implementado metodologías y herramientas que han ido cambiando la manera en que el ser humano interactúa con este entorno. De esta manera, lo que antes era percibido únicamente por los sentidos ahora también puede ser apreciado desde dispositivos electrónicos: desde la imagen de un paisaje o la voz de un ser querido hasta sistemas más complejos como el control de temperatura en una habitación. Al existir este gran enfoque en el desarrollo digital, suele desenfocarse como es que estos datos fueron recolectados en primer lugar, ya que el formato nato de un paisaje no viene dado en píxeles, de manera similar, una cadena de dígitos binarios no es naturalmente comprensible para el ser humano.

Con esto en mente, se describe la cadena de procesos a desarrollar para adquisición y procesamiento de datos (Figura 1.1). Primero, las señales físicas antes ejemplificadas son captadas por sensores, los cuales se encargan de convertirlas en señales eléctricas, como voltaje, corriente o carga [1], pero suelen ser señales débiles o con una cantidad de ruido considerable, por lo cual pasan por un proceso de filtrado y amplificación conocido como acondicionamiento de la señal; de esta manera se reduce el ruido para la siguiente etapa: la etapa del convertidor de datos [2]. El convertidor de datos convierte las señales analógicas captadas por el sensor, en señales digitales, las cuales son más fáciles de almacenar y manipular, requiriendo menos consumo de potencia. Finalmente, esta señal pasa por un proceso digital en el cual es modificada según las especificaciones deseadas para el tipo de aplicación.

Es por esto que, para que el desarrollo y vinculación de estas tecnologías digitales tenga lugar, es indispensable hablar primero sobre convertidores de datos.



Figura 1.1: Sistema de adquisición de datos.

1.2. Justificación.

El portal ENLACE de la Organización Panamericana de la Salud (OPS) reporta a la cardiopatía isquémica y el accidente cerebrovascular como las principales causas de mortalidad del año 2000 al 2019 en el continente Americano, el número absoluto de muertes por enfermedades cardiovasculares aumentó de 1.76 millones en el 2000 a 2.02 millones en 2019 [3] [4] (Figura 1.2.). México no se queda atrás, en 2022 las cinco principales causas de muerte a nivel nacional fueron: enfermedades del corazón, diabetes mellitus, tumores malignos, enfermedades del hígado y COVID-19, como se muestra en la gráfica recuperada del comunicado de prensa INEGI 2022 (Figura 1.3) [6].

Recientemente, un informe conjunto de la Organización Mundial de la Salud (OMS) y la Unión Internacional de Telecomunicaciones (UIT), hace un llamado a la acción para la inversión en



Figura 1.2: causas principales de muerte y pérdida en salud en las Américas, 2000-2019 recuperada de [5].

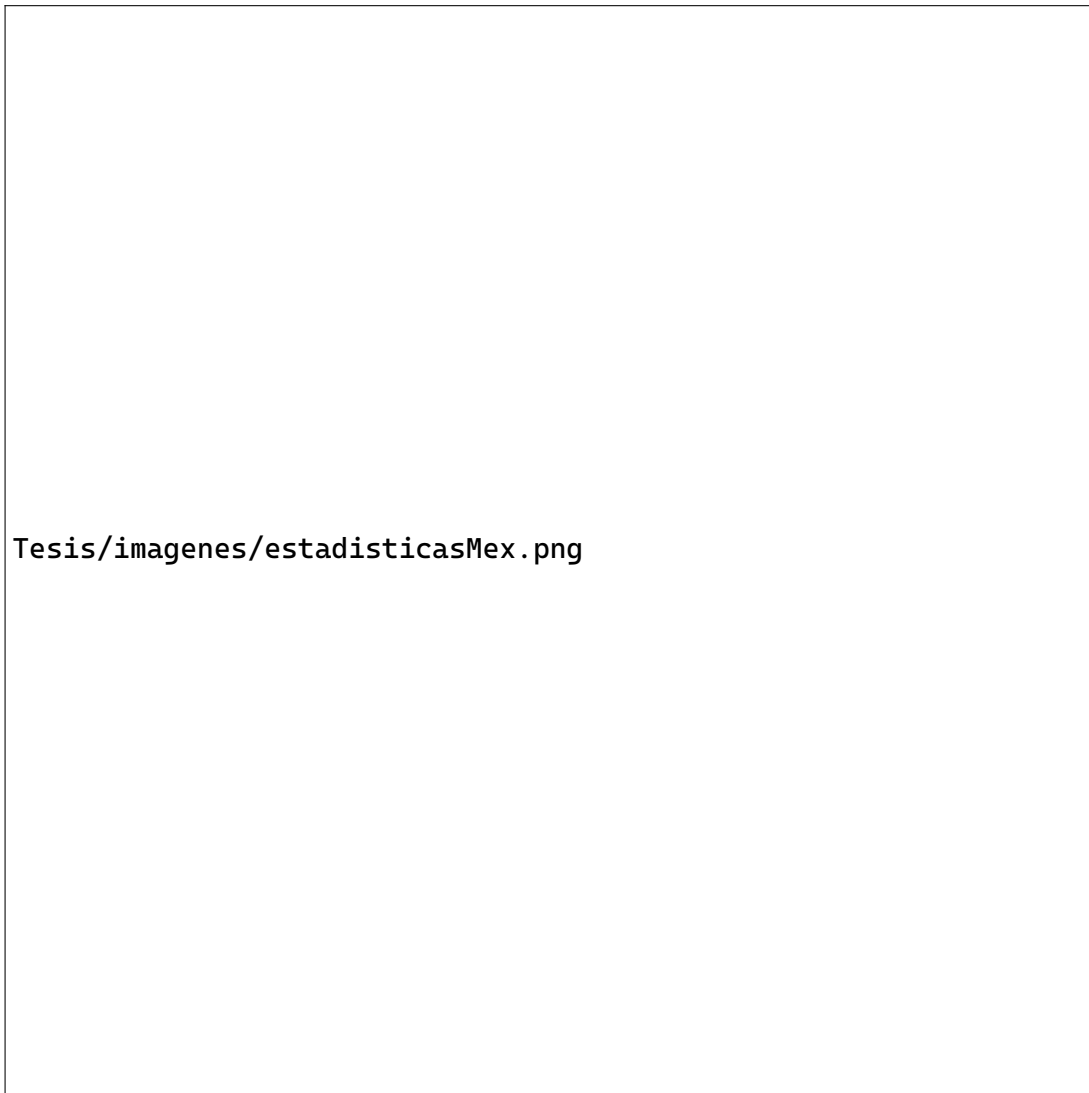


Figura 1.3: Estadísticas de Defunciones Registradas 2022 INEGI.

tecnología y digitalización en el área salud [7], argumentando que puede impulsar una gran transformación en el ámbito de la salud. “Instamos a intensificar la colaboración entre los sectores de la salud y la tecnología... Ya que las cardiovasculopatías, el cáncer, la diabetes y las enfermedades respiratorias crónicas, causan cada año más del 74% de las defunciones en el mundo, pero muchas de estas pueden ser previstas con un monitoreo y detección adecuados.” A pesar de los avances significativos logrados en la lucha contra estas enfermedades, aún queda mucho por hacer para integrar las tecnologías de salud digital en los sistemas de salud convencionales [4].

En los últimos años, la organización Panamericana de la Salud ha propuesto enfatizar la investigación y regulación de los dispositivos de medición clínica, ya que, la comisión de la OPS ha evaluado que de los aproximadamente 3000 dispositivos de medición que existen en el mercado, menos del 15% ha publicado evidencia sobre el desempeño y precisión de los mismos [8]. El consejo directivo recomendó a los gobiernos priorizar el desarrollo de marcos legales y organizativos que supervisen la calidad, eficacia y seguridad de las tecnologías sanitarias.

Ya que los convertidores de datos son elementos esenciales para la adquisición y procesamiento de señales, se propone el desarrollo y optimización de dispositivos médicos, a través de la implementación de un convertidor de datos que permita altas resoluciones, bajos consumos de potencia y fiabilidad con bajos costos de producción.

Un convertidor de datos A/D aumenta su resolución, incrementando el número de bits en el cuantificador. El problema en la mayoría de convertidores tradicionales es que esto implicaría un aumento en el consumo de potencia, mayores costos de producción, área en silicio y tiempo de respuesta.

Este trabajo de Tesis muestra la aplicación de una arquitectura más eficiente para la conversión analógica a digital con aplicaciones biomédicas, cuya naturaleza permita cumplir con las características requeridas pero sin afectar otros parámetros. La arquitectura que se propone y se trabajará a lo largo de este documento es la de un modulador Delta-Sigma [9].

1.3. Objetivos.

Diseñar y simular un modulador Delta-Sigma con especificaciones para aplicaciones biomédicas.

1.3.1. Objetivos específicos.

- Proponer la arquitectura para un modulador Delta-Sigma con un cuantificador multi-bit, describiendo las especificaciones para aplicaciones biomédicas.
- Crear un modelo comportamental en Matlab (Simulink) para evaluar los coeficientes y la resolución límite con base en la arquitectura.
- Describir la arquitectura a nivel descripción comportamental en la herramienta Cadence-Virtuoso.
- Realizar macro modelo en circuito eléctrico en Cadence Virtuoso.
- Analizar los resultados del diseño mediante un análisis espectral (estimación de la resolución efectiva).

1.4. Biopotenciales y Aplicaciones biomédicas.

En 1786 Galvani descubre a través de experimentación con ranas, que al estimular los nervios con electricidad, se ocasiona la contracción del músculo [10]. Existen ciertas células las cuales pertenecen a diferentes tejidos (glandular, nervioso y muscular) que tienen la capacidad de generar potenciales eléctricos debido al intercambio iónico entre sus membranas celulares. Estas células mantienen un potencial de reposo, que cambia a un potencial de acción al momento de ser estimuladas. [11]. Este potencial de acción, ya sea generado por una única célula o el promedio de grupos de células, se propaga a través de los tejidos, permitiendo su detección superficial [12].

Estas señales eléctricas, conocidas como biopotenciales, son monitoreadas, registradas y clasificadas como señales biomédicas [13]. Algunos ejemplos de estas señales son: ECG (que registra la actividad eléctrica del corazón), EEG (que mide la actividad eléctrica del cerebro), EMG (que evalúa la actividad eléctrica muscular) y el EGG (que mide la actividad eléctrica de los músculos estomacales). Es así como se emplean diferentes técnicas de registro y análisis para estudiar la actividad eléctrica en diversas partes del cuerpo, para un diagnóstico y seguimiento clínico adecuados.

Tipo de señal	Rango de frecuencia [Hz]	Amplitud
EEG	0.5-60	15 μ V-100 μ V
EMG	10-200	100 μ V-5mV
ECG	0.05-250	100 μ V (niños) 5 mV(adultos)
IEGM	0.7-70	2mV-10.5 mV
ENG	250-5.000	100 μ V

Tabla 1.1: Rangos de frecuencia y amplitud en los que normalmente operan las señales biomédicas. Tabla recuperada de [14].

Como en el estudio y manipulación de cualquier otra señal, es necesario conocer los rangos de frecuencia y amplitud en los que normalmente operan los biopotenciales, estos pueden variar según el método o dispositivo de adquisición utilizado [12]. Las señales biomédicas operan a velocidades bajas [14], la mayoría de sus componentes se encuentra en un rango de frecuencia desde DC hasta 5 kHz; en el eje de la amplitud también se encuentran valores relativamente pequeños, estos oscilan entre 100 μ V hasta 10 mV, como se observa en la Tabla 1.1. Es por esto que las especificaciones de ancho de banda en un convertidor ADC enfocado en aplicaciones biomédicas se encuentran entre DC y hasta 5 kHz y su rango dinámico corresponde entre 8 y 12 bits [15]. Como se muestra en las distintas interfaces y convertidores para aplicaciones biomédicas, desarrollados en: [16] [17] y [18] los cuales se desglosan en la siguiente tabla.

Aplicación específica	Adquisición de señales para un marca-pasos [16]	ADC para aplicaciones biomédicas[17]	Sistemas de adquisición datos biomédicos [18]
Resolución	8 bits	8 bits	8 bits
Tipo de convertidor	Modulador DS de tercer orden	Time-based ADC	Modulador DS se primer orden
Fs	8 kHz	45 kHz	512 kHz
OSR	16	convertidor tasa Nyquist	100
SNR	47 dB	47 dB	100 dB
BW	250 Hz	20 kHz	500 Hz

Tabla 1.2: Resolución, tipo de convertidor, Fs, OSR, SNR, BW encontrados en trabajos de conversión de datos y aplicaciones biomédicas previos.

Existe una tendencia creciente en el diseño y manufactura de dispositivos inalámbricos y portátiles para aplicaciones biomédicas, lo cual ha generado una fuerte demanda para una nueva generación de convertidores analógico-digitales energéticamente eficientes. Los dispositivos biomédicos deben de ser pequeños y portables, por lo que es indispensable que cuenten con baterías reducidas (pero duraderas) y su consumo en potencia debe ser el mínimo posible [19]. Por lo tanto, se proponen arquitecturas de conversión energéticamente eficientes, que logren velocidad y precisión moderadas; publicaciones recientes muestran que las arquitecturas algorítmicas, de aproximación sucesiva y de sobre-muestreo satisfacen estas características [17].

1.5. Especificaciones.

El modulador delta-sigma, que se desea diseñar, está orientado al procesamiento de señales biomédicas o biopotenciales. Pero, es importante mencionar que la etapa de adquisición y acondicionamiento de la señal, no es descrita en este trabajo. Se asume, el hecho de que cualquier señal biomédica que se desee aplicar al sistema ha pasado previamente por un proceso de amplificación y filtrado. Ya que, como se ha mencionado, el convertidor de datos, conforma una de múltiples etapas que se llevan a cabo en la recolección y procesamiento de señales biomédicas. Dicho esto, se presentan las especificaciones técnicas propuestas para el diseño del modulador, definidas en función de las características típicas de las señales biomédicas:

- Amplitud de la señal de entrada: 500 mV
- Resolución mínima: 10 bits
- Relación señal a ruido (SNR): superior a 60 dB
- Ancho de banda: 5 kHz

Estas especificaciones fueron seleccionadas considerando los rangos de amplitud y frecuencia en los que comúnmente operan este tipo de señales, los cuales se observan en la Tabla 1.1. Y establecen los estándares de calidad y eficacia que se esperan del convertidor, por lo que es importante definirlos. Y sus conceptos serán explicados más adelante.

1.6. Señales analógicas y digitales.

Los elementos mencionados anteriormente: imagen, sonido, temperatura, intensidad lumínica, presión, etc. Existen de manera continua y cambiante (en la mayoría de los casos) en cada momento del tiempo. Estas señales físicas son traducidas a través de sensores a: voltaje, corriente, carga o flujo magnético. Pero siguen siendo señales continuas, esto quiere decir que tienen un valor asignado para cada momento del tiempo y su amplitud varía en valor y forma. Estas son las principales características de las señales analógicas. Por otro lado, las señales digitales son discretizadas en tiempo y amplitud, esto quiere decir que tienen valores exactos

únicamente en momentos de tiempo determinados y a diferencia de las señales analógicas, no cuentan con un valor “todo el tiempo” sino en espacios exactos del tiempo y su amplitud suele oscilar únicamente entre 0 y 1. Las señales analógicas suelen ser convertidas a señales digitales, esto debido a la facilidad de procesamiento, transmisión y almacenamiento de datos con la que se cuenta en el dominio digital, ya que las señales analógicas requieren circuitos más específicos, además de mayor consumo de potencia y espacio de memoria, añadiendo que las señales analógicas producen mucho más ruido, por lo tanto, mayor distorsión de la señal. Una vez que la señal digitalizada es procesada, almacenada y transmitida, se transforma nuevamente al dominio analógico, debido a la interfaz humano-máquina, que se encarga de que estos datos, vuelvan a ser comprensibles para el ser humano, un ejemplo de esto es la transmisión y reproducción de audio/música.

1.7. Convertidores de datos.

Los convertidores de datos, son la interfaz que conecta el dominio analógico (señales físicas en su mayoría) con el dominio digital (utilizado en procesamiento de datos). Un convertidor de datos se desempeña como un transformador de señales: de una señal de tiempo y amplitud continuos a una de tiempo discreto y amplitud cuantificada y viceversa [20]. Existen diferentes topologías en el tema de convertidores de datos, pero la principal característica que los identifica, es qué tipo de dato van a transformar y en qué lo van a transformar, existiendo así:

- Convertidores analógico-Digital (Analog to Digital Converter ADC)

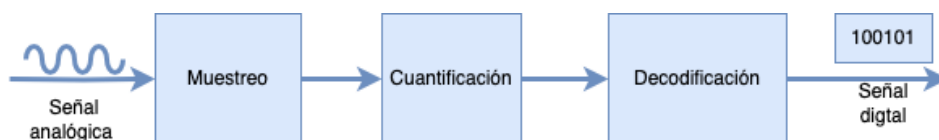


Figura 1.4: proceso interno de un ADC.

- Convertidores Digital-Analógico (Digital to Analog Converter DAC)



Figura 1.5: Proceso interno de un DAC.

El DAC, como su nombre lo indica, convierte señales en el dominio digital al dominio analógico, su funcionamiento parte de dos etapas principales: transconductancia y reconstrucción (Figura 1.5). Este convertidor traduce códigos binarios digitales, a cantidades de señales analógicas, las cuales están equitativamente distribuidas a lo largo de un rango dinámico [21]. Por otra parte, el ADC se encarga de convertir señales analógicas a digitales y normalmente se simplifica en 3 pasos: muestreo, cuantificación y codificación de datos[22] (Figura 1.4). Los convertidores de datos son esenciales para el procesamiento de datos, ya que las señales digitales, contienen menos distorsión, son más fáciles de almacenar, transmitir y su procesamiento consume muy poca potencia en comparación con las señales analógicas. De esta manera, se recolectan las señales analógicas, se trasladan al dominio digital, son procesadas y nuevamente transformadas al dominio analógico.

CAPÍTULO 2

Características de un ADC y procesamiento de señales.

2.1. Muestreo y cuantificación.

Muestreo.

El muestreo suele ser el primer paso en la transformación de una señal A/D. Este, parte de una señal continua en cada momento del tiempo, y la convierte en una señal discreta; las señales discretas contienen el valor en amplitud de una señal continua en el momento exacto del muestreo, es decir que cada valor corresponde a un **único valor** seleccionado en un **tiempo determinado** (Las señales muestreadas son secuencias o vectores de datos cuyos valores son asignados únicamente en sus momentos de muestreo [28]). A diferencia de las señales continuas estas no contienen un valor en todo momento del tiempo, solo en múltiplos de t_s .



