

INSTITUTO TECNOLÓGICO DE BUENOS AIRES

22.05 - ANÁLISIS DE SEÑALES Y SISTEMAS DIGITALES

TRABAJO PRACTICO N3

CONVERSORES A/D Y D/A

Grupo 4:

Agustín Ignacio GALDEMAN
Leg. 59827

Juan Martín LAGUINGE
Leg. 57430

Victor Christian OH
Leg. 56679

João ROSA
Leg. 62370

Profesor:

Daniel Andres JACOBY

Carlos F. BELAUSTEGUI GOITIA

Entregado: 12 de junio de 2020

Contents

1	Conversor Clásico	2
1.1	Diseño	2
1.1.1	Entrada de la Señal	3
1.1.2	DAC	3
1.2	Señal de Clock	3
1.2.1	Conexión ADC	4
1.3	Mediciones	5
1.3.1	Frecuencia de muestreo	5
1.3.2	Error de cuantización	6
1.3.3	Tensiones de entrada constante	9
2	Conversor $\Sigma\Delta$	12
2.1	Marca Teórico	12
2.1.1	Error de cuantización	12
2.1.2	Oversampling	13
2.1.3	Modulación y demodulación $\Sigma\Delta$	14
2.1.4	Noise shaping	15
2.1.5	Filtro decimador	16
2.2	Simulación	17

Conversor Clásico

El objetivo de esta práctica fue la simulación mediante el programa Proteus del siguiente diagrama en bloques:

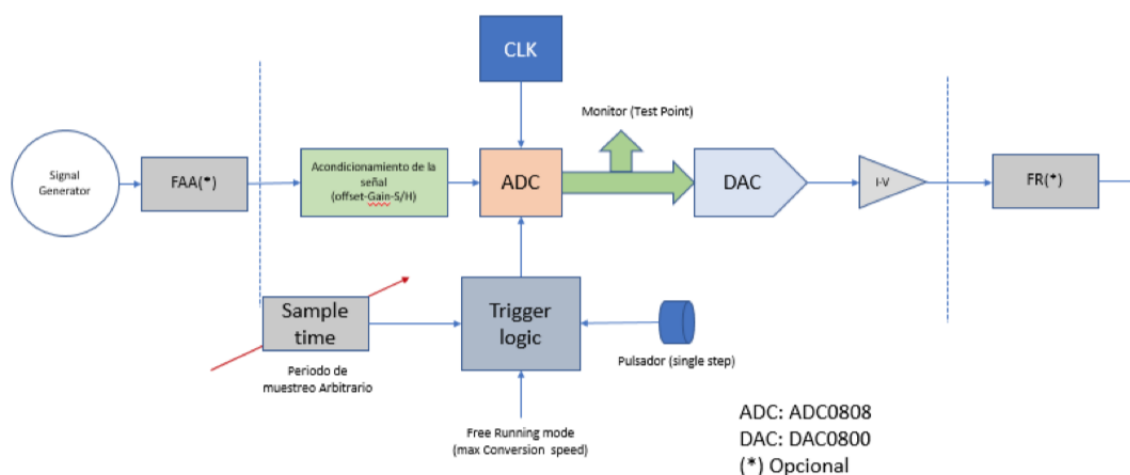


Figure 1.1: Diagrama en bloques de sistema simulado

A continuación se detallarán las elecciones y criterios de diseño y las mediciones relevante en el sistema Proteus.

1.1 DISEÑO

En la figura 1.3 se presenta la implementación completa en Proteus:

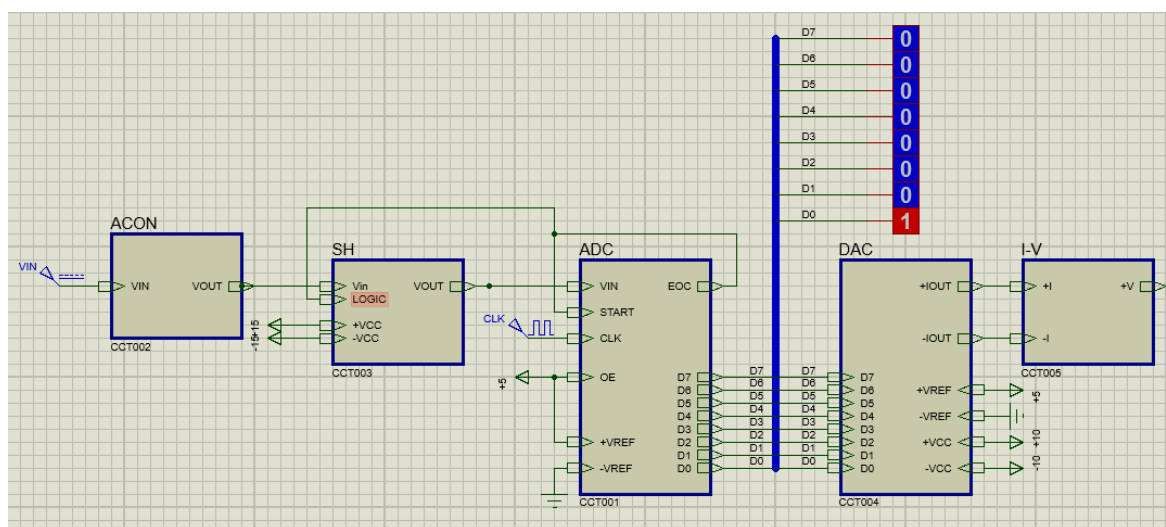


Figure 1.2: Diagrama en bloques de sistema simulado

1.1.1 ENTRADA DE LA SEÑAL

El circuito cuenta con una etapa de *offset* a la entrada, montando la señal inicial sobre una tensión continua de 2,5V para maximizar el rango dinámico de las etapas posteriores. Para lograr esto, se utilizaron dos amplificadores operacionales como sumadores.

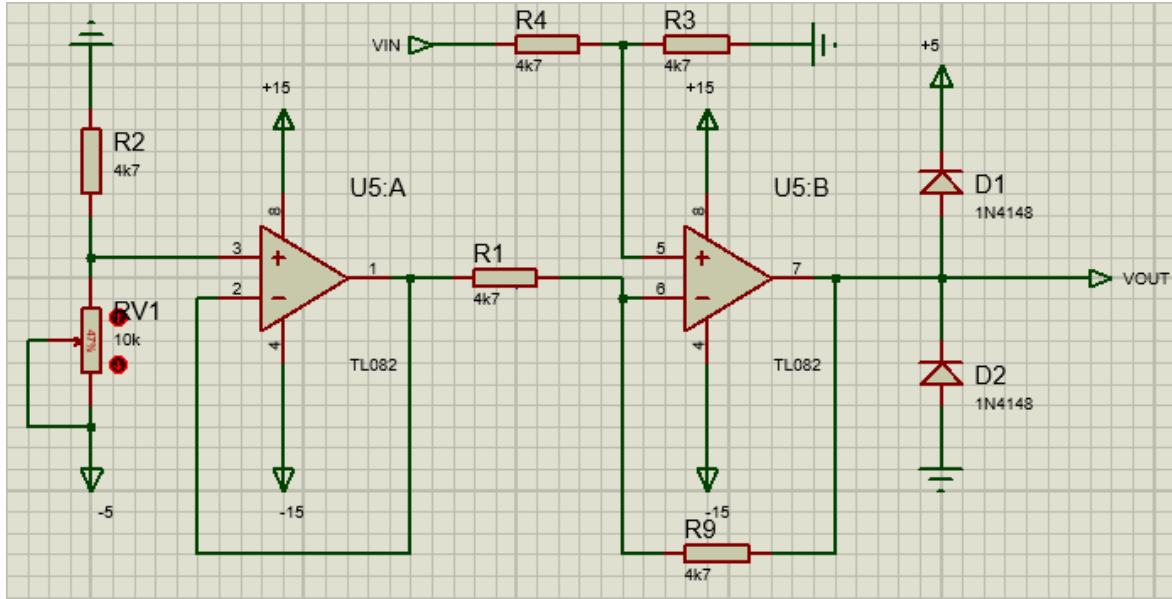


Figure 1.3: Diagrama en bloques de sistema simulado

Luego la señal ingresa al circuito de *Sample and Hold*, para mantener su valor constante mientras el convertor Analógico Digital convierte la señal. Se utilizó el ADC0808, seguido del convertor digital analógico. Para monitorear la salida digital del ADC, se conectaron "puntas lógicas", cada una leyendo un bit de la señal digital.