Figura 2.1: Imagen recuperada de [28].

La cantidad de muestras que se toman en un segundo indica la frecuencia de muestreo f_s (muestras/segundo) y el inverso de esta frecuencia es una constante de tiempo periódica que indica la cantidad de tiempo que existe entre cada muestra $t_s = \frac{1}{f_s}$. La representación matemática de una señal sinusoidal discretizada es [28]:

$$x(n) = \sin(2\pi f_o n t_s) \quad (2.1)$$

Donde n identifica qué número de muestra adquirida de la señal muestreada; t_s es el tiempo que tarda en tomarse cada muestra y $n * t_s$ indica el tiempo exacto en el que es tomada la muestra n .

Cuantificación.

Un convertidor analógico cuenta con una etapa de cuantificación la cual consta de dividir el rango dinámico del convertidor (es decir $V_{max} - V_{min}$ presentes en la entrada) o también llamado voltaje de escala completa (*full scale* V_{FS}) entre 2^n intervalos de cuantificación, existiendo $2^n - 1$ voltajes con nivel de cuantificación, donde n representa la resolución en bits del cuantificador. El rango de amplitud que corresponde a cada intervalo, es llamado Δ o V_{LSB} dependiendo la literatura, pero ambos hacen referencia al incremento mínimo de voltaje que se puede

distinguir entre dos niveles de cuantificación consecutivos [22] [29]. El cálculo de este, está dado por la siguiente ecuación:

$$\Delta = V_{LSB} = \frac{V_{REF}}{2^n - 1}$$

Una vez que estos intervalos son establecidos, la amplitud de la señal de entrada se le asigna un voltaje de referencia que corresponde su nivel de amplitud ciertos instantes de tiempo. De esta manera, la amplitud antes variante en todo momento del tiempo, ahora es asignada a valores fijos, los cuales son determinados según el número de bits con los que vaya a contar el convertidor de datos.



Figura 2.2: Ejemplo de niveles de cuantificación y cuantificación de la señal, imagen obtenida de [30].

2.2. Error de cuantificación.

Se define como la diferencia de voltaje presente, entre una señal continua y la señal discretizada de la misma. El error de cuantificación existe de manera inherentemente al momento de que una señal es cuantificada, es decir, es parte natural del proceso de cuantificación, por lo tanto, **siempre** debe ser considerado, independientemente de la resolución presente en el convertidor.

Esto es debido a que la señal cuantificada está sujeta a valores discretos y siempre existirá una diferencia entre el valor de la señal cuantificada y el valor de la señal analógica V_{in} , siendo esta diferencia, el error de cuantificación. El cual es expresado generalmente con la siguiente

ecuación:

$$Y = X_{in} + \epsilon_Q$$

La cual se puede describir como la salida Y del cuantificador, es igual a la señal de entrada X_{in} más un error de cuantificación llamado ϵ_Q (siendo este último una señal que puede variar entre valores positivos y negativos).

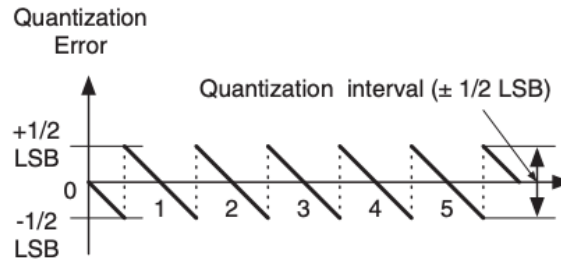


Figura 2.3: Comportamiento del error de cuantificación, imagen recuperada de [20].

Si la señal de entrada se cuantifica en pasos uniformes, y su distribución permite que todos los niveles de cuantificación sean atravesados con la misma probabilidad, el error de cuantificación tiende a distribuirse de forma uniforme en el intervalo $[-\Delta/2, \Delta/2]$ como se observa en la Figura. 2.3, y este es una variable aleatoria, entonces, puede modelarse como ruido blanco.

Potencia del ruido de cuantificación.

La potencia promedio en el tiempo del ruido de cuantificación, se asume como una función de probabilidad constante en el rango de $[-\Delta/2, \Delta/2]$, por lo tanto, la potencia promedio del ruido de cuantificación está dada por:

$$P_Q = \int_{-\infty}^{\infty} \epsilon^2 \cdot p(\epsilon_Q) d\epsilon_Q = \int_{-\Delta/2}^{\Delta/2} \frac{\epsilon_Q^2}{\Delta} d\epsilon_Q = \frac{\Delta^2}{12}$$

Esta se encuentra directamente ligada al paso de cuantificación, por lo que, entre mayor sea el número de bits en el cuantificador, la potencia del ruido disminuirá, intuitivamente podemos deducir que esto es debido a que, entre mayor sea el número de bits, la señal será más sensible a los cambios, por lo que el error de cuantificación disminuirá y por ende su potencia.

Densidad espectral de potencia del ruido.

El espectro de potencia del ruido, representa cómo se distribuye la potencia del error sobre todo el intervalo de frecuencia $[-f_s/2, f_s/2]$ y su unidad es W/Hz o dB/Hz. Y se denota por la siguiente ecuación: [20]).

$$S_e(f) = \frac{\Delta^2}{12} \cdot \frac{1}{f_s}$$

La PSD es plana en $[-f_s/2, f_s/2]$ porque el ruido de cuantificación se modela como ruido blanco en banda base, como se ha explicado anteriormente; véase la Figura 2.4.

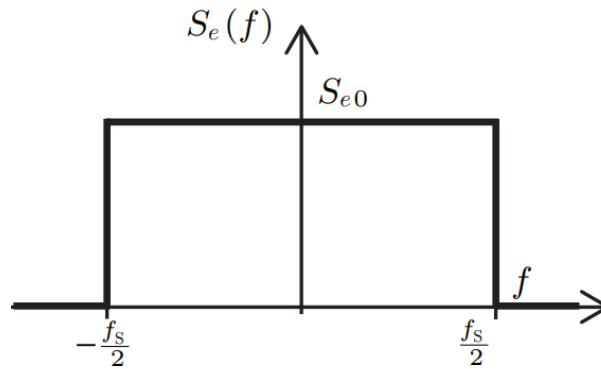


Figura 2.4: Distribución de la potencia sobre unidad de Hz en el rango de frecuencia $[-f_s/2, f_s/2]$.

En palabras prácticas, la densidad de potencia del ruido de cuantificación, se distribuye uniformemente sobre el rango de frecuencia, donde su nivel de potencia es inversamente proporcional al rango de frecuencia, ya que, si se aumentará la frecuencia de muestreo, no disminuiría la potencia en el tiempo del ruido, pero si disminuirá la densidad espectral de potencia de este, ya que cuenta con más rango de frecuencia para distribuirse, un punto importante en el desarrollo del modulador Delta-Sigma. Al aumentar f_s , la densidad espectral de potencia $S_e(f)$ disminuye porque, la misma potencia total se distribuye en un rango de frecuencias más amplio. Sin embargo, la potencia promedio del ruido P_Q permanece constante, ya que depende únicamente del paso de cuantificación Δ .

2.3. Resolución, relación señal a ruido y número equivalente de bits.

Resolución.

La resolución es un parámetro que indica los intervalos de cuantificación con los que se ha procesado la señal. Esta se relaciona directamente con la cantidad de intervalos de cuantificación del convertidor: a mayor número de intervalos, menor será la pérdida de información. Esto se debe a que el convertidor se vuelve más sensible a los pequeños cambios en la señal de entrada. Cuantos más intervalos tenga el cuantificador (lo que implica que cada intervalo será más pequeño), mayor será la resolución. Esto significa que el convertidor podrá detectar variaciones más sutiles en la señal.

En los convertidores analógico-digitales (ADC), la resolución suele medirse en bits, ya que a mayor número de bits (N), mayor será la precisión del sistema. En este caso, el número de niveles de cuantificación es 2^N , y por lo tanto, el rango dinámico se divide en más partes, aumentando la capacidad del convertidor para representar con exactitud la señal original.

Por tanto, la resolución de amplitud puede definirse como la capacidad del convertidor para identificar e interpretar pequeños cambios de voltaje [19].

$$N(\text{número de bits}) = \text{resolución del convertidor de datos}$$

Relación señal a ruido.

El error de cuantificación es una de las principales herramientas para medir la conversión respecto de la señal original. Este error es interpretado como ruido blanco debido a su distribución uniforme por todo el ancho de banda de la señal.

Este error (o pérdida de información), sumado a diferentes tipos de ruidos (como ruido térmico o ruidos provocados por los componentes físicos), distorsionan la señal y disminuyen la capacidad de preservar las características de la señal original[20]. Por lo tanto, es necesario, medir la calidad de la señal cuantificada con respecto a la señal de entrada. La manera de evaluar cuán cercanas son las señales es con la relación señal a ruido (Signal-to-Noise Ratio SNR). La cual compara la potencia de la señal con la potencia proveniente del ruido, esta relación obedece a la siguiente ecuación:

$$\text{SNR} = \frac{P_{\text{señal}}}{P_{\text{ruido}}}$$



Figura 2.5: Señal comparada con el ruido, referente a SNR.

Una alta SNR significa mejor calidad y claridad en la señal. Mientras que una baja SNR significa mayor dificultad para distinguir la potencia de la señal de la potencia del ruido, esto

se traduce en distorsiones o pérdida de información, se ejemplifica en la Figura 2.5 donde existe una alta SNR, por lo tanto, la señal se diferencia fácilmente del ruido.

Normalmente, es más útil visualizar esta relación en decibelios por lo que la ecuación queda de la siguiente manera:

$$SNR|_{dB} = 10 \cdot \log \frac{P_{señal}}{P_{ruido}}$$

La ecuación para el calculo de potencia del ruido esta dada por [20] :

$$P_Q = \int_{-\infty}^{\infty} \epsilon_Q^2 \cdot p(\epsilon_Q) d\epsilon_Q = \int_{-\Delta/2}^{\Delta/2} \frac{\epsilon_Q^2}{\Delta} d\epsilon_Q = \frac{\Delta^2}{12}$$

Y el calculo de potencia de una señal sinusoidal esta dado por [20]:

$$P_{sin} = \frac{1}{T} \int_0^T \frac{X_{FS}^2}{4} \sin^2(2\pi ft) dt = \frac{X_{FS}^2}{8} = \frac{(\Delta \cdot 2^n)^2}{8}$$

Por lo tanto la ecuación para el calculo de SNR para una señal sinusoidal es [20]:

$$SNR_{sine}|_{dB} = (6.02 \cdot +1.78)dB$$

Número equivalente de bits (equivalent number of bits ENOB).

El número equivalente de bits (*Equivalent number of Bits ENB*) parte de la relación señal a ruido, que para una señal sinusoidal convencional muestreada a tasa Nyquist sería:

$$ENOB_{sin} = \frac{SNR|_{dB} - 1.78}{6.02}$$

Esta ecuación calcula el número de bits del convertidor, partiendo de su relación señal a ruido, por lo tanto, es otro parámetro para evaluar la resolución, calidad y precisión del convertidor, desde un enfoque más práctico y realista.

2.4. Tipos de convertidores.

Los convertidores de datos pueden clasificarse según diferentes características, pero una de las principales clasificaciones es la frecuencia con la que se muestrean los datos. De esta clasificación surgen dos vertientes: convertidores tasa Nyquist y convertidores de sobre muestreo. Los convertidores de tasa Nyquist son aquellos en los que su frecuencia de muestreo es seleccionada acorde al mínimo requerido para evitar aliasing según el teorema de Nyquist-Shannon [28]. Esta frecuencia es igual o ligeramente mayor al doble de la frecuencia máxima presente en la señal que se desea muestrear.

El segundo grupo de convertidores son aquellos que muestrean a una tasa mucho mayor de la frecuencia de Nyquist, siendo conocido como convertidores de sobremuestreo, estos convertidores muestrean a frecuencias mucho más altas del mínimo requerido, con el objetivo de esparcir el ruido en todo el rango de frecuencia, desde dc hasta f_s , de esta manera el ruido

presente en el ancho de banda de la señal disminuye y por consiguiente la potencia de este. La diferencia del límite en la frecuencia de muestreo f_s se explica gráficamente, en la Figura 2.6 se observa que la banda de transición es mucho mayor en el caso del sobremuestreo, dando lugar a una distribución de ruido sobre un mayor rango de frecuencia.



Figura 2.6: convertidor tasa Nyquist y sobre muestreo en el dominio de la frecuencia, recuperada de [20].

2.5. OSR y ENOB.

OSR

Partiendo de que $2f_B$ es el mínimo teórico planteado por Nyquist, la relación de sobre muestreo (Oversampling Ratio OSR) ejemplifica cuántas veces “de más” se está muestreando

una señal, ya que se obtiene el factor de sobremuestreo. Si el OSR es igual a 1, la señal se está muestreando a tasa Nyquist.

$$OSR = \frac{F_s}{2f_B}$$

ENOB

El número efectivo de bits de un convertidor de datos de sobre muestreo, representa su resolución real, considerando la presencia de ruido. El ENOB puede incrementarse al aumentar la tasa de sobremuestreo (OSR), esta relación se expresa como:

$$ENOB = n + 0.5 \cdot \log_2 \cdot (OSR)$$

De esta ecuación se deduce que, para mejorar la resolución en un bit, el OSR debe aumentar por un factor de 4. Es decir, para obtener un aumento de n bits en el ENOB, el OSR debe cumplir:

$$OSR = 4^n$$

El ENOB, también está ligado al ancho de palabra digital que entrega el convertidor de datos, al contar con más bits en la palabra digital, hay mayor resolución digital. Esta relación en un convertidor de sobre-muestreo suele rebasar la resolución base del cuantificador, ya que, la resolución efectiva (ENOB) mide cuántos bits de la salida digital del convertidor realmente representan información útil y esta mejora cuando el ruido se distribuye en un rango de frecuencia mayor.

2.6. ADC tipo Flash

El convertidor de datos analógico-digital tipo flash es el convertidor más rápido debido a su arquitectura simple y eficiente. Esta se basa en un arreglo de resistores que funcionan como divisores de voltaje del rango dinámico del convertidor. Es decir, reparten el voltaje total en varios niveles iguales. Para lograr eso, los resistores suelen tener el mismo valor, con el objetivo de que todos los niveles de cuantificación tengan el mismo tamaño. Esto permite dividir el rango completo en partes uniformes.

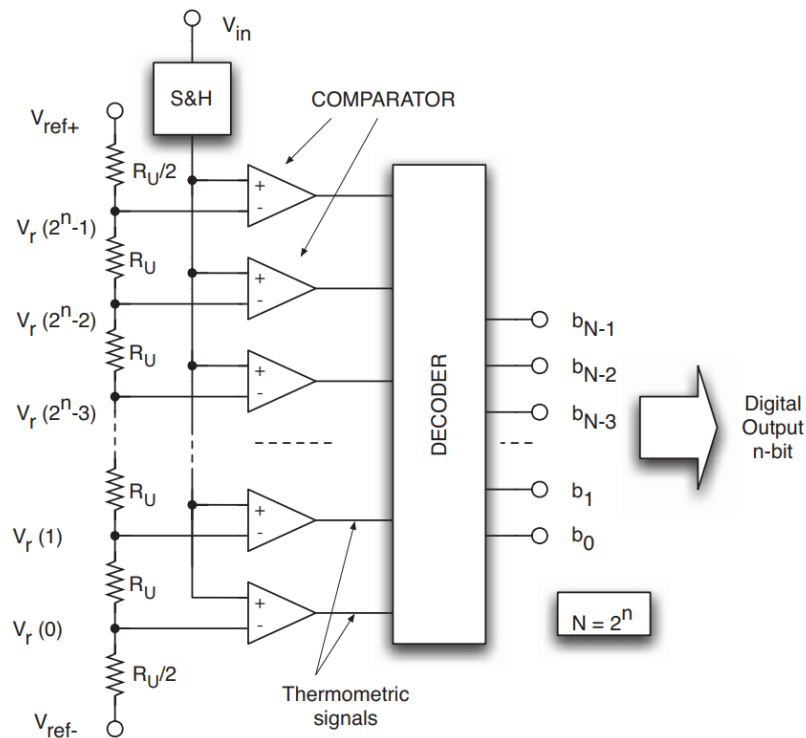


Figura 2.7: Circuito típico de un convertidor Flash, imagen recuperada de [20]

A estos niveles de referencia se conectan los comparadores, que son los encargados de analizar, al mismo tiempo, la señal de entrada. El número de comparadores que se necesitan es $2^n - 1$ donde n es el número de bits que tendrá la salida. Cada comparador revisa si la señal que se quiere convertir es mayor o menor que su voltaje de referencia. Si la señal es mayor, el comparador entrega un 1; si es menor, entrega un 0. Como todos los comparadores trabajan en paralelo, el proceso de conversión es casi instantáneo, lo cual hace que este tipo de arquitectura sea la más rápida de todas (Figura 2.7).

El resultado de estas comparaciones es un conjunto de unos y ceros que forman lo que se conoce como un código termométrico, donde los unos aparecen de forma continua hasta cierto punto y luego vienen solo ceros. Este tipo de código no es práctico para usar directamente, por lo que se necesita un decodificador que lo transforme en un número binario estándar, que es el formato más común para que lo interpreten otros sistemas digitales.

Gracias a esta forma de operar, el convertidor flash es ideal para aplicaciones donde se necesita una conversión muy rápida, como en video o equipos de comunicación. Sin embargo, a medida que se necesitan más bits de resolución, la cantidad de comparadores aumenta de forma exponencial, lo cual puede hacer que el circuito sea más costoso y ocupe mucho más espacio y consuma mayor potencia. Es por esto que, aunque es el más veloz, no siempre es el más conveniente para cualquier tipo de aplicación.

2.7. Análisis de respuesta en frecuencia.

El concepto de análisis en frecuencia, en esencia, se refiere a aplicar una entrada dada a un sistema y analizar la respuesta de este, mientras se varía la frecuencia en la entrada aplicada. Esto se debe a que el circuito no va a presentar el mismo comportamiento en todo el espectro de frecuencia. La respuesta en amplitud del sistema, normalmente mantiene su amplitud o amplifica su amplitud (depende de la configuración del circuito) pero, conforme la frecuencia va aumentando, eventualmente llega un punto en que la amplitud es atenuada como se muestra en la figura ilustrativa (Figura. 2.8). Esto se debe a que entre más alta sea la frecuencia, al circuito le cuesta más “seguirle el paso” a los cambios en la señal, esto debido a las características de los componentes físicos y pérdida de energía. Es por esto que se determina un ancho de banda útil de la señal f_{bw} el cual indica el punto en el cual la amplitud en la señal empieza a decaer (Figura. 2.9). Los análisis en frecuencia son útiles, ya que la demanda de llevar a los circuitos a frecuencias más altas ha requerido un sólido entendimiento de sus limitaciones de velocidad [29]. Por lo tanto, un análisis de frecuencia, evalúa el sistema al realizar un barrido de frecuencia.

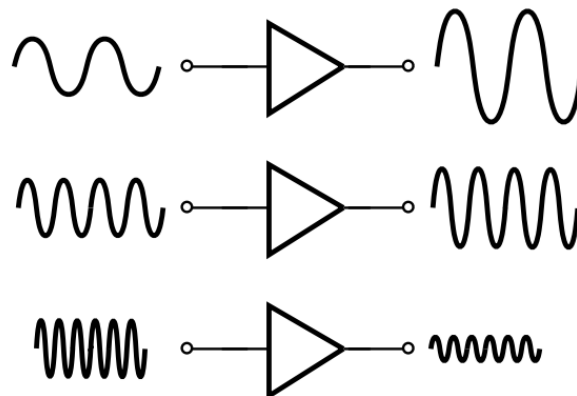


Figura 2.8: Imagen ilustrativa de la atenuación de amplitud a altas frecuencias, imagen recuperada de [29].



Figura 2.9: Imagen ilustrativa de la frecuencia de corte en la que se atenúa la amplitud e una señal, recuperada [29].

2.8. Conceptos clave en la transformada discreta de Fourier.

“La DFT permite analizar, manipular y sintetizar señales en maneras que no sería posible con procesamiento de señales continuas” [28]. La transformada discreta de Fourier, es una ecuación matemática que permite identificar los componentes en frecuencia que integran una señal discreta. Esta es representada por la siguiente ecuación en su forma rectangular:

$$X(m) = \sum_{n=0}^{N-1} x(n)[\cos(2\pi nm/N) - j\sin(2\pi nm/N)]$$

Donde:

- m = el índice de salida en el dominio de la frecuencia, su longitud varía desde cero hasta $N-1$.
- $X(m)$ = El componente m th de salida de la DFT.

- n = El índice en el dominio del tiempo de las muestras de entrada
- $x(n)$ = La secuencia de muestras de la entrada.
- N = El número de muestras de la secuencia de entrada y el **número de puntos de frecuencia** en la salida de la transformada discreta de Fourier (longitud de la secuencia, número de bins) [28].

Modulador Delta-Sigma.

3.1. Filtro integrador.

El integrador es un componente fundamental del modulador delta-sigma, ya que en este, ayuda a moldear el ruido (Figura 3.1).

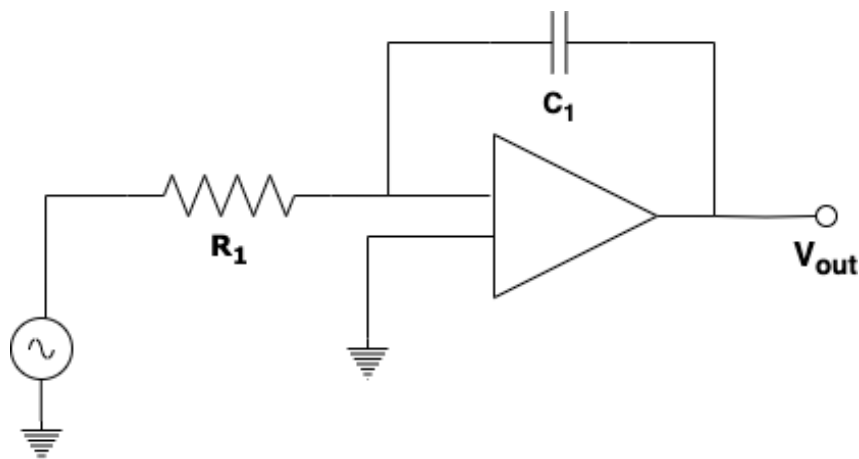


Figura 3.1: Circuito de filtro integrador estándar.

El circuito integrador básico se comporta como un filtro pasa bajas, ya que a bajas frecuencias la impedancia del capacitor aumenta, como se observa en la siguiente relación:

$$X_c = \frac{1}{2\pi fC}$$

Al aumentar la reactancia, la oposición al paso de corriente es mayor y el capacitor tiene tiempo de cargarse y descargarse adecuadamente, aunque de disminuir aún más la frecuencia la impedancia del condensador se vuelve demasiado grande, acercándose a un circuito abierto y reduciendo el circuito a la operación en circuito abierto [29]. Mientras que a altas frecuencias, los cambios tan abruptos en la corriente, impiden que las placas del capacitor se carguen adecuadamente, la carga que se acumula entre cada “subida y bajada” es tan pequeña que la

salida no detecta el cambio por lo que V_{out} tiene un cambio mínimo. Si se ve desde el punto de vista de la reactancia, a frecuencias altas, esta disminuye, por lo que el capacitor se comporta como un corto circuito, por lo tanto, la tierra virtual es reflejada a V_{out} enviando este a cero. La ecuación de voltaje de salida del integrador es la siguiente:

$$V_{out} = -\frac{1}{R_1 C_1} \int V_{in} dt$$

La cual se calcula realizando el análisis respectivo, asumiendo condiciones ideales.

3.2. Modulador Delta-Sigma.

Los convertidores A/D tradicionales tienen la desventaja de ser sensibles a las imperfecciones en el circuito, los moduladores $\Delta\Sigma$ carecen de esta desventaja, ya que muestran una sensibilidad muy baja a las no linealidades en la mayoría de las etapas que lo comprenden. [24]. El modulador Delta-Sigma, se describe como un convertidor de datos de sobre muestreo, cuyas características de: sobre muestreo, retroalimentación de la salida y filtros integradores, moldean el ruido a frecuencias altas, lejos de la banda útil de la señal, disminuyendo así la potencia de este, mejorando la SNR y el ENOB, lo que se traduce en mejor calidad en la señal de salida. Su arquitectura estándar está compuesta por un filtro integrador el cual filtra el error a frecuencias altas, un ADC que cuantifica y discretiza la señal, y un DAC interno en el bucle de retroalimentación que transporta la señal de salida a un punto suma ubicado en la entrada del filtro integrador, este DAC es necesario, ya que se debe convertir la señal a la forma de la señal de entrada para ser restadas adecuadamente (Figura 3.2).



Figura 3.2: arquitectura estándar de un modulador $\Delta\Sigma$ de primer orden.

Como se puede observar, este es un sistema no lineal y dinámico, esto es debido al efecto de cuantificación del ADC y a la memoria del integrador [31], ya que los sistemas no lineales son más difíciles de analizar y controlar, el sistema es linealizado “un sistema no lineal determinista se reemplaza por uno lineal estocástico” [24]. De esta manera, el error de cuantificación se modela como ruido de cuantificación, el cual es inyectado como una perturbación en el sistema. En términos rigurosamente estrictos, esta representación no es 100 % fiel al comportamiento del sistema, pero este modelo ha sido aceptado y empleado, por la comunidad científica a lo largo de muchos años, arrojado buenos resultados. De esta manera se obtiene el modelo lineal estándar de la Figura 3.3.

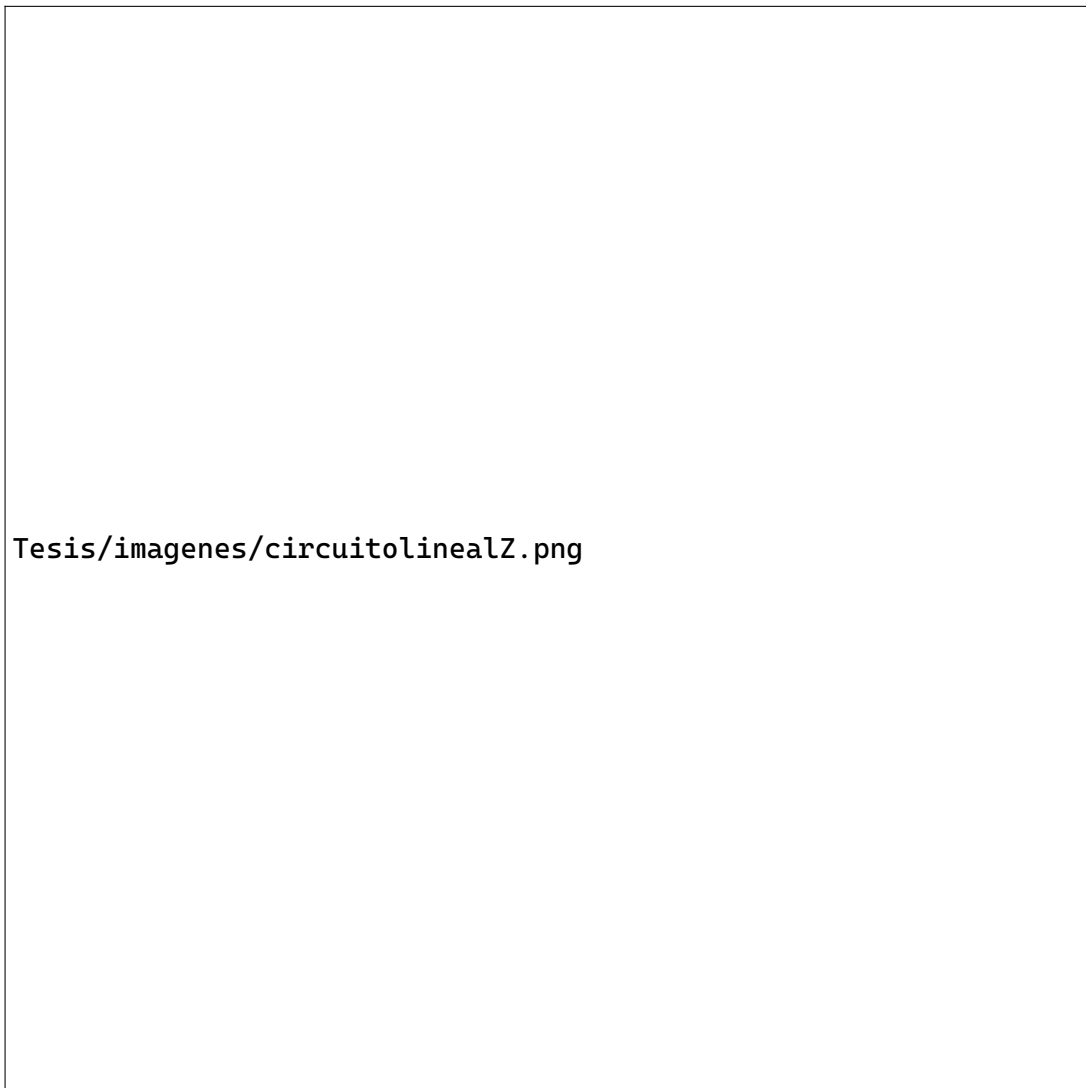


Figura 3.3: Circuito modulador delta-sigma de primer orden linealizado.

Desde el punto de vista del control, los ADC convencionales son sistemas de lazo abierto, la señal de salida es función directa de la señal de entrada, la cual es afectada por el ruido. Por lo tanto, los moduladores Delta-Sigma implementan un lazo de retroalimentación, que tiende a corregir el error, dando lugar así a un sistema control de lazo cerrado. Al tratarse de sistemas lineales, y según el principio de superposición, la respuesta del sistema ante la suma de dos entradas es igual a la suma de las respuestas individuales que el sistema tendría ante cada entrada por separado. Por lo tanto, al aplicar dos señales simultáneamente, la salida del sistema corresponde a la suma de las salidas que se obtendrían si cada señal se aplicara de forma individual.

Por lo tanto, cuando se presentan dos entradas (entrada al sistema y perturbación) en un sistema de lazo cerrado en un sistema lineal, cada una de estas es tratada de manera independiente. Para el análisis de la perturbación se asume que el sistema está inicialmente relajado (no hay

entrada aplicada) y se calcula la respuesta $Y_e(Z)$ únicamente para la perturbación. Por lo que el diagrama de bloques, queda de la siguiente manera:



Figura 3.4: Diagrama de bloques para el cálculo de la respuesta al error.

De donde se deriva la función de transferencia para el ruido:

$$Y_e(z) = E(z) + [H(z) \cdot -Y_e(z)]$$

$$Y_e(Z) = \frac{E(z)}{1 + H(z)}$$

Por otra parte, para el cálculo de la salida $Y_u(Z)$ asociado a la entrada, se asume que la perturbación tiene un valor de cero, por lo que se asume el diagrama de bloques de la siguiente manera (Figura.3.5).

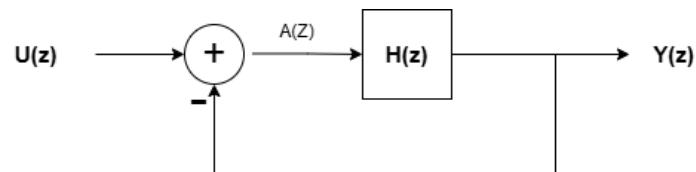


Figura 3.5: Diagrama de bloques para el cálculo de respuesta aplicada en la entrada.

De donde se deriva la función de transferencia que responde al estímulo de la entrada:

$$\begin{aligned} Y_u(z) &= H(z)[U(z) - Y_u(z)] \\ Y(z)(1 + H(z)) &= H(z)U(z) \\ Y_e(Z) &= \frac{U(z) \cdot H(z)}{1 + H(z)} = \frac{1}{\frac{1}{H(z)} + 1} \end{aligned}$$

Como se ha explicado, la respuesta a la aplicación simultánea de la entrada y la perturbación se obtiene sumando las dos respuestas individuales [33]. La respuesta $Y(Z)$ producida es:

$$Y(Z) = Y_e(z) + Y_u(Z)$$

En el sistema completo (Figura. 3.3), la salida $Y(Z)$ es realimentada al punto suma, donde se compara con la entrada de referencia $U(Z)$. De esta manera, se obtienen dos funciones de transferencia, una asignada al ruido NTF y otra asignada a la señal de entrada STF .

3.2.1. STF Y NTF.

El comportamiento deseado del modulador es que el ruido no este presente en el ancho de banda, sin afectar la señal esperada a la salida. Por lo tanto, se propone que la ganancia de la NTF sea alta en las frecuencias fuera de la base, ya que esta describe el comportamiento del ruido, mientras que la STF debe tener una ganancia baja en la banda base, definiendo así una NTF ideal como la inversa de una STF ideal: [24]

$$NTF = \frac{1}{STF} = \begin{cases} 0, & -f_B \leq f \leq f_B \\ 1, & \text{para el resto de los valores} \end{cases}$$

Esta ecuación establece que el valor de la ganancia de NTF será 0 para las frecuencias contenidas en la banda base, mientras que será 1 en las frecuencias fuera de este rango. Mientras que la ganancia de STF se comporta de manera inversa, 1 para las frecuencias dentro de la banda base, y 0 para frecuencias mayores. Comportándose como un filtro pasa altas para el ruido y un filtro pasa bajas para la señal. Por consiguiente, el ruido de cuantificación es moldeado al rango de frecuencias mayor a f_B .

3.2.2. Moldeado de ruido.



Figura 3.6: Imagen ilustrativa del moldeado de ruido, obedeciendo al orden del modulador, imagen recuperada de [9].

El moldeado de ruido constituye la médula espinal de un modulador delta-sigma. Este proceso permite redistribuir el ruido de cuantificación fuera de la banda útil, mejorando así la relación señal-ruido en las frecuencias de interés. El moldeado de ruido se logra gracias a la combinación de varias etapas:

- Sobre-muestreo, que incrementa la tasa de muestreo para desplazar el ruido de cuantificación hacia frecuencias más altas.
- Retroalimentación, que permite al sistema anticipar y compensar el error de cuantificación en las muestras siguientes.
- Integración, en la cual se acumula el error de cuantificación a lo largo del tiempo, reforzando el efecto de moldeado.

En cada ciclo, la señal de salida del modulador es comparada con la señal deseada (la entrada), y la diferencia resultante (el error) se realimenta e integra. Esto da como resultado una señal de salida que contiene la señal original deseada, junto con una versión modificada del error de

cuantificación.

Cabe destacar que, en general, cada integrador adicional en la arquitectura del modulador incrementa el orden del sistema de moldeado de ruido, aumentando la pendiente de atenuación del ruido fuera de la banda útil. Esto significa que el ruido de cuantificación se concentra cada vez más en las frecuencias altas, como se observa en la Figura.3.6. Facilitando su posterior eliminación mediante un proceso de decimación, el cual consta de aplicar un filtrado digital y submuestrear la señal. El filtro digital se encarga de eliminar el ruido presente entre f_B y f_s , permitiendo así que el procesamiento posterior sea únicamente con la señal útil, pero, al ser este un proceso puramente digital, es recomendable submuestrear la señal, para disminuir la frecuencia de muestreo f_s . Esto, debido a que el procesamiento digital, ocupa relojes de altas frecuencias que consumen demasiada potencia, y ya que el ruido de la señal ha sido eliminado, es conveniente disminuir la frecuencia de la señal a aproximadamente la típica frecuencia de Nyquist. Disminuyendo el tamaño de los datos de salida, al tomar una muestra, cada k muestras.

Con base en este concepto, se propone que el grado de integrador diseñado a lo largo del documento, sea un modulador de segundo orden, ya que este moldea con una pendiente más prominente que un modulador de primer orden, pero sin llegar a desestabilizarse como sería el caso de los moduladores iguales o superiores al tercer orden.

3.3. Moduladores de tiempo continuo y tiempo discreto

La principal característica que diferencia un modulador de tiempo continuo (CT *continuous-time*) y uno de tiempo discreto (DT *Discrete-time*) radica en la ubicación de la etapa *sample and hold* que es la encargada de muestrear la señal. En moduladores CT esta se encuentra, en la entrada del cuantificador, como se muestra en la imagen (Figura.3.7).



Figura 3.7: Modulador de tiempo continuo, imagen recuperada de [34].

En otras palabras, el modulador de tiempo continuo, se basan en el desplazamiento del muestreador dentro del bucle del modulador, esto se traduce en que la señal que pasa a través del filtro integrador es una señal continua. Además, el dominio en que se implementa el circuito del filtro en un CT, es circuitería y diseño analógico.

Mientras que en moduladores de tiempo discreto, la etapa de muestreo se encuentra antes del proceso de modulación (Figura. 3.8) este es aplicado fuera del bucle del modulador, en la entrada del sistema, por lo que técnicamente no forma parte del circuito del modulador, quien, todo el tiempo opera con señales discretizadas. Se puede observar que el filtro en el circuito, se construye en un dominio digital, por lo que obedece a los ciclos de reloj del sistema.