1.1.2 DAC

La conversión digital analógica se realizó mediante el convertor DAC0800, y a su salida se agregó una etapa de salida uni-polar recomendada en la hoja de datos del fabricante:

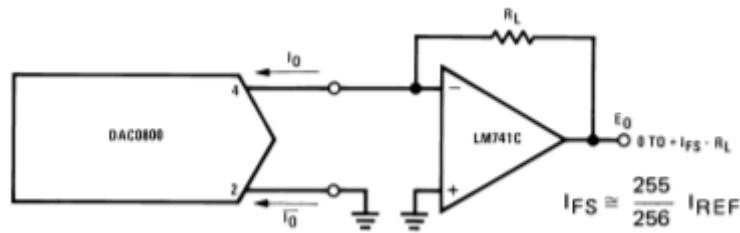


Figure 1.4: Conexión entre el DAC y el convertor de Corriente a Tensión

1.2 SEÑAL DE CLOCK

Para el convertor analógico digital utilizado, la hoja de datos especifica un tiempo de conversión típico de $100\mu s$ para una frecuencia de Clock de $640kHz$. Si se multiplican: $100\mu s \cdot 640kHz = 64$ ciclos de Clock por conversión. Si se considera el peor tiempo de conversión dado por la hoja de datos, el cálculo anterior arroja 75 pulsos de Clock por conversión. Además, debe tenerse en cuenta el tiempo de *End of Conversion*, que es igual a 8 ciclos de Clock más $2\mu s$. Por ende se llega a un número de 84 ciclos de reloj para el peor de los casos, y alrededor de 70 ciclos de reloj en el mejor de los casos.

Por otro lado se conoce que la frecuencia máxima admitida por el convertor es de $1,28MHz$, si se divide esta frecuencia por los ciclos de clock por conversión esperados: $\frac{1,28MHz}{70} = 18,28kHz$. Este valor es la cota superior para la frecuencia de muestreo. Por lo tanto, no se recomienda la utilización de este convertor si se trabaja con frecuencias de audio, las cuales van de 20 a $20kHz$.

1.2.1 CONEXIÓN ADC

Para realizar conversiones sucesivas, la hoja de datos del conversor recomienda la realimentación de la señal de EOC a la señal de *Start* del ADC. Cuando se recibe un flanco ascendente en *Start*, la señal de final de conversión se mantiene alta durante el tiempo de EOC_{delay} , y todavía no inicia la conversión. Luego, el EOC pasa de estado alto a estado bajo y comienza la conversión. Cuando esta finaliza, se la salida del EOC se invierte nuevamente, donde el flanco ascendente muestra que terminó la conversión. Al realimentar el EOC hacia la entrada *Start* se logra que el conversor tome como nuevo *Start* el flanco ascendente del fin de conversión. Como la señal de entrada debe ser estable mientras que EOC esté en un estado bajo, debe ser estabilizada con el circuito de *Sample and Hold*. Por ende, la misma señal de EOC y *Start* deben tomarse para que el Sample and Hold muestree entre conversiones y mantenga el nivel durante la conversión de sincrónicamente.

Un problema es que la duración de la señal de EOC es incierta, de entre 0 y 8 pulsos del Clock según la hoja de datos. Si este tiempo resultara menor al transitorio de *Sample* del SH, la etapa posterior de *Hold* interrumpiría el transitorio de sample, manteniendo un valor de señal incorrecto y generando errores en la muestra de la señal. A pesar de analizar específicamente la salida del sample and hold y las señales correspondientes, no se pudo encontrar en la simulación de Proteus el comportamiento descrito. No obstante, sí se observa un error en la señal final, vista con más claridad cuando se compara con una señal senoidal:

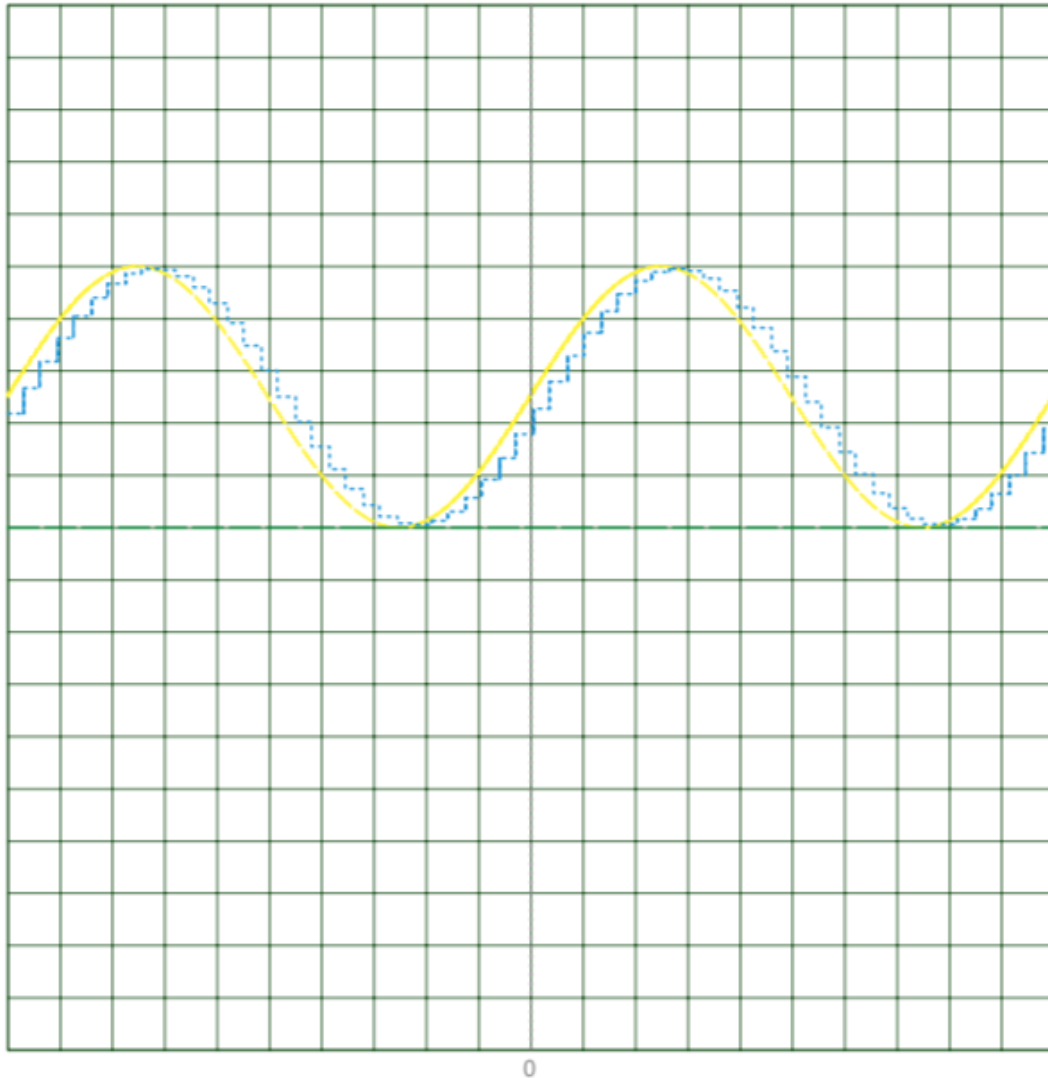


Figure 1.5: Señales de entrada y salida con error de muestreo.