Figura 3.8: Modulador de tiempo discreto, imagen recuperada de [34].

Las ventajas clave de los moduladores $\Delta\Sigma$ de tiempo continuo (CT) sobre los de tiempo discreto (DT) son su mayor eficiencia energética y la posibilidad de alcanzar frecuencias de muestreo más altas. En los moduladores DT, el bloque de muestreo y retención (*S&H*) es implementado típicamente con interruptores *switched-capacitor* (*SC*) o bloques digitales secuenciales. Este enfoque requiere el uso de conmutadores que operan a alta velocidad, así como una red de relojes internos veloces, lo cual conlleva un mayor consumo de potencia. Además, el mismo proceso de conmutación de relojes, impone una restricción sobre la frecuencia máxima de muestreo alcanzable.

Mientras que los moduladores CT procesan la señal en el dominio analógico de manera continua, utilizando integradores. Todo el lazo de retroalimentación opera sin muestreo ni reloj, y el único punto donde se realiza una conversión discreta es justo antes del cuantificador, mediante un bloque de muestreo mucho más sencillo. Gracias a esto, los moduladores CT pueden operar a frecuencias de muestreo del orden de los GHz, ya que no dependen de relojes rápidos ni de switches conmutando constantemente. Esta característica les permite alcanzar frecuencias de muestreo mucho más altas.

Como consecuencia de estas limitaciones estructurales, los moduladores DT, que también requieren sobre-muestreo para obtener una buena relación señal-ruido (SNR), han quedado restringidos a anchos de banda relativamente estrechos, mientras que los moduladores CT se han vuelto la opción preferida para aplicaciones de alta velocidad y bajo consumo.

Aunque los $\Delta\Sigma$ CT cuentan con sus propias complicaciones, mayor sensibilidad al jitter de reloj, retrasos en la realimentación que pueden llegar a afectar la función de transferencia del modulador, y puede llegar a presentar errores debido a las respuestas transitorias impredecibles [24].

CAPÍTULO 4

Diseño del modulador delta-sigma

4.1. Ejemplos de arquitecturas existentes.

Las arquitecturas utilizadas en diseños previos de moduladores delta-sigma varían según el enfoque de cada investigador, pero tomando en cuenta sus características principales, se pueden clasificar en los siguientes tipos [23]:

- Cascada de integradores con retroalimentación distribuida.
Como su nombre lo indica la arquitectura se basa en conectar una cantidad N de integradores (normalmente acompañado de una ganancia de escalamiento a_n) en cascada (aumentando así el orden del modulador) y conectando cada uno de ellos a retroalimentación proveniente de la salida del DAC, esto con el objetivo de aumentar el OSR y consecuentemente aumentar el número efectivo de bits en el modulador. Pero, existe el riesgo de que este sufra mayor inestabilidad, debido a la cantidad de integradores presentes en él (Figura 4.1).



Figura 4.1: Configuración cascada de integradores con retroalimentación distribuida, imagen recuperada de [24].

- Cascada de integradores con suma de trayectoria directa ponderada.
Esta arquitectura igual que la anterior, conecta N integradores en cascada, pero, en vez de contar con retroalimentación directa cuenta con trayectorias de control adelantado (*feed forward*) las cuales conectan la salida de cada integrador a la entrada del cuantificador (normalmente multiplicando la señal por una ganancia a_n) Figura. 4.2. Mientras que, cuenta con una única retroalimentación que va desde la salida del DAC a la entrada del primer integrador. Este tipo de arquitectura proporciona más libertad de diseño en la selección de las ganancias.



Figura 4.2: configuración de cascada de integradores con suma de feed-forward ponderada. Imagen recuperada de [24].

- Cascada de integradores con retroalimentación y feed-forward distribuidos.
Esta arquitectura combina las dos anteriores, ya que cuenta con N integradores conectados en cascada, cada uno con una retroalimentación proveniente de la salida del DAC que es escalada por un coeficiente a_i , y a la vez, cada integrador recibe la señal de entrada escalada por un coeficiente a_{sigi} mediante una ruta de feedforward.
Al incluir tanto N trayectorias de retroalimentación como N trayectorias de control adelantado, se obtiene un mayor grado de libertad en el diseño, lo que permite ajustar el comportamiento de cada etapa de manera más flexible. Esta capacidad de control mejora el desempeño del modulador, aunque también implica un mayor consumo de potencia y área en silicio.



Figura 4.3: Configuración cascada de integradores con retroalimentación y feed-forward distribuidos, imagen recuperada de [24].

Como se puede observar, la variación entre estas arquitecturas radica principalmente en la cantidad de trayectorias de retroalimentación y de control adelantado presentes en cada una, las cuales permiten un mejor control en el comportamiento del modulador, aunque también implican un mayor consumo de recursos.

4.2. Propuesta de diseño ideal.

El diseño propuesto parte de las especificaciones mostradas en el Capítulo 1, sin embargo, se pretenden mostrar las ventajas de los moduladores en tiempo continuo. El diseño propuesto se basa en una arquitectura CIDFBFF (*Cascada de integradores con retroalimentación y feed-forward distribuidos*). Ya que, es esta la que logra controlar de una manera más independiente y clara la estabilidad del modulador. Por lo cual se espera que el estudiante alcance mejores resultados que con otro tipo de arquitecturas.

Primeros pasos.

Una vez plantada la topología básica del modulador Delta-Sigma de segundo orden en capítulos anteriores, y seleccionada una arquitectura de retroalimentación de partida, se realizan los primeros prototipos de diseño. Estos prototipos, en una primera etapa, analizan el comportamiento matemático del circuito (no su comportamiento electrónico), tomando en cuenta que cada componente presente en el sistema obedece a las ecuaciones matemáticas que describen su funcionamiento. Este enfoque inicial es necesario para verificar si el razonamiento lógico-matemático sobre el cual se basa el diseño es adecuado, ya que, de fallar este, es imposible trasladar el diseño a un simulador electrónico. Si, por el contrario, se decidiera testear directamente en entornos de simulación electrónica, de existir fallas, es aún más complicado rastrear si el error es originado en algún defecto electrónico (conexiones equivocadas o rotas, errores de convergencia, etc.) o si se origina más bien, desde el planteamiento y resolución del problema. Por lo que realizar primero el circuito a nivel comportamental-matemático permite disminuir errores, y de existir estos disminuye el campo de opciones de donde estos pueden ser originados. Es por esto, que las primeras pruebas de diseño fueron realizadas en el software MATLAB-simulink el cual cumple con las características comportamental-matemáticas mencionadas previamente. Además de simplificar el proceso de selección de ganancias en los integradores, la retroalimentación y en el lazo de control directo, gracias a su rapidez y facilidad de realizar un análisis espectral al circuito.

4.2.1. Consideraciones de estabilidad

Las consideraciones de estabilidad hacen referencia a los valores asignados en las ganancias que acompañan a la retroalimentación, a los integradores y en el lazo de control adelantado, las cuales ayudan a regular el comportamiento del modulador.

- Ganancias de retroalimentación.
 - Ayudan a que el sistema no oscile ni se satura.
 - Ayudan a moldear el ruido.
- Ganancias presentes en el lazo de alimentación directa. (*Trayectoria directa*).
 - Estas ganancias llevan parte de la señal de entrada directamente a ciertos puntos del modulador.
 - Corrigen distorsiones introducidas por los integradores.
- Ganancias presentes en los integradores.
 - Ayudan a que el rango de salida de cada integrador no se dispare.
 - Definen la región del espectro donde el modulador trabaja adecuadamente.

El cálculo de estas ganancias parte de las funciones de transferencia pertenecientes al ruido (NTF) y la señal de entrada del modulador (STF), pero en la práctica se ha observado que estos valores no son exactamente adecuados, son más una guía. En la práctica, el estudiante observe

que para el cálculo correcto de estos es necesario simular y observar el comportamiento del modulador.

Para el cálculo de ganancias, se establece al modulador delta-sigma en dominio Z (para señales discretas), lo que se traduce en un modulador de tiempo discreto (DT). Esto es debido a que el análisis en tiempo discreto, no necesita un factor de escalamiento para el cálculo de la magnitud del integrador, ya que este es independiente de F_s debido a que la naturaleza inherente de un DT está ligada al tiempo de cuantificación. Es por esto que tiene menor sensibilidad a los cambios, por lo que calcular las ganancias de estabilidad se vuelve un proceso más directo (Figura 4.4).

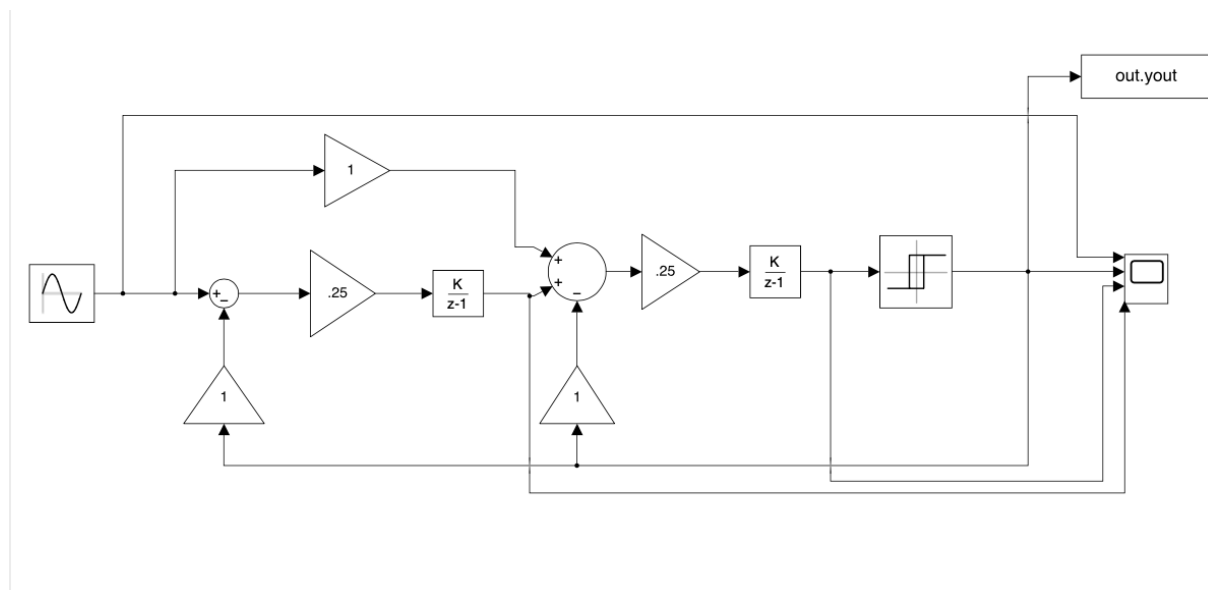


Figura 4.4: Prototipo de modulador DS en tiempo discreto de 1 bit.

El diseño del modulador comenzó asignando valores tentativos a las ganancias, los cuales se fueron adaptando según la respuesta a las simulaciones y el comportamiento del modulador: que moldee adecuadamente el ruido, no se saturen los integradores, que tenga una SNR y número efectivo de bits adecuados y comprobar que la respuesta del sistema presente la menor cantidad posible de armónicos dentro de la banda útil. Es por esto que el primer prototipo a realizar fue de un solo bit. Ya que encontrar las ganancias adecuadas en este vuelve más sencilla la selección de ganancias para un modulador multi-bit.

Una vez que se identifican las ganancias que mantienen al integrador estable, evitando que se saturen, el modulador en tiempo discreto es transportado a tiempo continuo. Esto es debido a que en la aplicación el estudiante desea aplicar un (delta-sigma DS) de tiempo continuo debido a sus ventajas, menos consumo de potencia, el CT contiene menos ruido y la simplicidad del circuito al no ocupar filtro anti-alias. Por lo que se adecúa el circuito cambiando los integradores discretos a integradores continuos, en la práctica esto se traduce a integradores digitales o analógicos, y multiplicándolos por la frecuencia de muestreo, para ajustarlos al valor de frecuencia necesario.

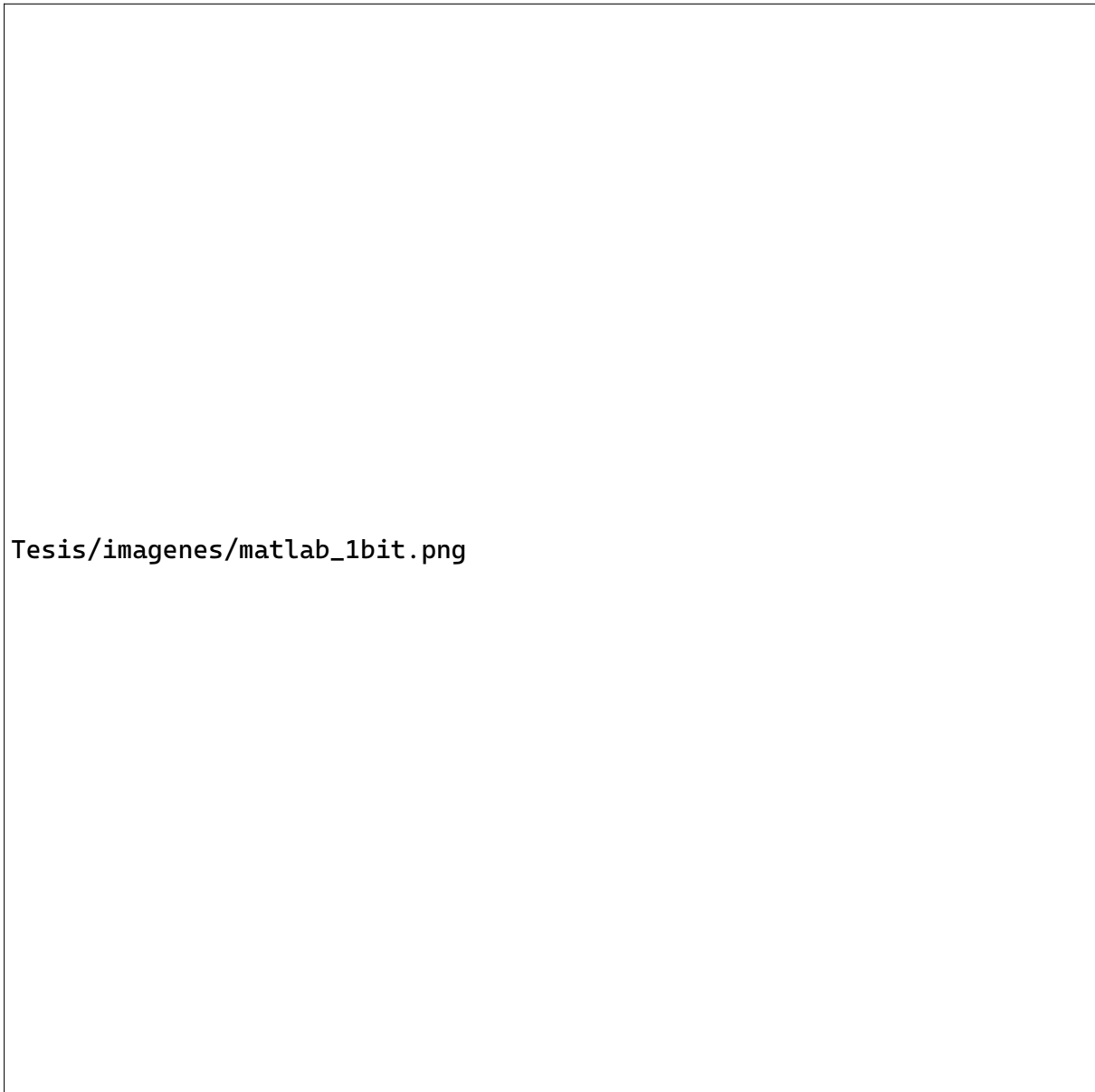


Figura 4.5: Arquitectura de un modulador Delta-Sigma de un solo bit propuesta por el estudiante.

4.3. Implementación del circuito en MATLAB-simulink CT.

Es importante destacar, que el estudiante comenzó la preparación de su tema de tesis, desde un estudio básico del procesamiento de señales. Para ello, abordó conceptos fundamentales como señales continuas, discretizadas, transformada de Fourier (entre otros), pero esto no fue únicamente desde conceptos teóricos, sino también acompañado de simulaciones,

interpretación y comprensión de sus respectivos análisis en el dominio de la frecuencia en la herramienta MATLAB. Estas actividades sirvieron para familiarizarse con palabras clave, funciones específicas y configuración de la herramienta. Además del desarrollo de habilidades para interpretación de resultados, si bien esta etapa de formación no se refleja de forma directa en los resultados finales, resulta relevante mencionar el conjunto de tareas que contribuyeron al proceso, diseño y simulación.

Es por esto que al diseñar el prototipo del modulador delta-sigma en CT, el estudiante ya contaba con estos recursos técnicos. El modelo se compone de dos etapas complementarias, el script elaborado en MATLAB en el cual se plantean condiciones iniciales (las cuales son empleadas por el diagrama a bloques) y donde se realiza el análisis en frecuencia. La segunda etapa es el diagrama de bloques desarrollado en MATLAB-simulink en el cual radica la configuración del modulador. A continuación se presenta un breve resumen del funcionamiento de cada etapa.

Script en Matlab.

El código completo puede encontrarse en el apéndice A de esta tesis. En este, se encuentran las especificaciones del modulador: número de bits del cuantificador, frecuencia de muestreo, relación de sobre muestreo de la señal (OSR), ancho de banda en el que el modulador trabaja eficientemente, frecuencia de la señal de entrada aplicada, entre otras especificaciones que ayudan a evitar el derrame espectral para un mejor análisis en frecuencia (como número de bins en la FFT y número de periodos evaluados por la misma). Estas especificaciones son enviadas a Simulink, el cual devuelve la salida del diagrama de bloques al script. Como dato informativo se menciona que es importante que el archivo script y el archivo .slx generado por simulink, se encuentren en la misma carpeta de archivos, para que estos puedan ser vinculados adecuadamente. Además, señalar que hay pequeñas variaciones en el código encargado de leer los datos que regresa Simulink, según la versión de MATLAB con la que se trabajó.

Una vez que simulink ha enviado a MATLAB un vector de salida, se establecen parámetros que se utilizarán para un adecuado análisis de barrido en frecuencia, se empieza estableciendo un vector de ventaneo, el cual ayuda a disminuir el derrame espectral, se normaliza la frecuencia, se calcula cuántos bins de la FFT cubren el ancho de banda útil, se inicializa el vector $yy1$ con una dimensión de $(1,N)$, una vez inicializado el vector $yy1$ se le asignan los valores del vector de salida enviados por simulink, desde 1, hasta $N+1$, luego se inicializa otro vector, llamado $ptot$, con las mismas dimensiones $(1,N)$ y finalmente se calcula la SNR y el espectro de densidad de potencia (utilizando la FFT) a través de la función $calcSNR$, luego se convierte el SNR a resolución equivalente en bits usando la fórmula teórica. El objetivo de este código es visualizar el contenido espectral de la señal y evaluar el rendimiento en términos de SNR y resolución efectiva en bits.