Se proponen dos soluciones: la primera no muestrear la señal, lo que precisará de una frecuencia de muestreo mucho mayor a la razón de la frecuencia de la señal de entrada, reduciéndose por consecuencia el ENOB del sistema. Por otro lado, una posibilidad es no utilizar la realimentación de la señal de EOC sugerida por el

fabricante y reemplazarla por una lógica que mantenga la señal de muestreo alta por cuantos pulsos de clock se necesiten, y luego de entrar en el período de *hold* se envíe la nueva señal de *Start* al conversor. Esta solución puede ser la única manera de utilizar este conversor en frecuencias del rango de los kilohertz, pero como se explicó antes, la máxima frecuencia de muestreo posible está limitada por las especificaciones del conversor. Limitar aún más la frecuencia de las señales de entrada con el fin de evitar el uso de un sample and hold, o introducir una lógica circuital que aumente la cantidad de pulsos de reloj por cada etapa de muestreo resultará en una frecuencia de muestreo aún menor. Por ende, se concluye que este conversor no posee las características para ser utilizado en frecuencias medias, y a pesar de las limitaciones que introduce, el circuito de sample and hold se destaca como una implementación sencilla que permite operar en un rango de frecuencias razonablemente amplio.

1.3 MEDICIONES

1.3.1 FRECUENCIA DE MUESTREO

En la sección anterior se calculó la frecuencia de muestreo máxima admisible teórica: $18,28kHz$. Se quiere en primer lugar verificar que esta frecuencia sea correcta, y luego que la cantidad de clocks por conversión concuerde con la teórica (70):

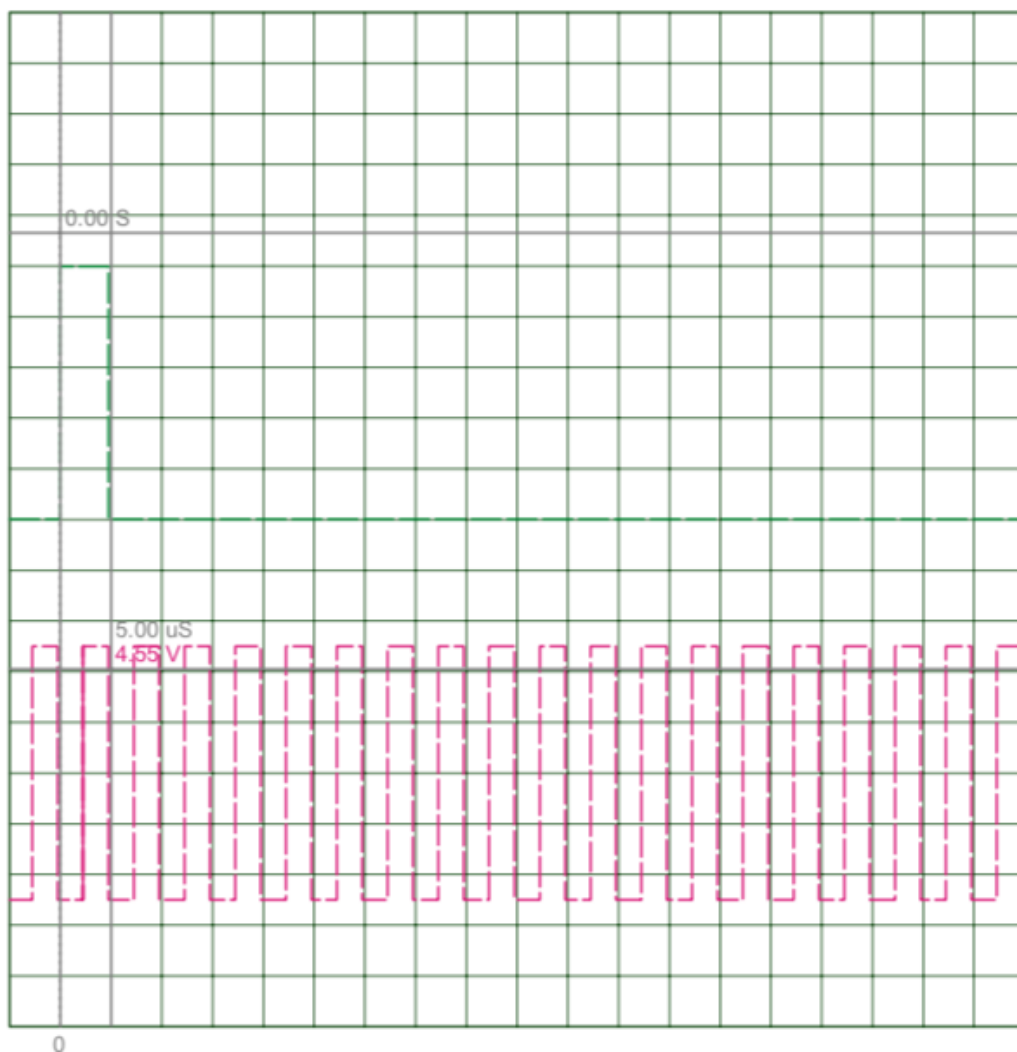


Figure 1.6: Clock y señal de EOC con zoom.

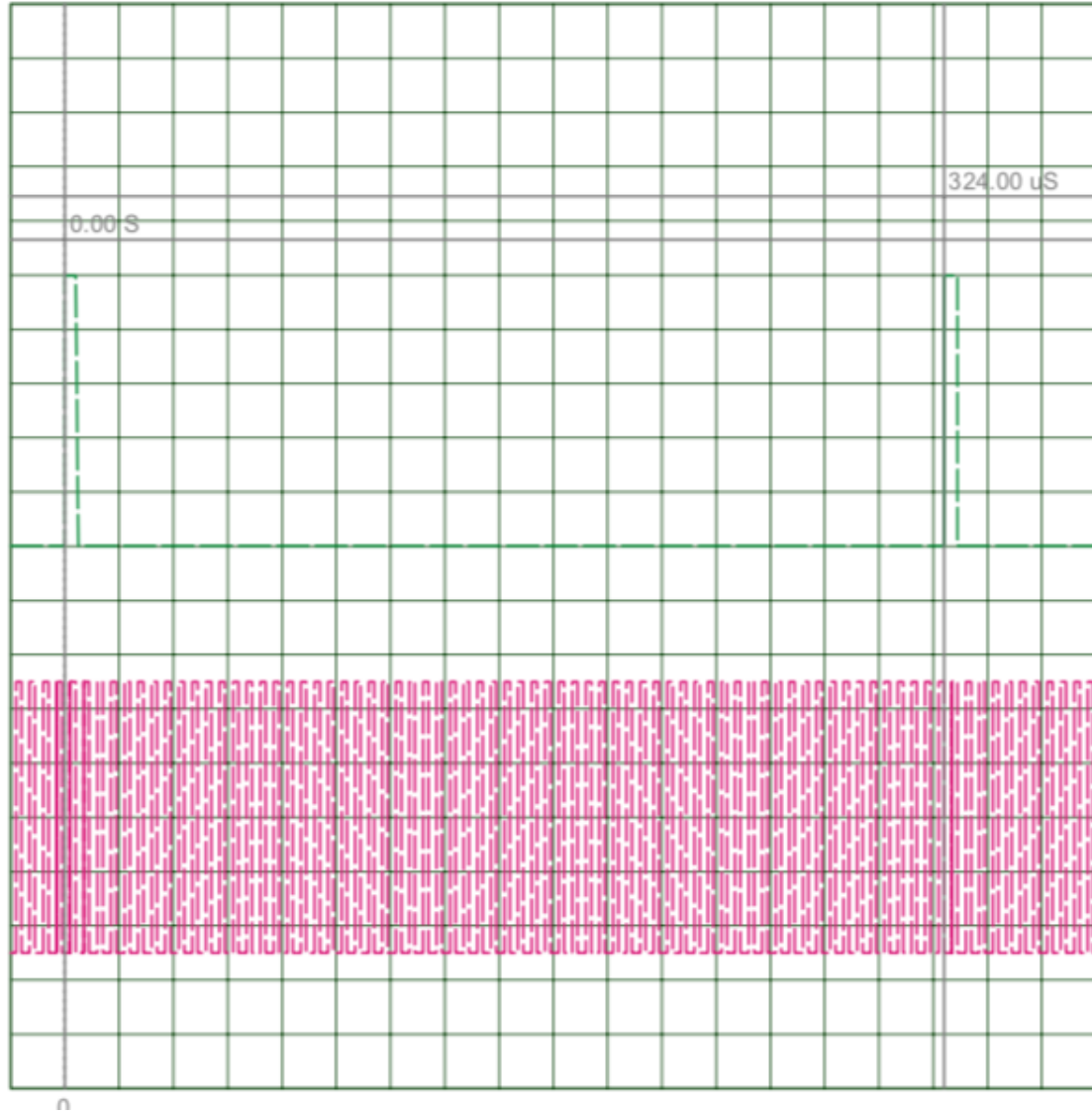


Figure 1.7: Clock y señal de EOC.

Se pudo chequear la primera afirmación con una frecuencia de clock máxima de $1MHz$, ya que al aumentarla se generaban problemas de CPU con Proteus. No obstante, con una $f_{CLK} = 1MHz$ se espera una frecuencia de muestreo de alrededor de $14,28kHz$, y la obtenida fue de $14,48kHz$ bastante próxima a la teórica. Esta frecuencia de muestreo significa una diferencia de un clock por conversión, es decir, la conversión ocurrió en una vuelta de reloj menos (69).

1.3.2 ERROR DE CUANTIZACIÓN

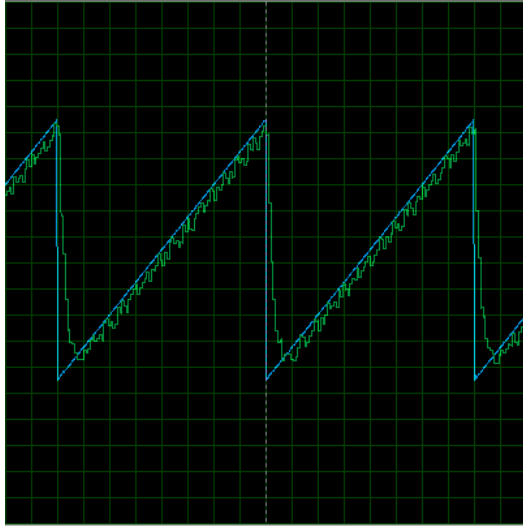
Utilizando una rampa de $100Hz$ con amplitud máxima, se propone como método para calcular el error de cuantización, restar la señal de salida a la entrada. Se obtuvo una señal con amplitud 1LSB. Para calcular el SNQR:

$$SNQR = 20 \cdot \log\left(\frac{5}{LSB}\right) \quad (1.1)$$

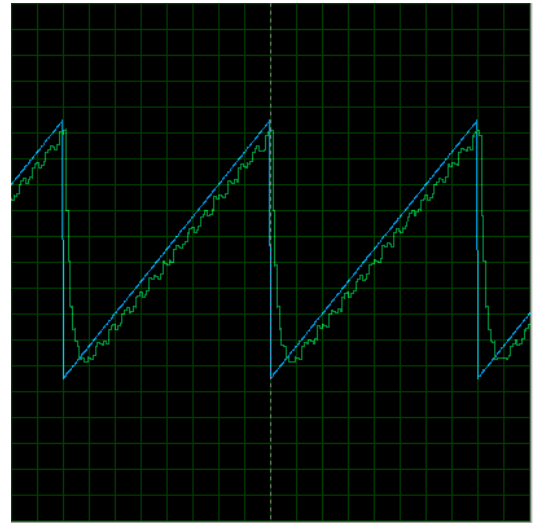
donde $LSB = V_{ref} \frac{1}{2^{bits}}$. Así se obtiene un SNQR de $48,16dB$. Para calcular el número efectivo de bits:

$$ENOB = \frac{SNQR - 1,76}{6,02} = 7,7bits \quad (1.2)$$

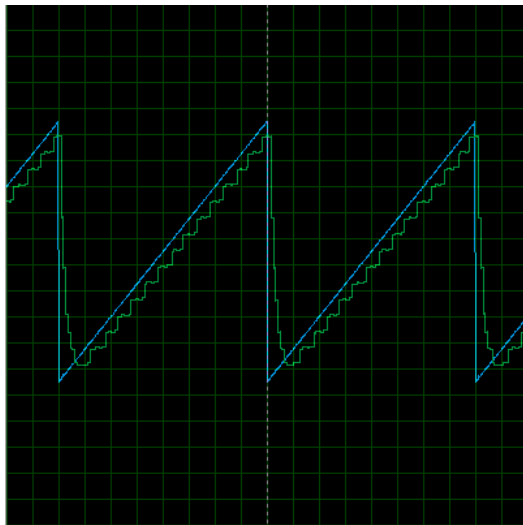
Luego se fue reduciendo el número de bits que atravesaban el bus da datos para comparar las diferentes salidas:



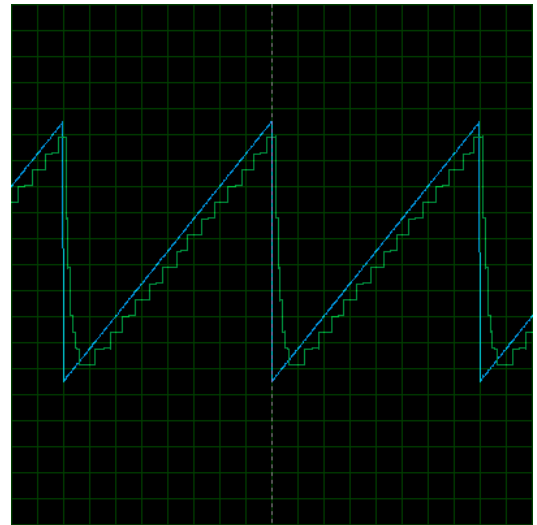
(a) 8 Bits



(b) 7Bits

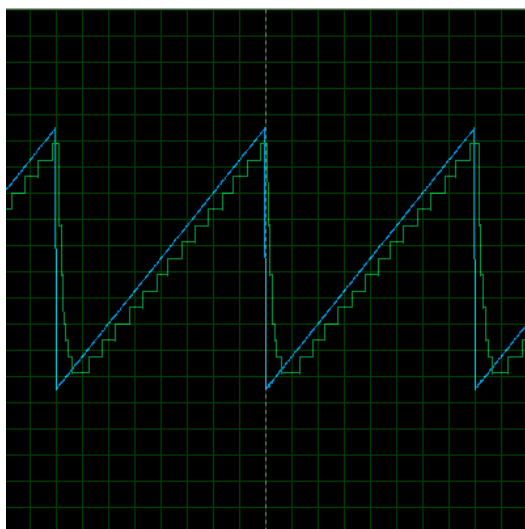


(c) 6Bits

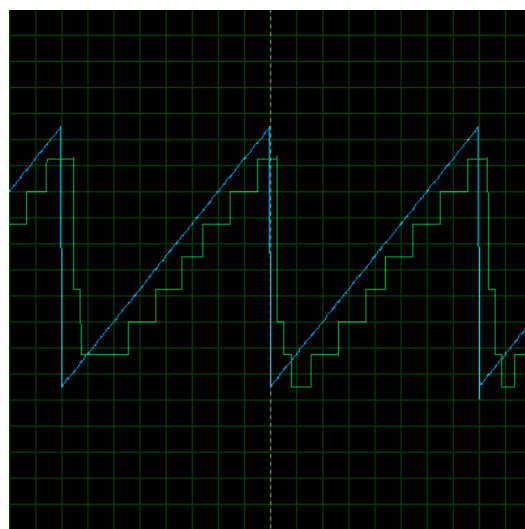


(d) 5 Bits

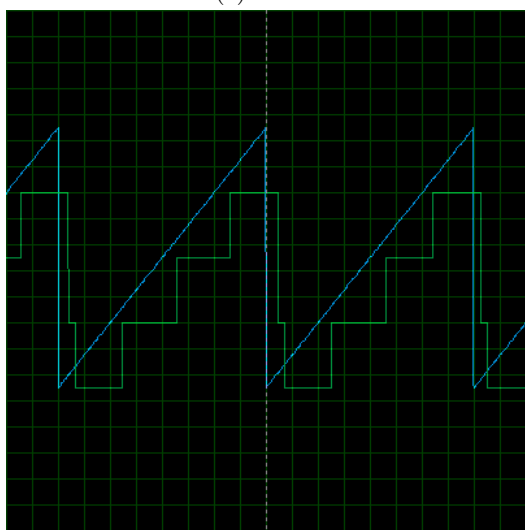
Figure 1.8: Rampa con diferente número de bits



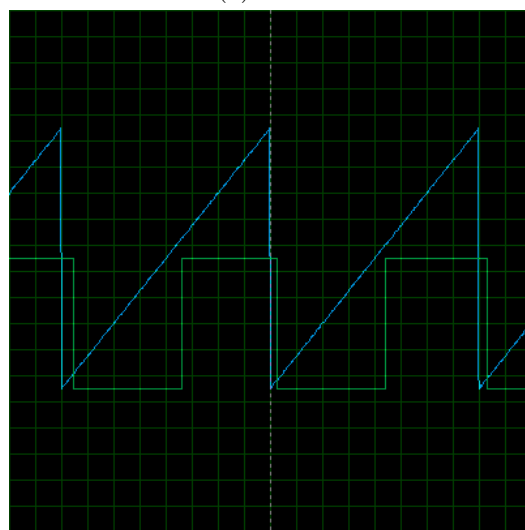
(a) 4 Bits



(b) 3 Bits



(c) 2 Bits



(d) 1 Bit

Figure 1.9: Rampa con diferente número de Bits

Puede verse claramente como el efecto de escalonado es cada vez más significativo a medida que se disminuye el número de bits. El ancho de banda teórico para una rampa es infinito, pero al elegir la frecuencia de filtrado de la señal, una frecuencia mas baja significa una rampa con pendiente más constante. La desventaja es que el flanco descendente será menos pronunciado por la falta de altas frecuencias en la transición. En conclusión se llega a una relación de compromiso entre el tiempo de fall de la rampa su pendiente.

1.3.3 TENSIONES DE ENTRADA CONSTANTE

Se ingresó al sistema con una serie de 12 tensiones constantes que barren el rango de tensión admisible del circuito. Se obtuvo el valor en binario de la digitalización de las tensiones, y las tensiones a la salida, y se compararon las tensiones de entrada con dichos valores.

$V_{in}[V]$	$V_{adc}[bin]$	$V_{dac}[V]$
-2.50	00000000	0.00
-2.48	00000001	0.16
-2.46	00000010	0.07
-2.00	00011001	0.49
-1.50	00110011	1.17
-1.00	01001100	1.31
-0.50	01100110	1.99
0.00	01111111	2.48
0.50	10011001	2.99
1.00	10110010	3.52
1.50	11001100	3.61
2.00	11100101	4.57
2.50	11111111	4.98

Table 1.1: Caption

A continuación se muestran los datos gráficamente:

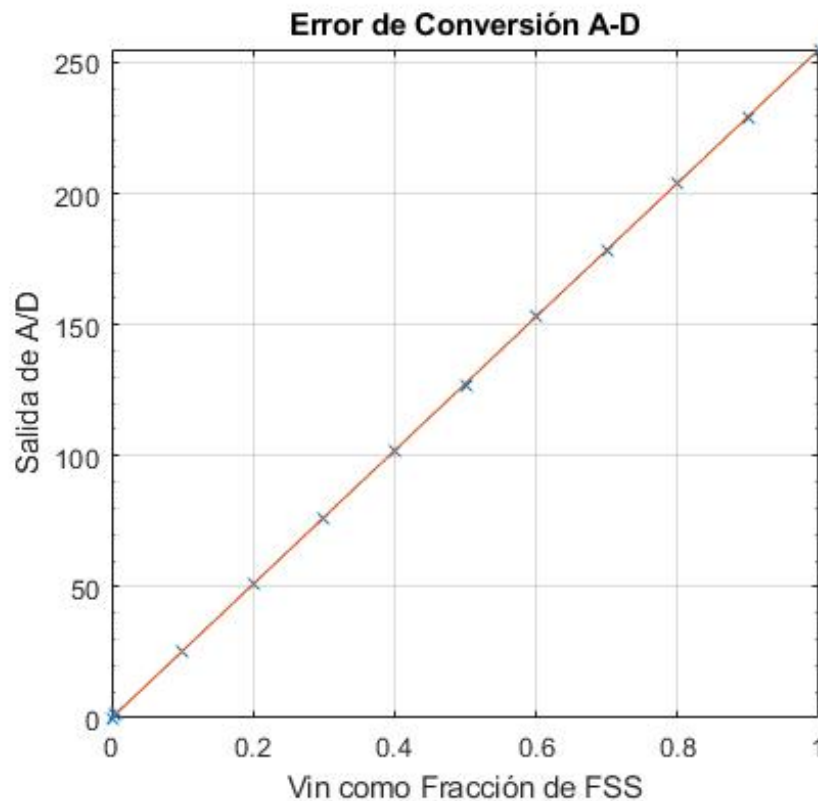


Figure 1.10

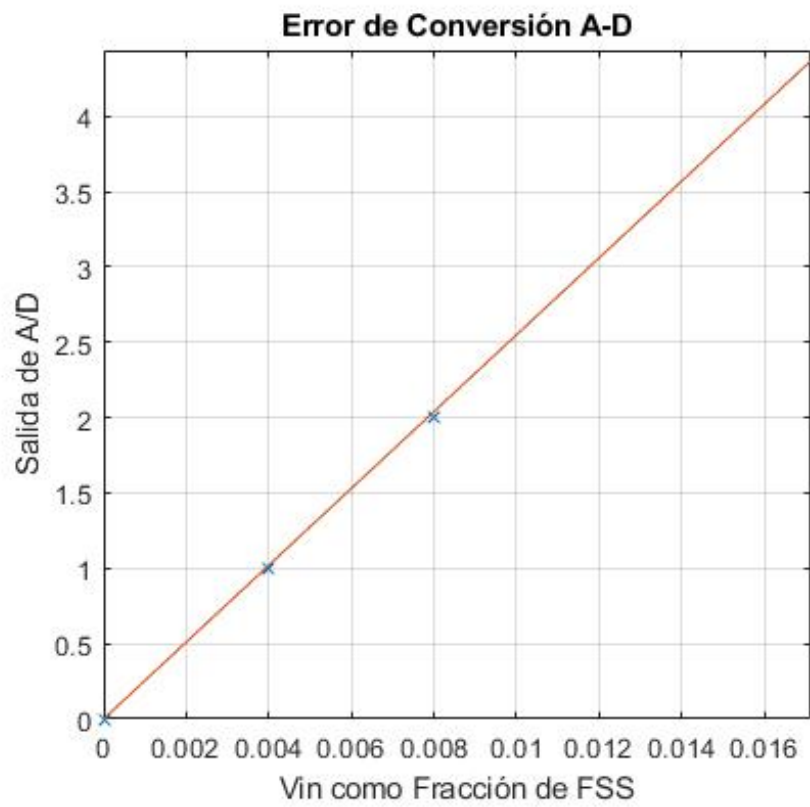


Figure 1.11

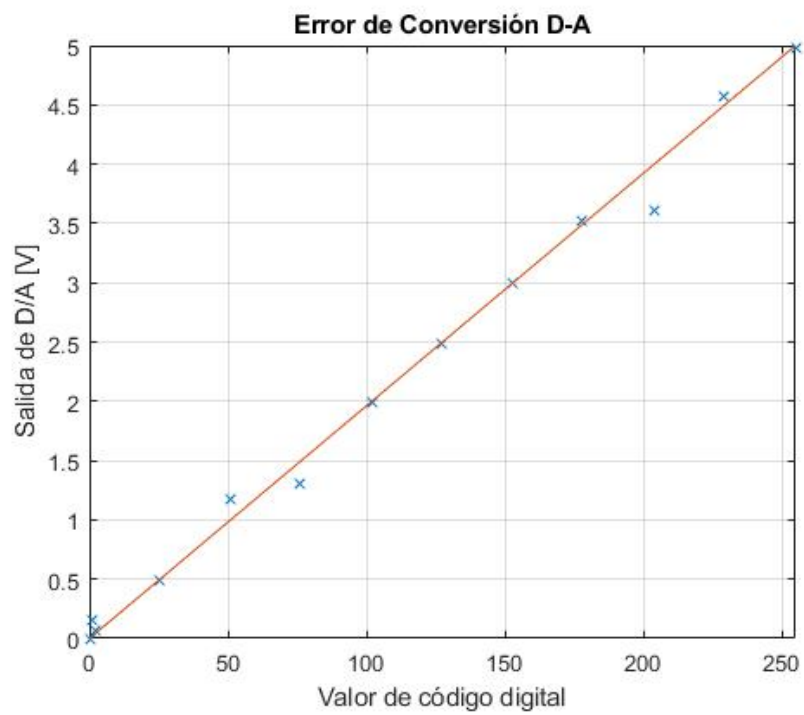


Figure 1.12

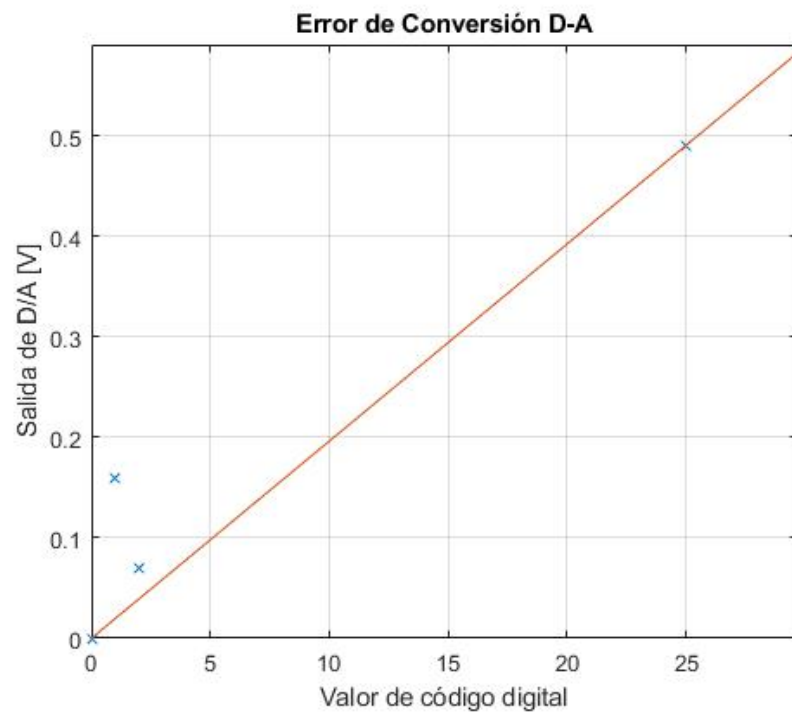


Figure 1.13

Se observa que para valores más altos de tensión la conversión es precisa. Sin embargo, para valores más bajos, la señal es más susceptible al error de conversión.

Conversor $\Sigma\Delta$

Cuando se intenta digitalizar señales cuya banda base no se extienda a un amplio rango de frecuencias, pero se requiera un excelente rango dinámico y una gran precisión tenemos que el error de cuantización de los conversores clásicos no resulta suficiente.

Para este tipo de señales se utiliza un proceso combinando un muestreo a frecuencias muy superiores de la de Nyquist (Oversampling) con conversiones de baja resolución y luego realizar un promedio de las muestras para tener una mejor resolución, junto con un cambio a la frecuencia de muestreo (Downsampling). Este tipo de tecnologías son llamadas conversores $\Sigma\Delta$.

2.1 MARCA TEÓRICO

El conversor A/D $\Sigma\Delta$ se encuentra representado genéricamente por el siguiente diagrama:

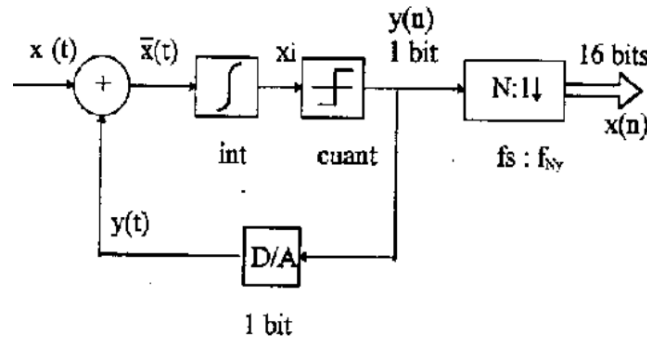
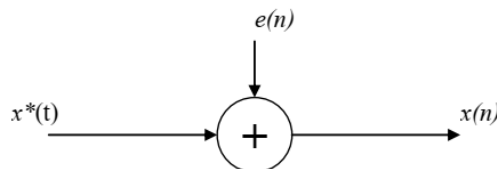


Figure 2.1: Diagrama de un conversor $\Sigma\Delta$ de primer orden

A continuación se desarrollara el comportamiento de cada componente por separado junto con las consecuencias de las condiciones de funcionamiento.