4.3.1. Diagrama de bloques en simulink.

El diagrama de bloques elaborado en simulink toma como antecedentes de diseño, la teoría establecida en secciones anteriores. La descripción del modelo es la siguiente: una entrada sinusoidal continua, la cual está conectada a un punto suma donde se sustrae la señal de retroalimentación (la salida del modulador) a la señal de entrada, luego es multiplicada por

dos ganancias distintas, una de 0.25 y otra de F_s , estas ganancias fueron calculadas simulando diferentes ganancias y observando cuál de ellas, estabilizaba de una manera más óptima al modulador. Una vez multiplicada por las ganancias, la señal es integrada en el bloque $\frac{1}{s}$, esta etapa acumula el error entre la entrada y la salida cuantificada, funcionando como primer integrador. Posteriormente, esta señal es enviada a otro punto suma, donde se suma la señal que viene desde entrada, la cual es multiplicada con una ganancia unitaria con el objetivo de que preserve mejor la señal. Este adelanto de la señal, sirve para contrarrestar las perturbaciones del circuito y además tiene el beneficio de acelerar la respuesta del circuito. En este mismo punto suma, nuevamente se resta la salida del cuantificador, recordemos que la retroalimentación en el circuito es clave para un adecuado moldeo del error de cuantificación (*noise shaping*).

La señal de salida del segundo integrador, vuelve a ser multiplicada por una ganancia de 0.25 y f_s respectivamente, estas ganancias además, evitan que el integrador $\frac{1}{s}$ se sature, como últimas etapas, la señal es muestreada por un bloque *S&H*, aquí se puede observar claramente que el circuito diseñado por el estudiante es un modulador Delta-Sigma de tiempo continuo, ya que comienza a trabajar con señales discretizadas, después de integrarlas. Y finalmente la señal es cuantificada con un cuantificador de 1 bit, que en el caso de simulink, es mejor colocar un comparador, un comparador. La señal de salida del modulador, es enviada nuevamente al script a través del bloque out.yout, donde, como se ha mencionado previamente, será analizada espectralmente.

Análisis en frecuencia del primer prototipo de 1 bit.



Figura 4.6: Análisis en frecuencia del modulador DS diseñado en MATLAB-simulink de 1 bit.

La gráfica proporcionada por MATLAB muestra el comportamiento de la señal desde DC hasta $f_s/2$, cuyos ejes son frecuencia vs. densidad espectral de potencia (esta medida en decibelios) cuyo pico de potencia está ubicado aproximadamente 5 kHz. Como se puede observar, los armónicos o ruido presentes en la banda útil de la señal contienen muy poca potencia, en contraste con la potencia presente en la señal de entrada. Conforme el eje de frecuencia tiende a aumentar, también lo hace la potencia del ruido, este, a pesar de tener mucho mayor potencia, está presente únicamente en frecuencias mucho más altas que la señal útil, por lo tanto, puede ser eliminado con un filtro pasa baja, obteniendo como resultado la señal útil con mucho menos ruido y distorsiones. El moldear el ruido de cuantificación a frecuencias altas es justo el objetivo del *noise shaping* y del modulador Delta-Sigma, por lo tanto, es visualmente apreciable que el modulador está cumpliendo su propósito, eliminando el ruido de la señal de interés.

Por otro lado, en el análisis cuantitativo del modulador, los parámetros de evaluación son la SNR y el ENOB. Se observa una SNR relativamente alta, mayor a 50 dB, y un ENOB mucho mayor al número de bits reales presentes en el cuantificador, lo cual evidencia una mejora efectiva en la resolución gracias al sobre muestreo y al modelado del ruido. En cuanto a datos no tan favorables, se observa que la gráfica tiene una pendiente mucho menor de la que se estima en moduladores de segundo orden, sin embargo, en general, el comportamiento mostrado por la PSD indica que el sistema está operando correctamente dentro de los principios esperados de la modulación Delta-Sigma.

4.3.2. Diagrama a bloques del modulador de tres bits.

Una vez que el estudiante comprueba el adecuado rendimiento del primer prototipo de 1 bit, y basándose en la misma arquitectura, se hacen los cambios pertinentes para convertirlo en un modulador de tres bits. Esto se debe a que, al utilizar un modulador multi-bit, los parámetros de desempeño como la SNR y el ENOB mejoraran significativamente, lo que se traduce en una mayor resolución en el proceso de cuantificación. Además, se obtendrá una mayor fidelidad en la señal reconstruida.

Otro aspecto clave es que al incrementar el número de bits, se disminuye la potencia del ruido de cuantificación sin necesidad de aumentar la OSR, lo cual es beneficioso tanto desde el punto de vista del rendimiento como del consumo de recursos. Según la ecuación de potencia del ruido de cuantificación, esta disminuye exponencialmente conforme se incrementa el número de niveles de cuantización, el cual, aumenta proporcionalmente con el número de bits en el cuantificador. Esto permite alcanzar niveles de precisión comparables a los de moduladores de orden superior, pero sin necesidad de complicar la estructura del lazo de realimentación ni incrementar excesivamente la frecuencia de muestreo. Por lo tanto, la implementación del modulador de 3 bits representa una evolución natural del diseño inicial, orientada a optimizar el desempeño sin comprometer la simplicidad de la arquitectura original.

El primer paso en la elaboración del modulador delta-sigma de tiempo continuo de 3 bits, basado en el primer prototipo, fue cambiar el bloque del comparador, por un bloque cuantificador, donde se asigna la cantidad de bits deseados desde el script de MATLAB. Posteriormente, se evaluó el diseño con las ganancias del circuito anterior, pero los resultados arrojados por el modulador, no tenían el rendimiento esperado. Es por esto, que se decide cambiar el valor de las ganancias en los integradores, a base de distintas simulaciones, el estudiante seleccionó el valor de 0.75 para ambos integradores.



Figura 4.7: Diagrama de bloques en simulink del modulador $\Delta\Sigma$ de 3 bits.

El proceso que sigue la señal es el mismo que se describe en el modulador de 1 bit, únicamente cambia la ganancia en los integradores, para evitar la saturación de estos. El modelo final, utilizado en MATLAB-simulink, se encuentra en la Figura 4.7. En cuanto al script de MATLAB, asociado a este, también fue basado y modificado a partir del diseño de 1 bit. El código puede encontrarse en la sección de apéndices. Los cambios a destacar son el número de bits y el ancho de banda, ya que en el primer prototipo, el estudiante se centró más en el comportamiento del circuito que en las especificaciones deseadas. Teniendo en cuenta que esta segunda versión es la versión final, al menos del comportamiento ideal-matemático (cuya arquitectura será simulada en Cadence) este ya debe incluir las especificaciones de ancho de

banda planteadas en los objetivos del documento.

Análisis en frecuencia del modulador de 3 bits

Con el análisis en frecuencia se obtiene la gráfica de la Figura 4.8, la cual muestra en el eje x un barrido en frecuencia, mientras que el eje y representa la densidad espectral de potencia (DSP) la cual es medida en decibelios por Hertz (dB/Hz). Como se ha mencionado anteriormente, la DSP permite analizar cómo se distribuye la potencia de la señal a lo largo del dominio de la frecuencia, en este caso desde DC hasta $f_s/2$. En esta gráfica se aprecia que la densidad espectral en el ancho de banda útil de la señal es muy baja, lo que se traduce en menos distorsión y pérdida de información. La potencia de la señal útil permanece concentrada en un pico claro, en el orden de los kHz, mientras que el ruido ha sido desplazado a frecuencias más altas. Esto se manifiesta en la pendiente ascendente del espectro, indicando que la energía del ruido ha sido moldeada fuera de la banda de interés, cumpliendo así el objetivo del noise shaping, con la finalidad de ser posteriormente eliminada mediante un filtro pasa bajas.

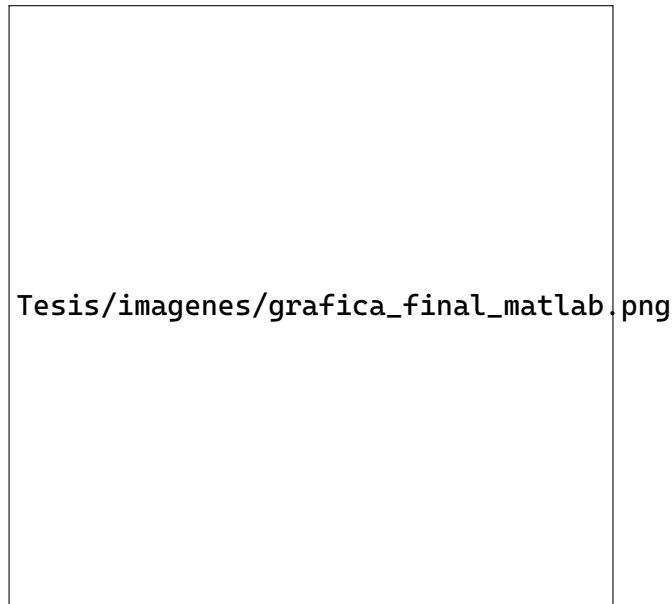


Figura 4.8: Análisis en frecuencia del modulador de 3 bits DS diseñado en MATLAB-simulink.

A diferencia del primer prototipo, este diseño presenta un moldeado de ruido mucho más efectivo, reflejando mejor el segundo orden del modulador. Como resultado, la SNR ha mejorado de forma significativa, alcanzando un valor de 85.3 dB. Asimismo, el número de bits efectivos (ENOB) calculado es de 13.88 bits, superando ampliamente el mínimo deseado de 10 bits para la aplicación. Esta mejora evidencia la efectividad del uso de un cuantificador multi-bit combinado con una arquitectura de segundo orden para incrementar la resolución sin necesidad de aumentar demasiado la OSR. Una vez que el diagrama de bloques elaborado en Matlab, comprueba el adecuado funcionamiento del circuito, a partir de los parámetros de moldeado de ruido, SNR y ENOB. Y que se seleccionan las ganancias adecuadas para la estabilidad del sistema, este es trasladado a su contraparte electrónica.

Resultados.

5.1. Implementación de macro modelo en Cadence-Virtuoso.

Una vez que el diagrama ideal elaborado en MATLAB muestra el rendimiento deseado, se desarrolla el circuito a un nivel comportamental, en la plataforma de diseño de sistemas electrónicos Cadence Virtuoso, sustituyendo los elementos del diagrama de bloques por sus componentes electrónicos equivalentes (Figura. 5.1). Es decir, los bloques de integradores son sustituidos por circuitos analógicos integradores, el cuantificador por un convertidor analógico-digital tipo flash, el DAC por un decodificador con salida decimal con una ganancia de amplificación y las ganancias del circuito por los cálculos proporcionales de resistencias y capacitores.

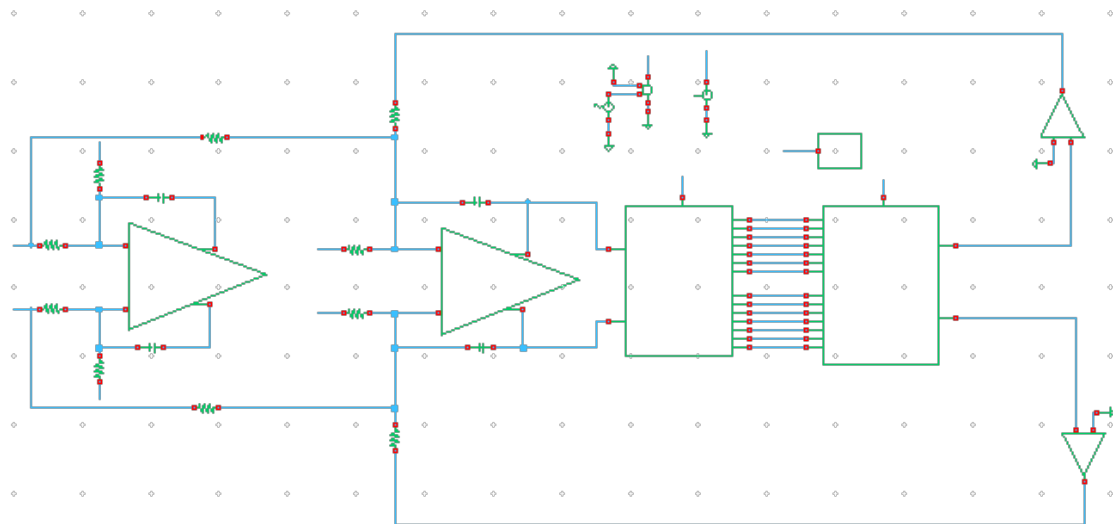


Figura 5.1: Modulador Delta-Sigma de segundo orden, elaborado en CADENCE virtuoso.

Cuando se dice que el circuito se establece a un nivel comportamental, es debido a que los componentes más complejos como el comparador, el decodificador y el amplificador, fueron

diseñados en Verilog-A, o modelados a su nivel comportamental a partir de otros elementos electrónicos, estos no fueron diseñados a nivel transistor. A continuación se desglosan los elementos más importantes del circuito, los cuales fueron diseñados por el estudiante, junto con una leve explicación del proceso de diseño.

Amplificador *fully differential*.

El amplificador empleado en el integrador, es un *fully differential* (Figura.5.2) el cual cuenta con dos entradas y dos salidas, la diferencia entre este tipo de op amp y los amplificadores convencionales *single ended* es que, los segundos, únicamente proporcionan una salida. Se decide trabajar con este tipo de amplificador, ya que el estudiante sí realizó el circuito con amplificadores *single ended*, pero el rendimiento presentado en este circuito no alcanzaba los parámetros de calidad esperados en su contraparte comportamental-matemática, elaborada en MATLAB. Por lo tanto, se optó por un amplificador *fully differential* el cual, por sí sólo, cuenta con una etapa de eliminación del ruido, como se ha explicado anteriormente.

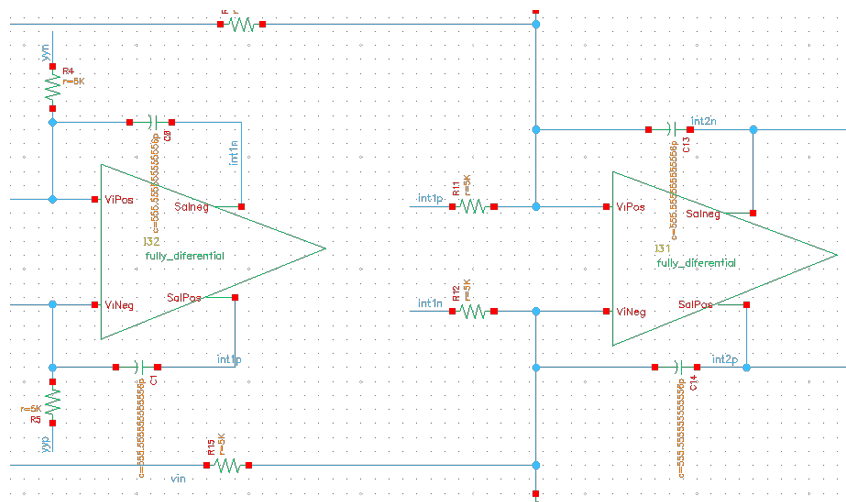


Figura 5.2: Circuitos integradores, elaborados con amplificadores *completamente diferenciales* ideales.

Como se ha mencionado, este circuito no fue diseñado a nivel transistor, más bien fue modelado basándose en sus características ideales. El amplificador operacional ideal se comporta como una fuente de voltaje controlada por voltaje (con una ganancia en voltaje e impedancia de entrada infinitas y una impedancia de salida 0 a la salida, entre otras características) este amplificador no es la excepción, un op amp *fully differential* también puede ser modelado como una fuente de voltaje controlada por voltaje, al contar con dos salidas (las cuales son simétricas), cada una es relacionada a una terminal (positiva y negativa) de la fuente amplificada, las cuales proporcionan las salidas deseadas (Figura. 5.3). El circuito cuenta con un capacitor conectado a tierra para evitar problemas de convergencia, los cuales impidan que el modelo pueda ser simulado adecuadamente.

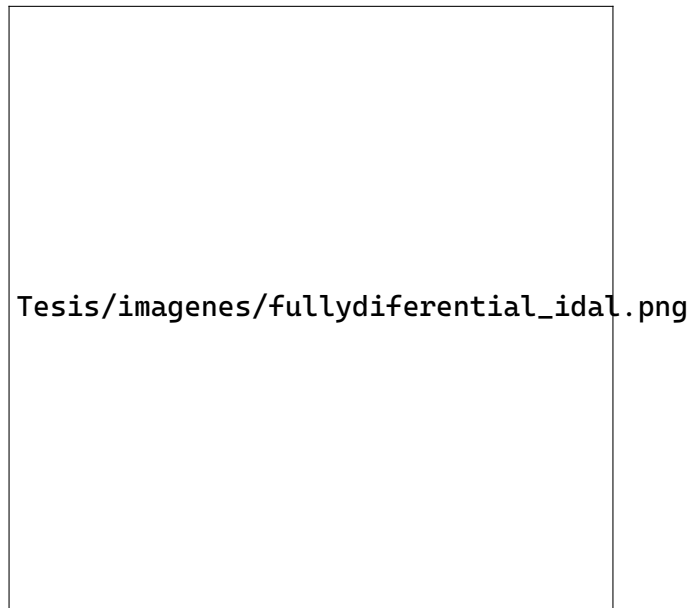


Figura 5.3: Circuito interno del modelo ideal del amplificador completamente diferencial.

Ganancias equivalentes, cálculo de resistencias y capacitancias.

El circuito elaborado en MATLAB simulink cuenta con ganancias de estabilidad las cuales son sustituidas por resistencias y capacitancias, ya que las ganancias más significativas son las que acompañan al integrador, el estudiante se centró en el cálculo de estas, por lo tanto, se parte de la función de voltaje del integrador:

$$V_{out} = -\frac{1}{RC} \int V_{in} dt$$

Cuya función de transferencia en el dominio S es equivalente a:

$$H(S) = \frac{V_{out}(S)}{V_{in}(S)} = -\frac{1}{sRC}$$

se sustituye $s = j\omega$

$$H(j\omega) = \frac{j}{j\omega \cdot RC}$$

Por lo tanto, el valor de la ganancia es igual a la magnitud de la función de transferencia:

$$|H(j\omega)| = K = \frac{1}{\omega RC} = \frac{1}{2\pi f_s RC}$$

En el modelo desarrollado en simulink de la Figura. 4.7 se observa que no se incluye el valor de 2π por lo que, este viene implícito en el valor de la ganancia $g = 2\pi * K$. Por lo tanto, la ecuación para el cálculo de RC queda de la siguiente manera:

$$K = \frac{1}{f_s * RC * 2\pi} = g = \frac{1}{f_s * RC}$$

Se propone un valor estándar de resistencias igual a:

$$R = 5k\Omega$$

En realidad, el valor de la ganancia ya fue calculado en el diagrama de bloques en simulink, el cual es $g = 0.75$. Por lo tanto, el cálculo de las capacitancias, se desarrolla en la siguiente ecuación:

$$C = \frac{1}{f_s \cdot R \cdot g} = \frac{1}{480kHz \cdot 5k\Omega \cdot 0.75}$$

Por lo que:

$$C = 555.555pF$$

ADC tipo flash.

El siguiente bloque en el modulador es el convertidor de datos analógico-digital, pieza esencial en el sistema, ya que aquí es donde sucede el proceso de cuantificación y muestreo (de donde se deriva, también, el error de cuantificación).

El estudiante diseñó un ADC tipo flash de 3 bits, se elige esta arquitectura en específico, ya que es una de las más veloces (siempre y cuando trabaje con un número de bits pequeños <8) debido a que convierte directamente la entrada analógica en una salida digital sin pasos intermedios. Se colocan 2^n resistores en serie, los cuales dividen el rango dinámico del voltaje en $2^n - 1$ partes iguales, el rango dinámico (o *full scale* FS) en el que trabaja este ADC va desde -900 mV hasta 900 mV. Por lo tanto, Δ que asigna la distancia entre los voltajes generados por los divisores de voltaje queda como sigue:

$$\Delta = \frac{FS}{2^n - 1} = \frac{900mV - (-900mV)}{2^3 - 1} = \frac{1.8}{7}$$

$$\Delta = 257.143mV$$

$$\frac{\Delta}{2} = 128.571mV$$

El primer valor y el último valor están a $\Delta/2$ de distancia del valor mínimo V_{min} y el valor máximo V_{max} respectivamente, esto es debido a la distribución del ruido de cuantificación, pero en la práctica, para lograr estos valores y el espacio Δ posterior entre cada uno de ellos, se logra colocando el mismo valor de resistencia en cada corte, excepto al principio y al final donde se coloca $R/2$. A continuación se muestra una gráfica que muestra la relación entre la entrada analógica y la salida digital, donde se puede apreciar que el ruido de cuantificación está adecuadamente distribuido.

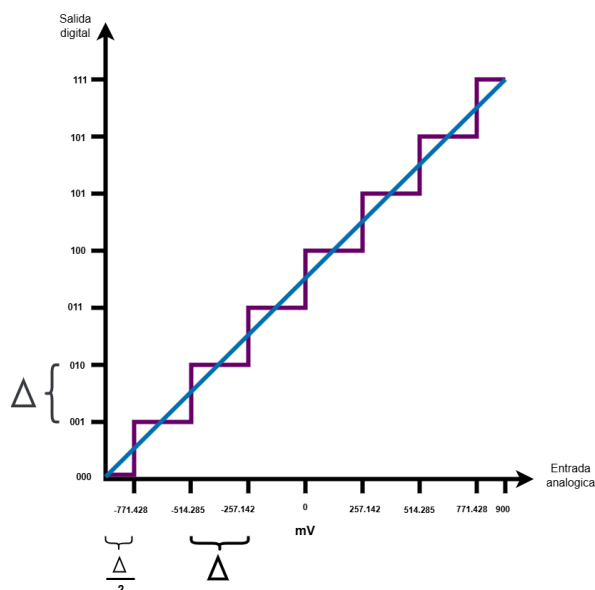


Figura 5.4: Relación entrada-salida del ADC flash propuesto.

Por lo tanto, el valor de voltaje que se obtiene de cada divisor, de menor a mayor sería: -771.428mV , -514.285mV , -257.142mV , 0V , 257.142mV , 514.285mV y 771.428mV . Estos valores son introducidos a un comparador, el cual también fue elaborado por el estudiante y su diseño se desglosará más adelante. El comparador (el cual también es *fully differential así como el flash*) compara la señal de entrada con los valores de referencia, asignando un valor, ya sea uno o cero, dependiendo de si el voltaje de referencia sobrepasa o no, los distintos valores de referencia, este tipo de salida es llamada termométrico y se ilustra en la siguiente tabla:

Voltaje V_{in}	Salida en los comparadores
$V_{in} > 771.428\text{mV}$	1 1 1 1 1 1 1
$514.285\text{mV} < V_{in} < 771.428\text{mV}$	0 1 1 1 1 1 1
$257.142\text{mV} < V_{in} < 514.285\text{mV}$	0 0 1 1 1 1 1
$0 < V_{in} < 257.142\text{mV}$	0 0 0 1 1 1 1
$-257.142\text{mV} < V_{in} < 0$	0 0 0 0 1 1 1
$-514.285\text{mV} < V_{in} < -257.142\text{mV}$	0 0 0 0 0 1 1
$-771.428\text{mV} < V_{in} < -514.285\text{mV}$	0 0 0 0 0 0 1
$V_{in} < -771.428\text{mV}$	0 0 0 0 0 0 0

Tabla 5.1: Relación de rangos de voltajes, asignados a la salida termométrica que genera el convertidor flash.

Normalmente, este diseño es conectado a un codificador, el cual transforma el código termométrico a binario o decimal, según sea solicitado. En los primeros prototipos elaborados por el estudiante, el codificador era colocado a la salida de los comparadores, arrojando una salida decimal, pero en la implementación del proyecto final, se decidió colocar este en otro bloque, quedando el ADC tipo flash como se muestra en la imagen (Figura. 5.5).

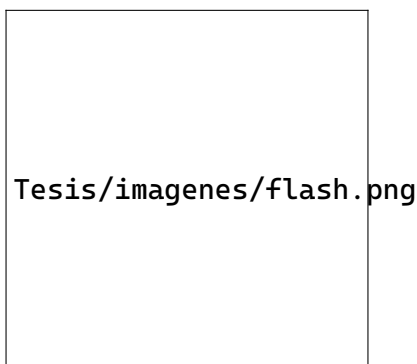


Figura 5.5: Modelo interno del ADC diferencial flash.

Diseño del comparador diferencial.

El comparador diferencial fue modelado utilizando Verilog-A, un lenguaje de descripción especializado en circuitos analógicos. A diferencia de Verilog, que está orientado al diseño de sistemas digitales, Verilog-A permite representar el comportamiento de componentes analógicos, esta característica lo hace ideal para describir elementos que interactúan directamente con señales analógicas, permitiendo su integración en simulaciones sin que sean tratados como bloques de electrónica mixta, como ocurriría si se utilizaran modelos en Verilog convencional. El uso de lenguajes como Verilog-A permite modelar las celdas a partir de su comportamiento deseado, especificado directamente en código. Esto evita la necesidad de implementar el diseño electrónico completo del bloque, lo cual resulta ventajoso cuando se quiere reducir la complejidad del análisis, especialmente en presencia de efectos no ideales como capacitancias o resistencias parásitas. El código elaborado por el estudiante puede encontrarse el capítulo 6/Apéndices de este documento.

Para la elaboración de este bloque, el estudiante en una primera estancia, se familiarizó con el lenguaje de descripción de hardware Verilog-A, esto a partir del manual existente en la web, "Affirma" de donde fueron utilizados los ejemplos existentes en este, como bases de apoyo para desarrollar los códigos presentados en esta tesis [25].

El código comienza declarando las variables que se utilizarán como entradas y salidas en el dispositivo, siendo estas, dos entradas de voltaje (positivo y negativo) y dos salidas de voltaje, ya que el amplificador operacional es *fully differential*, todo el circuito es trasladado a esta configuración. También se declaran las entradas para el voltaje de referencia, una para el voltaje positivo y otra para el voltaje negativo, respectivamente, más una entrada que funge como la entrada para el reloj del sistema, la cual es imprescindible para el adecuado funcionamiento del bloque. Todas estas variables son declaradas como tipo eléctricas, esto para que el sistema tenga en claro que debe recibir o entregar un voltaje en estas. Lo siguiente en declararse son los parámetros configurados como "reales" estos parámetros, tienen un valor asignado en el código, que responden a: el valor de voltaje, de umbral, de retardo en el tiempo, velocidad de subida y bajada y los valores de voltaje que se asignan a las salidas. Una característica de las variables "parámetros reales" es que, aunque ya cuentan con un valor asignado, este puede ser modificado desde el entorno de diseño de celdas, es decir cuando ya

se esté utilizando el bloque generado, únicamente entrando a “editar objeto” con la letra “Q” como cualquier otro elemento de la librería, esto le da versatilidad a la celda, para adaptarse a diferentes especificaciones.

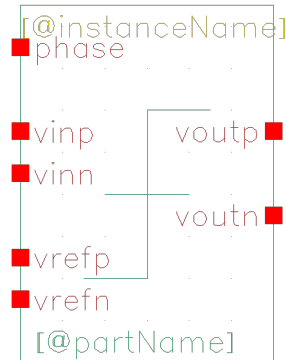


Figura 5.6: Comparador *fully differential* modelado a partir del lenguaje de descripción de hardware Verilog-A.

Se inicializa el bloque analógico, estableciendo que su evaluación se realizará cada vez que la señal de reloj atraviese el voltaje de umbral V_{th} . Esto es especialmente relevante, ya que, al tratarse de una señal periódica, el cruce por el umbral ocurre cada t_s dado que el ciclo del reloj corresponde al tiempo de muestreo. Es importante enfatizar que todo el circuito está sincronizado con la misma señal de reloj, por lo que el sistema en su conjunto puede considerarse síncrono, ya que el flujo general de datos o decisiones ocurre en sincronía con el reloj. Después de esto, comienza la descripción del comportamiento del comparador, si el voltaje recibido en V_{inp} es mayor que él V_{refp} la salida será igual a v_{dda} , la cual fue configura desde el circuito a 1V, por otra parte, si V_{inp} no es mayor a V_{refp} a V_{outp} se le asignará el voltaje v_{ssa} , en este caso 0v.

La siguiente sección del código, describe el mismo comportamiento, pero para la entrada de voltaje negativa, si V_{inn} es mayor a V_{refn} se le asigna a la salida 1V, si no, 0V. Es importante mantener estos valores en 1 y 0, ya que más tarde son leídos en el codificador, por funciones *case* anidadas, por lo que tener diferentes valores, podría generar errores inesperados.

Finalmente, se tiene la etapa de asignar los valores de salida a las variables de voltaje V_{outp} y V_{outn} esto con la función *transition* a la cual se le debe asignar el tiempo de subida y bajada para el cambio de voltaje, el cual ya se estableció previamente, en los parámetros reales. Se genera una nueva celda a partir del código, con el comportamiento deseado, como se puede observar en la Figura. 5.6. Esta celda es la que se emplea en el ADC tipo flash.

Decodificador termométrico-decimal.

Otro componente clave del modulador, también desarrollado por el estudiante, es el bloque codificador termométrico-decimal. En el circuito final, este bloque se encuentra encapsulado justo después del ADC flash. Su código, al igual que el del bloque anterior, se incluye en la sección de apéndices. La función principal del codificador es traducir el código termométrico

generado por la salida del ADC flash a su correspondiente valor decimal, es decir, realizar el proceso inverso al descrito previamente en la Tabla 5.1.

Se empiezan declarando las entradas, la cuales van desde t_1 hasta t_7 esta cantidad de entradas, es porque el codificador lee las 7 salidas de los comparadores. Se declara una única salida decimal d_1 y una entrada de reloj clk . Se sigue un proceso similar al anterior al declarar variables eléctricas y parámetros reales y se inicializa el bloque analógico. El cual también es evaluado cada t_s segundos.

El código se basa en funciones *case* anidadas una dentro de la otra, empieza evaluando del dígito más significativo, al menor, y si este es cero, evalúa el valor del siguiente, esto hasta llegar al 4to dígito, a partir de este, en lugar de evaluar el 5to dígito, evalúa directamente el 7mo, el código está configurado de esta manera para obtener simetría en la señal de salida, ya que de lo contrario, los valores negativos de la señal de salida, tendrían un comportamiento asimétrico a los valores positivos, para mejor entendimiento de este error obsérvese la Figura. 5.7. Donde se puede observar que la distribución de la señal de salida, aunque ligeramente, es asimétrica.



Figura 5.7: Salida cuantificada del ADC flash con distribución de niveles asimétricos.

Es por esto que el decodificador evalúa inversamente los valores negativos. Si él V_{in} es $>V_{ref}$ continúa evaluando, si no, asigna un valor de voltaje. Esta característica sucede por tratarse de valores negativos, ya que para que sea simétrica a la entrada positiva, se debe evaluar si V_{in} “más negativa” que V_{ref} y no “más positiva”.

Este error también pudo ser corregido, desde el comparador, pero el estudiante opto por ajustarlo desde el codificador. Esta configuración también es posible gracias a que las entradas

tienen configuración de código termométrico, no se activaran las entradas a menos que ya estén activas las anteriores, de ser una entrada binaria esto no sería posible, ya que tiene combinaciones más variadas que las que ofrece un código termométrico. Es así como se consigue la simetría deseada en la salida de la figura 5.8.



Figura 5.8: Salida cuantificada del ADC flash con distribución de niveles simétricos.

Es importante recordar que el ADC flash cuenta con una configuración diferencial, por lo tanto, es necesario un decodificador para la entrada positiva y otro para la entrada negativa, aclarar que se utilizó el mismo codificador para ambos casos, simplemente invirtiendo el orden de las entradas como se muestra en la Figura. 5.9. Este circuito genera la señal de salida del modulador $\Delta\Sigma$, la cual es retroalimentada a los integradores para realizar el moldeo de ruido.

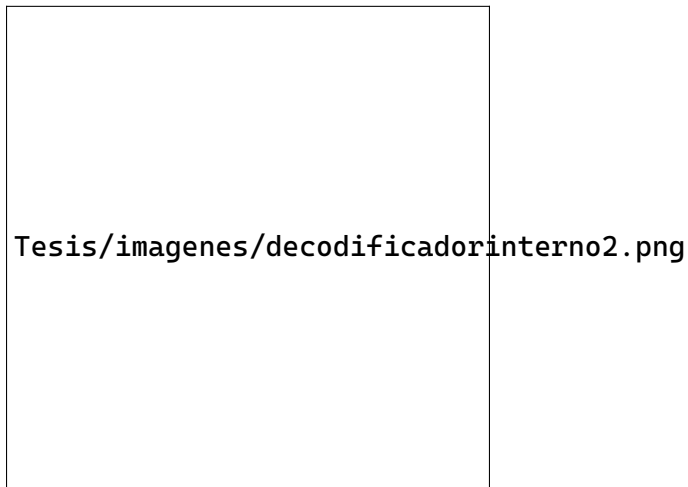


Figura 5.9: Decodificadores generados en Verilog-A, encargados de convertir la salida termométrica del ADC flash en salida decimal.

Señal diferencial de excitación.

Por último, se muestra la configuración en las fuentes que permite lograr señales de entrada diferenciales, esto fue logrado conectado la fuente que genera la entrada v_{ip} a la entrada negativa de una fuente de voltaje controlada por voltaje con una ganancia unitaria, esta configuración logra invertir el signo de la señal, pero mantiene la frecuencia, el valor de magnitud y valor en DC.

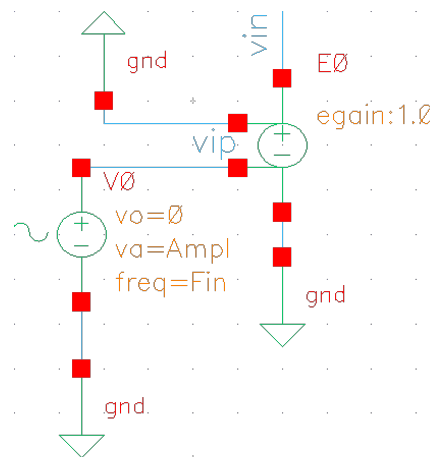


Figura 5.10: Configuración de señales diferenciales aplicadas en la entrada.

5.2. Análisis transitorio.

Al circuito se le aplica un análisis transitorio, esto quiere decir que se analiza como varían las señales a través del tiempo, lo cual ayuda a detectar inestabilidades, este tipo de análisis es el más cercano a lo que se observaría al conectar el circuito físico a un osciloscopio. Para realizar este análisis, se asignaron los mismos valores que al circuito elaborado en MATLAB, esto con la finalidad de comparar los resultados, por lo que $f_{in} = 3.193$ kHz, Amp = 506 mV y $t_s = 2.083$ μ s, el análisis, analiza el cambio en la señal durante 34.35 ms como se observa en la figura 5.11.



5.11

Figura 5.11: Valor asignado a las variables de: frecuencia de entrada, ganancia interna de los amplificadores, amplitud de la señal y tiempo de muestreo (inverso de la frecuencia de muestreo) para un análisis transitorio del modulador DS.

Este análisis genera la gráfica de la figura 5.12. En la cual se puede observar las señales diferenciales de entrada con una amplitud de 506.1 mV. Además, se observan las salidas del modulador, esta es una salida cuantificada con un pico de aproximada 700 mV, donde cada nivel cuenta con modulación por densidad de pulso, esta indica cuanto varía la señal en cada escalón, es como hacer un zoom, a al rango de señal V_{in} que es cuantificada a ese nivel, y observar que tanto varía. Esto es parecido a la modulación que existe cuando el cuantificador es de un solo bit, solo que cuenta con la diferencia que sucede en cada escalón del cuantificador.