2.1.1 ERROR DE CUANTIZACIÓN

Durante toda conversión A/D de N bits se va a encontrar presente el ruido de cuantización y puede modelarse de la siguiente manera:



En donde $X^*(t)$ es la señal de entrada, $e(n)$ es el ruido de cuantización y $x(n)$ es la señal digital. La resolución del conversor se encuentra dada por el mínimo cambio de voltaje o tensión que puede realizar y esté, se encuentra dado por la tensión de fondo de escala, V_{FS} y el número de bits, n por medio de la siguiente formula:

$$LSB = V_q = q = \frac{V_{FS}}{2^n}$$

Donde LSB es valor del bit menos significativo (Least significant bit) dado que es el mínimo salto posible. El error realizado durante la digitalización se encuentra comprendido en el intervalo de $[-\frac{q}{2}, \frac{q}{2}]$ y dado que en ese rango todos los valores son igualmente probables se tiene una variable aleatoria de distribución uniformemente distribuida, lo que resulta en una secuencia de ruido blanco estacionario sin correlación con la señal. A partir de esto obtenemos que la potencia de la señal es:

$$\sigma_e^2 = \frac{1}{q} \int_{-\frac{q}{2}}^{\frac{q}{2}} e^2 de = \frac{q^2}{12}$$

Dado que ingresemos con una señal sinusoidal de amplitud máxima V_{FS} , la relación señal ruido nos queda representada como:

$$SNR_{dB} = 10 \log \left(\frac{P_S}{\sigma_e^2} \right) = (1.76 + 6.02n) dB$$

Esto nos permite tener una idea de la resolución relativa de la señal al digitalizarse.

2.1.2 OVERSAMPLING

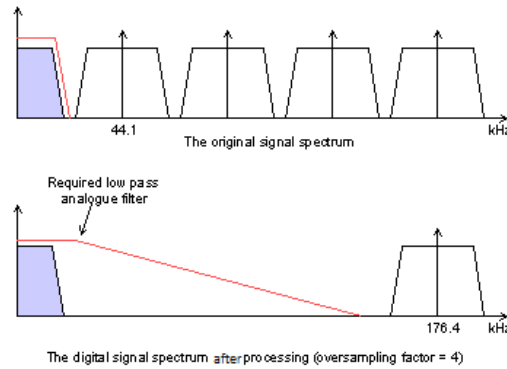
Oversampling consiste en realizar el muestreo de una señal a una frecuencia mucho mayor a la de Nyquist (f_{NY}) de forma tal que se define la siguiente relación:

$$f_s = L f_{NY}$$

Donde L suele tener un valor mayor o igual a 64 y toma valores de potencias de 2. Realizar esto nos genera las siguientes 2 ventajas:

- Relajamiento en el filtro antialiasing:

Como al aumentar la frecuencia de muestreo las repeticiones del espectro se realizan a una mayor distancia vamos a tener que el filtro antialiasing va a tener una banda de transición mayor, una mayor banda de transición significa que la curva del filtro pasabajos va a tener una pendiente de atenuación más lenta y esto se traduce en un menor orden para la realización del filtro.



Donde lo rojo indica el filtro.

- Reducción del ruido de cuantificación:

La potencia de ruido se distribuye uniformemente en todo el rango de la señal ($\frac{f_s}{2}$) con una densidad espectral de potencia $S_{ee}(f) = \frac{\sigma_e^2}{f_s}$. Entonces como la potencia de ruido es constante para la misma cantidad de bits, a una mayor frecuencia menor va a ser la densidad espectral de potencia y dado que integremos la potencia de ruido en la zona de trabajo, vamos a tener una menor potencia de ruido.

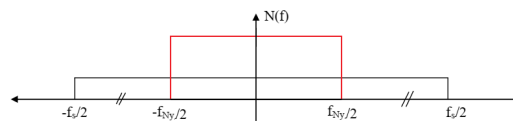


Figure 2.2: Comparación del ruido a f_{NY} vs oversampling

De la figura 2.2, podemos observar que el ruido cuando se muestrea a la frecuencia de Nyquist, curva roja, se encuentra concentrada en la zona de trabajo mientras que la curva de oversampling, curva negra, se encuentra más distribuida por el resto de frecuencias siendo menor a la anterior en la zona de trabajo. Cabe mencionar que otra consecuencia de la reducción del ruido de cuantificación es un mejoramiento de la calidad de resolución con un menor número de bits. Esta ganancia puede describirse por las siguientes ecuaciones:

$$L = \frac{f_s}{f_{NY}}$$

$$q = V_{FS} 2^{-n}$$

$$S_{ee}(f) = \frac{\sigma_e^2}{f_s}$$

$$\sigma_e^2 = \frac{q^2}{12}$$

Entonces, dado que quiera tener la misma cantidad de ruido en la zona de trabajo cuando trabajo en oversampling (f_{os}) que cuando trabajo en la zona de Nyquist (f_{NY}), voy a necesitar que las densidades de potencia sean iguales.

$$\frac{\sigma_{e_{NY}}^2}{f_{NY}} = \frac{\sigma_{e_{os}}^2}{f_{os}}$$

$$\sigma_{e_{NY}}^2 = \frac{\sigma_{e_{os}}^2}{L}$$

$$L = \frac{\sigma_{e_{os}}^2}{\sigma_{e_{NY}}^2} = \frac{q_{os}^2}{q_{NY}^2} = 2^{2(n_{NY} - n_{os})} = 2^{2\Delta n}$$

Donde n es el numero de bits utilizado para cada respectiva frecuencia de trabajo y Δn es la cantidad de bits ganados. Dado que se quiera despejar Δn reacomodamos la última ecuación y obtenemos:

$$\Delta n = 0,5 \cdot \log_2(L)$$

Por ejemplo, dado que se realizara un oversampling de $L = 64$ de 5 bits se obtiene un $\Delta n = 3$ con lo que nuestro conversor de 5 bits estaría trabajando como uno de 8 bits.

2.1.3 MODULACIÓN Y DEMODULACIÓN $\Sigma\Delta$

La etapa anterior al decimador encargada de realizar la conversión del A/D, es una derivación de la modulación $\Sigma\Delta$, y la etapa encarga después del decimador para la conversión del D/A es una derivación de la demodulación $\Sigma\Delta$.

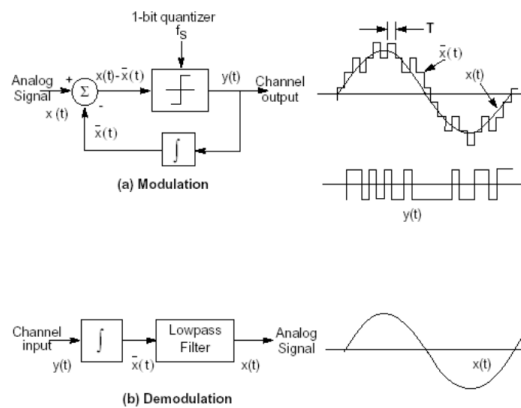


Figure 2.3: Modulación y demodulación

La modulación se realiza utilizando el esquema de la figura 2.3.a donde el integrador va a actuar como un predictor del error de cuantización y el lazo de realimentación se utiliza para ajustar la salida acorde al error de predicción cometido que luego es cuantizado. Mientras que el esquema de la figura 2.3.b el integrador va a

actuar como un contador sumando los pulsos cuantizados creando una señal con la forma de la señal de entrada original pero siendo formada por pulsos cuadrados y luego al filtrar esto por un pasa-bajos suavizando la señal se recupera la original.