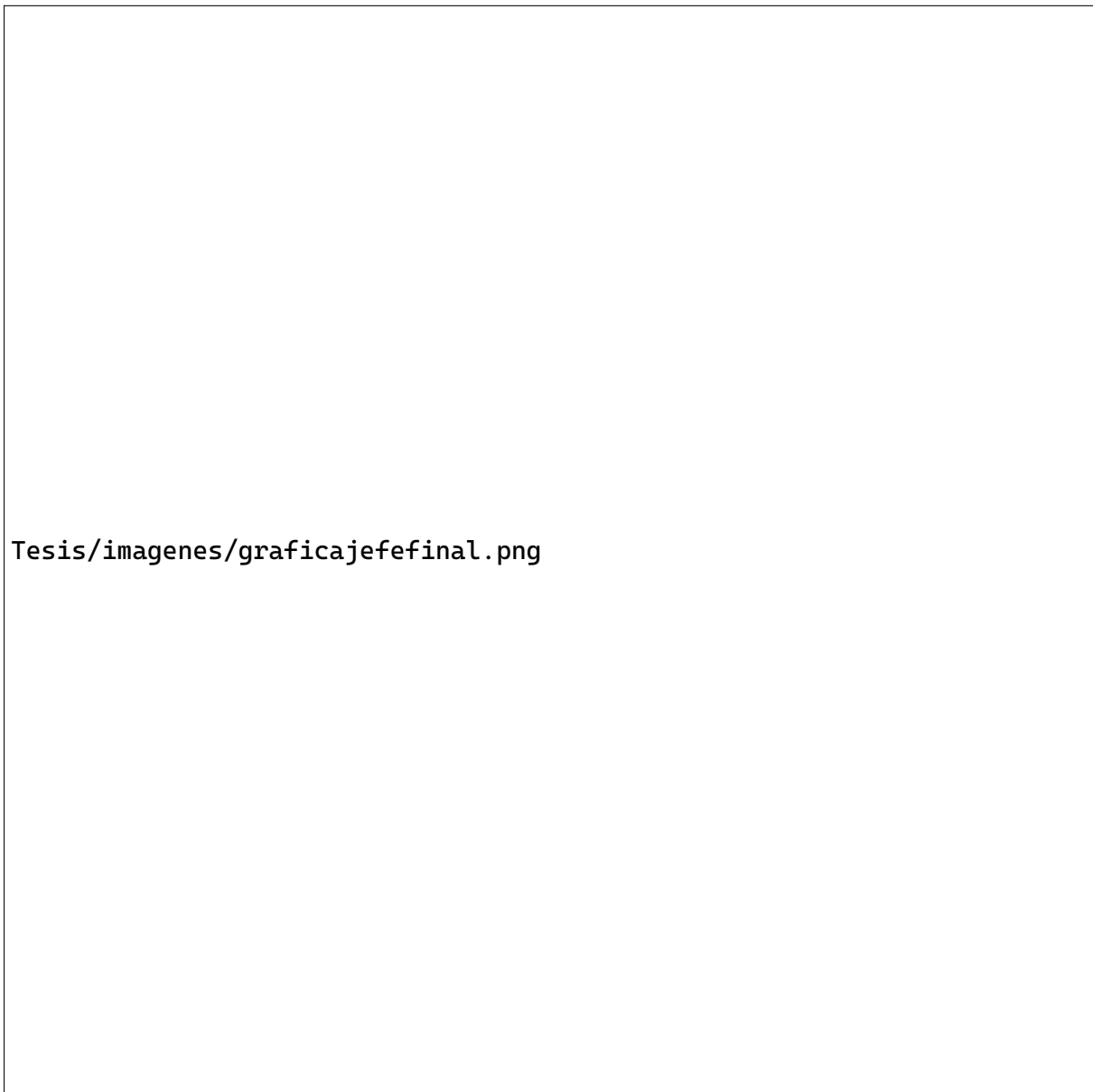


Figura 5.12: Análisis transitorio de la salida generada por el modular delta-sigma simulado en CADENCE-virtuoso.

5.3. Análisis en frecuencia.

Una vez realizado el análisis transitorio en Cadence, los datos deben ser analizados en el espectro de frecuencia, ya que es este el único que puede asegurar si el ruido es moldeado adecuadamente hacia frecuencias altas. Pero ahora que el circuito no está siendo simulado en el entorno MATLAB los datos deben ser trasladados. Por lo tanto, se crea un bloque el cual guarda los valores de salida cada ciclo de reloj en un archivo de texto, cuya ubicación

puede ser especificada desde las configuraciones del circuito y es enviado a una dirección de portafolio específico existente dentro del servidor de Cadence. Pero el estudiante aún no cuenta con acceso directo a este archivo de texto, ya que primero debe ser extraído de la carpeta del servidor. Para esto, se ocupa la aplicación *FileZilla* la cual está diseñada para facilitar la transferencia de archivos entre una computadora y servidores remotos (el caso de Cadence). Así, mediante esta aplicación el archivo .txt es enviado a la misma carpeta donde se encuentra el archivo de MATLAB.

Al igual que todos los códigos mencionados en esta tesis, este se puede encontrar en el apartado de apéndice. La diferencia principal de este código, es la lectura de datos de salida, ya que ahora no provienen de simulink, sino de Cadence, la única característica que este debe de cumplir, es estar ubicado en la misma carpeta de archivos que el código script de MATLAB, ya que la extracción de datos en vector, lo hace automáticamente.

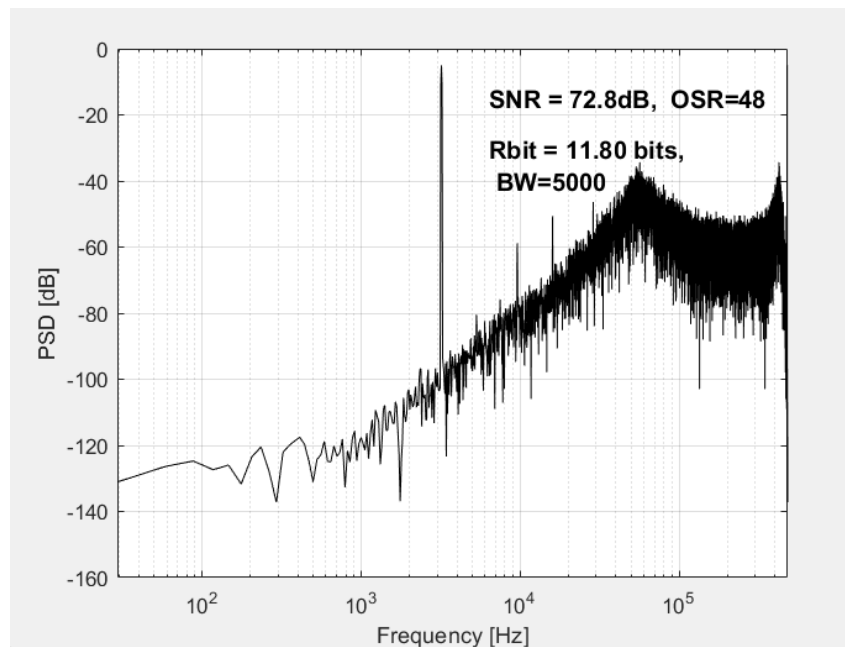


Figura 5.13: Análisis en frecuencia de la señal de salida generada por el modulador simulado en CADENCE virtuoso.

Una vez que se ejecuta el script, se genera la gráfica observada en la Figura. 5.13. La cual cuenta con un pico de potencia en aproximadamente 3 kHz el cual responde a la señal de entrada, se obtiene un adecuado moldeado de ruido para un modulador de segundo orden, donde la potencia del ruido en la banda útil es muy baja, mientras que en frecuencias altas, esta aumenta. Responde a una SNR alta de 72.8 dB, lo cual indica que el ruido no genera interferencias ni pérdida de información en la señal. Un número efectivo de bits de 11.8 bits, el cual sigue cumpliendo con el estándar deseado de 10 bits, teniendo más del triple de número efectivo de bits, que de bits reales. Con el ancho de banda establecido para la aplicación de 5 kHz.

Comparación de resultados con modelo ideal.

A continuación se superponen las gráficas diseñadas en un entorno ideal conforme al comportamiento matemático, con la gráfica generada por la simulación eléctrica. En primera instancia, se observa que el circuito eléctrico, no cuenta con la misma disminución de potencia en el ruido, esto puede ser debido a que los elementos modelados en Cadence los bloques utilizados en el diseño, son modelos ideales de los elementos, por lo tanto, esto también disminuye el rendimiento y precisión de los mismos. Por lo tanto, es lógico que también disminuya el valor de SNR y el número efectivo de bits.

Enfocándose en lo positivo, ambas señales cuentan con la pendiente asociada a un modulador de segundo orden, por lo tanto, el funcionamiento es óptimo, además a pesar de contar con más potencia en el ruido en general, la señal se acerca bastante a su modelo ideal. Por lo que se concluye es circuito como un circuito exitoso, que cumplirá su propósito de convertir señales, con menos consumo de potencia y menos área de silicio, a velocidades altas y sin distorsión o pérdida de información.



Figura 5.14: Análisis en frecuencia de la señal generada en el simulador electrónico CADENCE superpuesta con la señal generada en MATLAB-simulink.

Se elabora una tabla comparativa resaltando los datos más importantes de cada señal, se puede observar que comparten los mismos para metros de entrada, y lo único que varía es la

SNR y el ENOB, esto sucede a que en la simulación electrónica los bloques utilizados en el diseño, son modelos ideales de los elementos, por lo tanto, esto disminuye el rendimiento y precisión de los mismos.

Parámetros	circuito elaborado en Cadence	diagrama de bloques MATLAB
Fs	480 kHz	480 kHz
Fin	3.193359375 kHz	3.193359375 kHz
nbits	3	3
OSR	48	48
BW	5 kHz	5 kHz
SNR	72.8dB	85.3dB
ENOB	11.80 bits	13.88 bits

Tabla 5.2: Tabla comparativa, señal ideal generada en MATLAB vs señal generada en el simulador electrónico CADENCE.

En cuanto a la contribución de esta tesis al estado del arte, se incluye la Tabla 5.3 que compara el modulador diseñado, con moduladores previamente reportados, donde se evidencian las mejoras alcanzadas en términos de resolución y SNR. Estos resultados posicionan al diseño propuesto como una solución técnicamente competitiva para aplicaciones de instrumentación médica.

Aplicación específica	Adquisición de señales para un marca-pasos [16]	ADC para aplicaciones biomédicas[17]	Sistemas de adquisición datos biomédicos [18]	Modulador Delta-Sigma para aplicaciones biomédicas
resolución	8 bits	8 bits	8 bits	11.80 bits
tipo de convertidor	Modulador DS de tercer orden	Time-based ADC	modulador DS de primer orden	modulador DS de segundo orden
Fs	8 kHz	45 kHz	512 kHz	480 kHz
OSR	16	convertidor tasa Nyquist	100	48
SNR	47 dB	47 dB	100 dB	72.8 dB
BW	250 Hz	20 kHz	500 Hz	5 kHz

Tabla 5.3: Resolución, tipo de convertidor, Fs, OSR, SNR, BW encontrados en trabajos de conversión de datos y aplicaciones biomédicas previos comparados con el diseñado en el documento.

5.4. Desempeño del modulador aplicando una señal ECG

Respuesta temporal

Una vez que se ha comprobado la calidad y eficacia del modulador, se aplica una señal ECG a la entrada del mismo, con el objetivo de observar su desempeño ante una señal biomédica. La señal aplicada fue recuperada de un dispositivo de medición, el cual realizó la medición de una señal ECG real, para luego ser documentada y almacenada en un archivo de texto. El cual es leído y ejecutado cada ciclo de reloj por un módulo realizado en verilog-A. La forma de onda típica que se encuentra en una señal ECG se puede observar en la Figura. 5.15 la cual reporta

el potencial eléctrico, presente en las cámaras del corazón encargadas de bombear sangre a los pulmones y distribuirla por todo el cuerpo.



Figura 5.15: Imagen ilustrativa de una típica señal ECG, obtenida de [26].

Esta señal es reconstruida cada ciclo de reloj, a partir de los datos almacenados en el archivo de texto, y fue aplicada a la entrada del modulador, para observar y analizar la respuesta temporal de este (Figura. 5.16). Para mejorar el desempeño del modulador ante una señal ECG y minimizar el derrame espectral, se ajustó la frecuencia de muestreo a 350 kHz y se redujo el número de periodos de análisis a 64. Obteniéndose así la Figura. 5.16, donde se puede apreciar la señal de entrada ECG (rosa), su correspondiente señal diferencial (verde menta), la señal de salida positiva (roja) y su contraparte diferencial (verde). La señal cuenta con los típicos picos R y T característicos de una señal ECG, los cuales, como se observa en la imagen, son cuantificados y modulados por el convertidor de datos, comprobando que el modulador cuenta con un buen desempeño, para señales biomédicas.

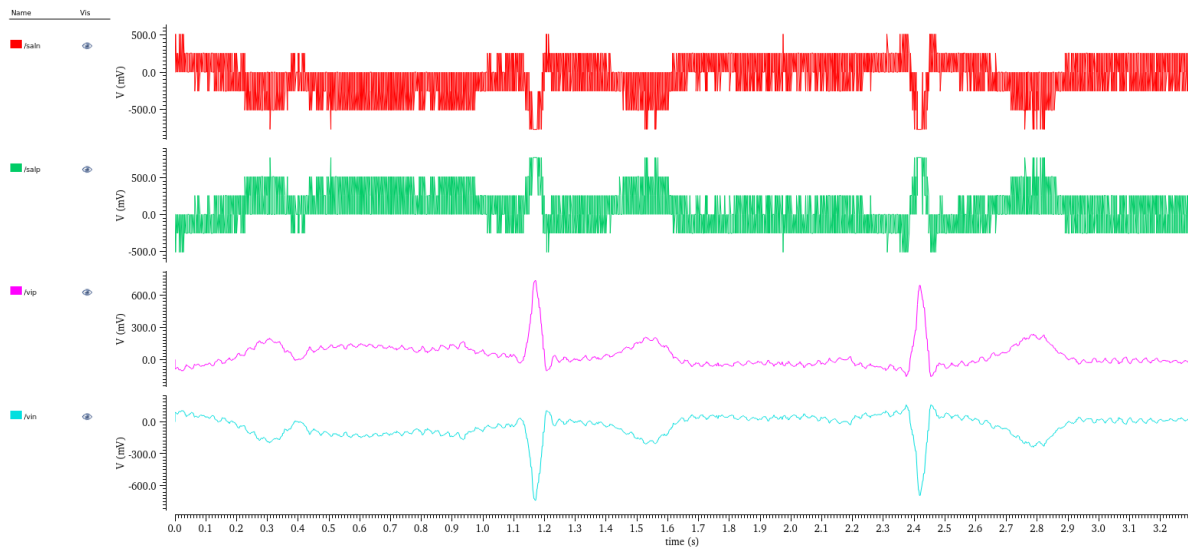


Figura 5.16: Análisis de respuesta temporal al aplicar una entrada ECG.

Respuesta en frecuencia

Una vez obtenidos los valores de salida del análisis temporal, estos son migrados a MATLAB, para ser analizados en el dominio de la frecuencia. Para la comparativa del adecuado comportamiento de la señal ECG en el dominio de la frecuencia, se recuperó del artículo [27], una imagen ilustrativa (Figura. 5.17) de como se comportan las señales ECG en el dominio espectral.



Figura 5.17: Desempeño típico de una señal ECG en el dominio de la frecuencia, imagen recuperada de [27].

Al comparar el resultado en frecuencia con la gráfica observada en [27] se observa que el comportamiento es el esperado para un barrido en frecuencia de una señal ECG. La cual presenta picos en bajas frecuencias, donde se pueden observar los componentes principales del ECG, como la onda P, el complejo QRS y la onda T. Donde la onda fundamental está cerca de 0.8 Hz, donde se encuentran los picos de potencia más altos, a aproximadamente -5 dB. Luego es posible encontrar subarmónicos, que identifican los segmentos ST y el complejo QRS. Podemos observar como la señal ECG permanece en frecuencias bajas, mientras que el ruido es moldeado a frecuencias altas.



Figura 5.18: Barrido en frecuencia de la señal obtenida al aplicar una entrada ECG al modulador $\Delta\Sigma$ diseñado en esta Tesis.

Por otro lado, en caso de que la señal recuperada presente anomalías características de enfermedades cardíacas, dichas irregularidades no serán atenuadas erróneamente por el modulador durante el proceso de filtrado. Esto se debe a una propiedad fundamental del modulador Delta-Sigma: la existencia de dos funciones de transferencia independientes, una para la señal y otra para el ruido. Gracias a esta característica, el modulador es capaz de distinguir entre la información útil contenida en la señal de entrada y el ruido cuantificación. Por lo tanto, incluso si la señal presenta desviaciones asociadas a una condición clínica, su contenido espectral permanecerá dentro de la banda útil y no será eliminado o distorsionado como si se tratará de ruido.

CAPÍTULO 6

Conclusiones y trabajos a futuro

6.1. Conclusiones.

El desarrollo de este proyecto permitió diseñar, modelar y analizar un modulador delta-sigma de segundo orden orientado a aplicaciones biomédicas, obteniendo un desempeño caracterizado por una resolución efectiva de 11.8 bits, un rango dinámico de 500 mV, un ancho de banda de 5 kHz y una SNR de 72.8 dB, cumpliendo con los requisitos establecidos al inicio del trabajo. En relación con el objetivo de aplicaciones biomédicas, se concluye que fue alcanzado satisfactoriamente. El modulador logró cuantificar y modular señales ECG preservando sus componentes frecuenciales, gracias a su capacidad de redistribuir el ruido fuera del ancho de banda de interés. Esto se explica por las funciones de transferencia independientes que caracterizan a los moduladores delta-sigma, las cuales permiten separar eficazmente la señal útil del ruido de cuantificación.

A lo largo del proceso, se consolidó un diseño eficiente en el rango de frecuencia típico de señales biomédicas (DC a 5 kHz), validado mediante simulaciones en MATLAB-Simulink y posteriormente en la plataforma Cadence-Virtuoso. La selección de un cuantificador multibit y de una arquitectura híbrida con retroalimentación y alimentación directa distribuidas (CIFBFF) se mostró adecuada, al equilibrar la complejidad del sistema con su estabilidad y calidad de salida. El análisis espectral evidenció un correcto moldeado de ruido, con concentración de energía en la banda útil y reducción significativa del ruido en la banda de interés. Por otro lado, la implementación en tiempo continuo permitió mejorar la eficiencia energética del sistema y simplificar el diseño al evitar etapas de muestreo adicionales.

El documento intenta explicar conceptos relacionados con procesamiento de señales y convertidores de datos, que en su mayoría suelen ser conceptos complejos, de una manera más digerible y amigable, con el objetivo de, que si este documento fuera el primer acercamiento de algún estudiante hacia los convertidores de datos, este no se sienta intimidado y sea un puente entre estos conceptos iniciales, y los libros especialistas, los cuales explican detallada y exhaustivamente el funcionamiento de un modulador delta-sigma. Como recomendaciones a la comunidad estudiantil, que deseen construir un convertidor de datos, con características similares, se recomienda: seguir la misma línea de trabajo, comenzar con ejercicios pequeños, de procesamiento de datos y construyendo convertidores pequeños, como el flash, ya que

esto hace el aprendizaje lineal y gradual, ayudando a que no sea tan complicado y con cada ejercicio o simulación reforzar los conocimientos teóricos con práctica en los simuladores. En las habilidades blandas, el ser consciente de la cantidad de horas de lectura necesarias, para comenzar a entender únicamente la superficie del tema, tener mucho entusiasmo por el tema, paciencia, perseverancia, disciplina. Y un equipo de apoyo tanto académico, como emocional, ya que este tipo de proyectos pueden llegar a ser muy exhaustivos y absorbentes.

6.2. Trabajo a futuro.

- Identificar anomalías cardíacas a partir de un banco espectral de señales ECG.
Como trabajo futuro, sería interesante desarrollar un sistema que, usando la señal ECG digitalizada por el modulador, pueda analizar el contenido en frecuencia en tiempo real. Dado que el diseño propuesto respeta fielmente las componentes de la señal original, y que a partir de un banco espectral se puedan identificar diferentes anomalías cardíacas. Aunque no necesariamente se pueda predecir un infarto, este tipo de análisis sí podría ayudar a advertir sobre su posible aparición, dando una señal de alerta temprana. Incluir esta funcionalidad en un dispositivo portátil o de monitoreo continuo podría ser de gran utilidad para la prevención y el seguimiento de enfermedades del corazón.
- Implementación de la etapa de acondicionamiento de señal.
En esta tesis se asumió una señal de entrada previamente filtrada y amplificada. Sin embargo, para una aplicación real, es necesario integrar un front-end analógico que incluya etapas de amplificación y filtrado pasa bajas. El diseño conjunto de estas etapas con el modulador podría mejorar la relación señal a ruido global del sistema.
- Diseño de algoritmos de decimación y filtrado digital
Después del proceso de modulación, es necesario realizar una decimación efectiva que preserve la banda útil y elimine el ruido moldeado. Desarrollar algoritmos de filtrado digital y la etapa de sub-muestreo, puede ampliar la aplicabilidad del sistema en dispositivos portátiles con bajo poder de procesamiento.

Apéndices

A. Script de MATLAB modulador de 1 bit.

Código 1: código script de MATLAB relacionada al primer prototipo del modulador con un solo bit en el cuantificador.

```
1 clear all;
2 clc;
3 format long;
4 nbit=1;
5 R=40;
6 Fs=5e6;
7 bw=Fs/(2*R);
8
9 Ts=1/Fs;
10 N=2^14;
11 nper=17;
12
13
14 Fin=nper*Fs/N;
15 finrad=Fin*2*pi;
16 Amp_dB=-3;
17 Ampl=10^(Amp_dB/20)*0.9;
18 Ntransient=0;           % Number of transient points for the initial
19
20
21 % Open Simulink diagram
22 options=simset('RelTol', 1e-6, 'MaxStep', 1/Fs); %Tolerance parameters
23                                           %for convergence Help
24 simout=sim('ds_comparador.slx', (N+Ntransient)/Fs, options);
25
26 %Calculo SNR
27 w=hann(N);           % Sets a Hanning window
28 f=Fin/Fs;           % Normalized signal frequency
29 fB=N*(bw/Fs);       % Base-band frequency bins (number of FFT points)
30 yy1=zeros(1,N);
31 yy1=simout.yout(1+Ntransient:1+N+Ntransient)';
32
33 ptot=zeros(1,N);    % Defines vector por locating the FFT transform
```

```
34 [snr,ptot]=calcSNR(yy1(1:N),f,fB,w',N); % Uses calcSNR function, must be
35                                     % in the same folder file
36 Rbit=(snr-1.76)/6.02; % Equivalent resolution in bits from data
37                               % converters theory
38
39
40 %-----%
41 %           Display result plots           %
42 %-----%
43 ptot=ptot-max(ptot)+Amp_dB;
44 figure(1);
45 semilogx(linspace(0,Fs,N), ptot(1:N), 'b');
46 grid on;
47 xlabel('Frequency [Hz]')
48 ylabel('PSD [dB]')
49 axis([0 Fs -160 0]);
50 text_handle = text(2*Fin,-120, sprintf('\bSNR = %4.1fdB, OSR=%d\n',...
51     snr,R), 'FontSize',12, 'FontWeight', 'bold');
52 text_handle = text(2*Fin,-130, sprintf('\bRbit = %2.2f bits,\n BW=%d\n'
53     ,...
54     Rbit,bw), 'FontSize',12, 'FontWeight', 'bold');
```

B. Script de MATLAB modulador de 3 bits.

```

1  clear all;
2  clc;
3  format long;
4  nbit=3;
5  R=48;
6  bw=5e3;
7  Fs=bw*R*2;
8  %nper+1/Fin=Calculo tiempo simulacion
9  Ts=1/Fs;
10 N=2^14;
11 nper=109;
12 Fin=nper*Fs/N;
13 finrad=Fin*2*pi;
14 Amp_dB=-5;
15 Ampl=10^(Amp_dB/20)*0.9;
16 Ntransient=0;
17
18
19 out=textread('ds_muestras_3bits.txt');
20 %vout=out-mean(out);
21 vout=out;
22
23 %calculo FFT
24 yy=fft(vout(1:(N)));
25 freq=(0:(Fs/N):(N-1)*(Fs/N));
26 w=hann(N);
27 f=Fin/Fs; % Normalized signal frequency
28 fB=N*(bw/Fs); % Base-band frequency bins (number of FFT points)
29 yy1=zeros(1,N);
30 yy1=vout(1+Ntransient:1+N+Ntransient);
31 %%
32
33 ptot=zeros(1,N); % Defines vector por locating the FFT transform
34 [snr,ptot]=calcSNR(yy1(1:N),f,fB,w',N); % Uses calcSNR function, must be
35 % in the same folder file
36 Rbit=(snr-1.76)/6.02; % Equivalent resolution in bits from data
37 % converters theory
38
39
40 %-----%
41 % Display result plots %
42 %-----%
43 ptot=ptot-max(ptot)+Amp_dB;
44 hold on;

```

```
45 figure(1);
46 %clf;
47 semilogx(linspace(0,Fs,N), ptot(1:N), 'k');
48 grid on;
49 xlabel('Frequency [Hz]')
50 ylabel('PSD [dB]')
51 axis([0 Fs -160 0]);
52 text_handle = text(2*Fin,-20, sprintf('\bSNR = %4.1fdB,  OSR=%d\n',...
53     snr,R), 'FontSize',12, 'FontWeight', 'bold');
54 text_handle = text(2*Fin,-30, sprintf('\bRbit = %2.2f bits,\n BW=%d\n',...
55     Rbit,bw), 'FontSize',12, 'FontWeight', 'bold');
56 hold off;
```

C. Código comparador fully differential

Código 2: Código Verilog-A del integrador

```
1 'include "constants.vams"
2 'include "disciplines.vams"
3 module comparatoreID (vinp, vinn, voutp, voutn, vrefp, vrefn, phase);
4 input vinp, vinn, vrefp, vrefn, phase;
5 output voutp, voutn;
6 electrical vinp, vinn, voutp, voutn, vrefp, vrefn, phase;
7 parameter real vth = 0.6;
8 parameter real ritardo = 0.0p;
9 parameter real trise = 50p;
10 parameter real tfall = 50p;
11 parameter real vdda = 1.2;
12 parameter real vssa = 0.0;
13 real outp, outn;
14 analog begin
15 @ ( cross (V(phase) - vth, 1 ) )
16 begin
17 if (V(vinp) > V(vrefp))
18 begin
19 outp = vdda;
20 end
21 else outp = vssa;
22 if (V(vinn) > V(vrefn))
23 begin
24 outn = vdda;
25 end
26 else outn = vssa;
27 end
28 V(voutp) <+ transition (outp, ritardo, trise, tfall);
29 V(voutn) <+ transition (outn, ritardo, trise, tfall);
30 end
31 endmodule
```

D. Código decodificador

Código 3: Código Verilog-A del integrador

```
1 'include "constants.vams"
2 'include "disciplines.vams"
3 module ter_decimal_posi_y_neg_900m(t1,t2,t3,t4,t5,t6,t7,d1,clk);
4 input t1,t2,t3,t4,t5,t6,t7;
5 output d1;
6 inout clk;
7 electrical t1,t2,t3,t4,t5,t6,t7,d1,clk;
8 real dec1;
9 real last_dec1;
10 parameter vth=0;
11 parameter tdel=0.0, rt=1p;
12 parameter fallt=1p;
13 analog begin
14
15 @(cross((V(clk)-vth) , -1 )) begin
16     case (V(t1))
17     0:begin
18         case (V(t2))
19         0:begin
20             case (V(t3))
21             0:begin
22                 case(V(t4))
23                 0:begin
24                     case(V(t7))
25                     0: dec1=-771.43m;
26                     1: begin
27                         case(V(t6))
28                         0: dec1=-514.28m;
29                         1:begin
30                             case(V(t5))
31                             0: dec1=-257.14m;
32                             1: dec1 =0 ;
33                             endcase
34                         end
35                     endcase
36                 end
37             endcase
38         end
39         1: dec1=0;
40         endcase
41     end
42     1: dec1=257.14m;
```

```
43         endcase
44     end
45     1: dec1=514.28m;
46     endcase
47 end
48 1: dec1=771.43m;
49 endcase
50 end
51 //last_dec1 = dec1;
52 V(d1)<+ transition(dec1,tdel,rt,fallt);
53 end
54 endmodule
```

E. Código de matlab, relacionando los datos de cadence

```
1 clear all;
2 clc;
3 format long;
4 nbit=3;
5 R=48;
6 bw=5e3;
7 Fs=bw*R*2;
8 %nper+1/Fin=Calculo tiempo simulacion
9 Ts=1/Fs;
10 N=2^14;
11 nper=109;
12 Fin=nper*Fs/N;
13 finrad=Fin*2*pi;
14 Amp_dB=-5;
15 Ampl=10^(Amp_dB/20)*0.9;
16 Ntransient=0;
17
18
19 out=textread('ds_muestras_3bits.txt');
20 %vout=out-mean(out);
21 vout=out;
22
23 w=hann(N);
24 f=Fin/Fs; % Normalized signal frequency
25 fB=N*(bw/Fs); % Base-band frequency bins (number of FFT points)
26 yy1=zeros(1,N);
27 yy1=vout(1+Ntransient:1+N+Ntransient);
28 %%
29
30 ptot=zeros(1,N); % Defines vector por locating the FFT transform
31 [snr,ptot]=calcSNR(yy1(1:N),f,fB,w',N); % Uses calcSNR function, must be
32 % in the same folder file
33 Rbit=(snr-1.76)/6.02; % Equivalent resolution in bits from data
34 % converters theory
35
36
37 %-----%
38 % Display result plots %
39 %-----%
40 ptot=ptot-max(ptot)+Amp_dB;
41 hold on;
42 figure(1);
43 %clf;
44 semilogx(linspace(0,Fs,N), ptot(1:N), 'k');
```

```
45 grid on;
46 xlabel('Frequency [Hz]')
47 ylabel('PSD [dB]')
48 axis([0 Fs -160 0]);
49 text_handle = text(2*Fin,-20, sprintf('\bSNR = %4.1fdB,  OSR=%d\n',...
50     snr,R), 'FontSize',12, 'FontWeight', 'bold');
51 text_handle = text(2*Fin,-40, sprintf('\bRbit = %2.2f bits,\n BW=%d\n',...
52     Rbit,bw), 'FontSize',12, 'FontWeight', 'bold');
53 hold off;
```


Referencias

- [1] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals & systems*. Pearson Educación, 1997. → [p1]
- [2] G. Vargas López *et al.*, “Construcción de un equipo de adquisición de señales para usarse con sensores en aplicaciones biomédicas.” B.S. thesis, ESPOL. FIEC., 1997. → [p1]
- [3] P. A. H. Organization, “Leading causes of death and disease burden in the americas: Noncommunicable diseases and external causes.” *PAHO*, 2024, fecha de consulta: Noviembre 2024. [Online]. Available: <https://doi.org/10.37774/9789275128626> → [p2]
- [4] OMS región de las Américas, “Enfermedades cardiovasculares,” Fecha de consulta: noviembre 2024. [Online]. Available: <https://www.paho.org/es/temas/enfermedadescardiovasculares> → [p2], [p5]
- [5] O. panamericana de la Salud. (2019) Enlace. Portal de datos. [Online]. Available: <https://www.paho.org/es/enlace/causas-principales-mortalidad-discapacidad> → [p3]
- [6] Instituto Nacional de Estadística y Geografía (INEGI), “Estadísticas de Defunciones Registradas (EDR) 2022,” Oct 2023, Comunicado de prensa. [Online]. Available: <https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2023/EDR/EDR2022-Dft.pdf> → [p2]
- [7] Organización Mundial de la Salud, “Reforzar la salud digital puede ayudar a prevenir millones de muertes por enfermedades no transmisibles,” 2024, Fecha de consulta: noviembre 2024. [Online]. Available: <https://www.who.int/es/news/item/23-09-2024-boosting-digital-health-can-help-prevent-millions-of-deaths-from-noncommunicable-diseases> → [p5]
- [8] P. D. Ordunez P., Lombardi C., “Hearts in the americas: a global example of using clinically validated automated blood pressure devices in cardiovascular disease prevention and management in primary health care settings,” *Human Hypertension*, no. 37, pp. 126–129, 2023. [Online]. Available: <https://www.nature.com/articles/s41371-022-00659-z.pdf> → [p5]
- [9] M. D. SUÁREZ, “Diseño del layout y caracterización de un circuito integrado cmos para aplicaciones biomédicas: convertidor analógico digital sigma delta de doble rampa,” 2013. → [p6], [p31]

- [10] A. T. Reali, “El origen de los biopotenciales,” *MedULA: revista de la Facultad de Medicina*, vol. 5, no. 1, p. 3, 1996. → [p6]
- [11] R. F. Yazicioglu, C. Van Hoof, and R. Puers, *Biopotential readout circuits for portable acquisition systems*. Springer Science & Business Media, 2008. → [p6]
- [12] F. V. G. E. De la Fuente Cortes G, “Diseño de un adc sar de 14 bits para aplicaciones biomédicas robusto a variaciones de proceso y temperatura,” Tesis de Maestría, Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, México, 2014. [Online]. Available: <https://inaoe.repositorioinstitucional.mx/jspui/handle/1009/175> → [p6], [p7]
- [13] , “A low-power data interface circuit and analog data converter for biomedical device,” Ph.D. dissertation, , 2013. → [p7]
- [14] S.-Y. Lee, C.-P. Wang, and Y.-S. Chu, “Low-voltage ota-c filter with an area-and power-efficient ota for biosignal sensor applications,” *IEEE transactions on biomedical circuits and systems*, vol. 13, no. 1, pp. 56–67, 2018. → [p7]
- [15] J. Van Rethy, M. De Smedt, M. Verhelst, and G. Gielen, “Predictive sensing in analog-to-digital converters for biomedical applications,” in *International Symposium on Signals, Circuits and Systems ISSCS2013*. IEEE, 2013, pp. 1–4. → [p7]
- [16] A. Gerosa, A. Maniero, and A. Neviani, “A fully integrated two-channel a/d interface for the acquisition of cardiac signals in implantable pacemakers,” *IEEE Journal of Solid-State Circuits*, vol. 39, no. 7, pp. 1083–1093, 2004. → [p7], [p65]
- [17] H. Y. Yang and R. Sarpeshkar, “A bio-inspired ultra-energy-efficient analog-to-digital converter for biomedical applications,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 11, pp. 2349–2356, 2006. → [p7], [p8], [p65]
- [18] E. Aksenov, Y. M. Ljashenko, A. Plotnikov, D. Prilutskiy, S. Selishchev, and E. Vetvetskiy, “Biomedical data acquisition systems based on sigma-delta analogue-to-digital converters,” in *2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 4. IEEE, 2001, pp. 3336–3337. → [p7], [p65]
- [19] O. J. O. D. Río, “La informática en la investigación biomédica. análisis de señales bioeléctricas,” vol. 6, no. 1, pp. 89–91, Enero-Junio 2018. [Online]. Available: <https://revistas.udistrital.edu.co/index.php/tia/article/view/13270> → [p8], [p16]
- [20] F. Maloberti, *Data converters*. Springer, 2007. → [p9], [p14], [p16], [p17], [p18], [p20]
- [21] T. C. Carusone, D. Johns, and K. Martin, *Analog integrated circuit design*. John Wiley & Sons, 2011. → [p10]
- [22] M. J. Pelgrom, *Analog-to-digital conversion*. Springer, 2013. → [p10], [p13]

-
- [23] L. A. Esperanza Hernández, “Modulador delta-sigma para búsqueda de patrones en señales ecg.” → [p37]
- [24] F. Gerfers and M. Ortmanns, *Continuous-time sigma-delta A/D conversion: fundamentals, performance limits and robust implementations*. Springer Science & Business Media, 2006, vol. 21. → [p26], [p27], [p30], [p35], [p38], [p39], [p40]
- [25] Cadence Design Systems, *Affirma Verilog-A Language Reference Manual*, Cadence Design Systems, Inc., San Jose, California, 2000, version 1.0. → [p56]
- [26] G. De La Fuente-Cortes, A. Diaz-Mendez, G. E. Flores-Verdad, and V. R. Gonzalez-Diaz, “On-chip fuzzy logic synthesis of a new ischemic and non-ischemic heartbeat classifier,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 1, pp. 476–480, 2020. → [p66]
- [27] V. R. Gonzalez-Diaz, “On-chip active filter for electrocardiographic signals through cmos pseudo resistors,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2025. → [p67], [p68]
- [28] R. G. Lyons, *Understanding Digital Signal Processing*. Prentice Hall, 2010. → [p11], [p12], [p17], [p22], [p23]
- [29] B. Razavi, *Principles of data conversion*. IEEE, 1994. → [p13], [p21], [p22], [p25]
- [30] A. Blogs. (2025) Solo arduino. Última consulta: 5 de febrero de 2025. [Online]. Available: <https://soloarduino.blogspot.com/2015/09/> → [p13]
- [31] J. C. Candy and G. C. Temes, *Oversampling delta-sigma data converters: theory, design, and simulation*. John Wiley & Sons, 1991. → [p27]
- [32] R. Behzad, “Design of analog cmos integrated circuits,” 2017. → [p]
- [33] K. Ogata, *Ingeniería de control moderna*. Pearson educación, 2003. → [p30]
- [34] O. Belotti, E. Bonizzoni, and F. Maloberti, “Exact design of continuous-time sigma-delta modulators with multiple feedback dacs,” *Analog Integrated Circuits and Signal Processing*, vol. 73, pp. 255–264, 2012. → [p33], [p34]