Cabe agregar, que es difícil realizar un análisis del comportamiento del sistema debido a las alinealidades introducidas por el cuantizador y el origen del nombre $\Sigma\Delta$ surge de reconfigurar el sistema completo (Modulación y demodulación) poniendo adelante de la etapa de modulación el integrador utilizado en la etapa de demodulación, adquiriendo el Σ por el integrador colocado al principio y el Δ por la etapa de modulación. A continuación se expone el sistema modificado:

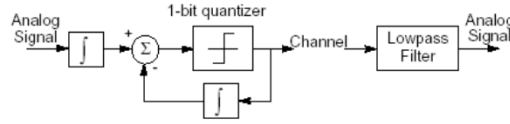


Figure 2.4: Modulación y demodulación modificada

Cabe mencionar, la modificación realizada a la modulación y demodulación puede simplificarse cambiando los integradores de la etapa de modulación por un solo integrador adelante del cuantizador de 1-bit y que esto realiza una alteración al ruido de cuantización que se vera en mayor profundidad en la sección siguiente.

Finalmente, el sistema puede realizarse de varios ordenes siendo el expuesto anteriormente el de primer orden. Donde el aumento de ordenes genera un aumento de realimentaciones como se puede observar en la siguiente figura:

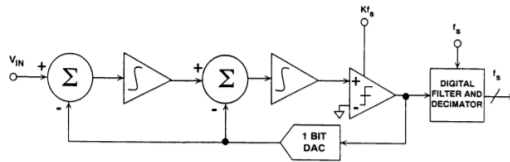
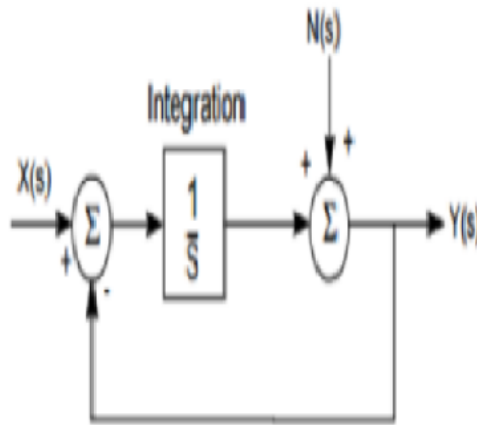


Figure 2.5: Conversor $\Sigma\Delta$ de segundo orden

2.1.4 NOISE SHAPING

Evaluando la respuesta en frecuencia del esquema de la modulación $\Sigma\Delta$ para la entrada $X(t)$ y el ruido $N(t)$ a partir del siguiente esquema:

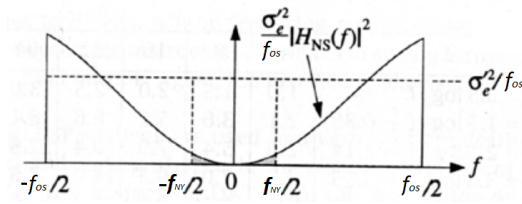


Obtenemos las siguientes funciones transferencias:

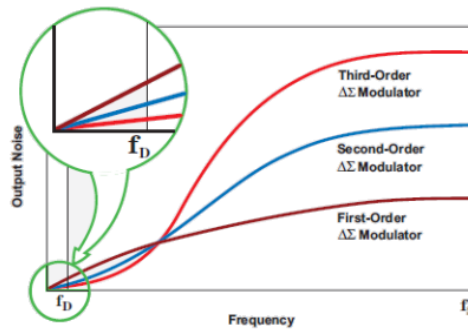
$$\frac{Y(S)}{X(S)}|_{N(S)=0} = \frac{1}{S+1}$$

$$\frac{Y(S)}{N(S)}|_{X(S)=0} = \frac{S}{S+1}$$

Se observa que el sistema va a comportarse como un filtro pasa bajos para la señal de entrada y como un filtro pasa altos para el error de cuantización. Luego, el espectro del error de cuantización va a modificarse de la siguiente manera:



Se puede observar que al combinarlo con oversampling tenemos una considerable disminución del ruido de cuantización en la zona de trabajo. Cabe mencionar, que el filtro generado para el ruido por el sistema cambia dependiendo del orden del mismo. A continuación se ilustra el mismo para diferentes ordenes:



2.1.5 FILTRO DECIMADOR

La ultima etapa utilizada en el conversor es el filtro decimador, el cual se encarga de bajar la frecuencia de oversampling a una menor frecuencia de sampleo cercana a la de Nyquist dando como resultado una mayor precisión, remover el ruido de cuantización fuera de la banda base aumentando la resolución efectiva de la salida digital y actuar como un filtro anti-alias para el ruido de cuantización (Noise shaping) generado por el modulador. Además, debido a que el oversampling se realiza a altas frecuencia sirve para facilitar el proceso de señales digitalmente (DSP) y no se tiene perdida de información al decimar.

La decimación realiza un promedio de 2^N muestras dado que se quiera realizar una resolución de N bits y quiera disminuirse la frecuencia 2^N veces, de forma tal que si se quiere realizar una decimación de 64 veces se realiza un promedio de 64 muestras y se pasa a tener de una resolución de 1 bit a 6 bits. La decimación se divide en 2 etapas:

- El filtro pasa bajos y el decimador:

El filtro pasa bajos cumple el papel de filtro anti-alias y disminuir el ruido que se encuentra fuera de la banda de trabajo y realiza un promedio de muestras. Normalmente, se combina la realización del decimador y el filtro pasa bajos realizando un filtro comb no recursivo con coeficientes unitarios por medio de etapas en cascada (CIC: Cascaded integrator Comb) debido a su fácil implementación. Su función transferencia es de la forma:

$$H(Z) = \left(\frac{1 - Z^{-L}}{1 - Z^{-1}} \right)^N$$

Donde L es el factor de oversampling y N la cantidad de etapas utilizadas. La ventaja de su utilización es que permite el trabajo a altas frecuencia en tiempo real pero la desventaja es que no permite eliminar el ruido de cuantización de manera efectiva. A continuación se muestra una manera de implementarlo:

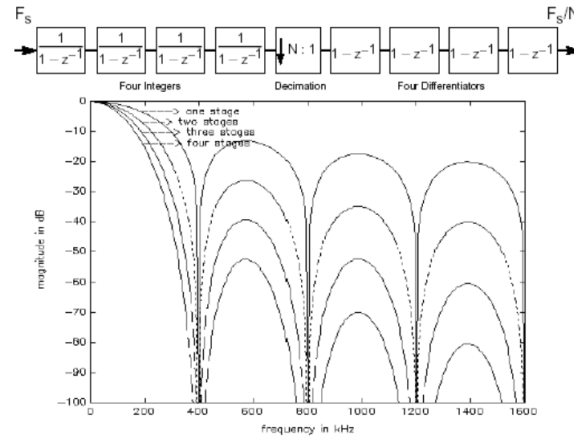


Figure 2.6: Filtro decimador CIC

- El filtro compensador:
El filtro de compensación depende del filtro pasa bajos utilizado y busca preservar la fase lineal de la señal, obtener la atenuación necesaria en la banda atenuada y compensar cualquier atenuación realizada en la banda de paso. Se implementa por medio de un filtro FIR el cual debido a que se utiliza después de la etapa anterior trabaja ya a la frecuencia decimada y por lo tanto tiene una mayor cantidad de tiempo para trabajar lo que permite realizarlo de un mayor orden que trabajando a la frecuencia de oversampling.

Cabe aclarar que filtro decimador y decimador son diferentes, el filtro decimador incluye tanto el filtro pasa bajos como el decimador mientras que el decimador por si solo, produce el downsampling que consiste en la disminución de la frecuencia.

2.2 SIMULACIÓN

Se realizó la simulación a través de python de un conversor ADC sigma delta de primer orden, de donde se obtuvieron las siguientes señales:

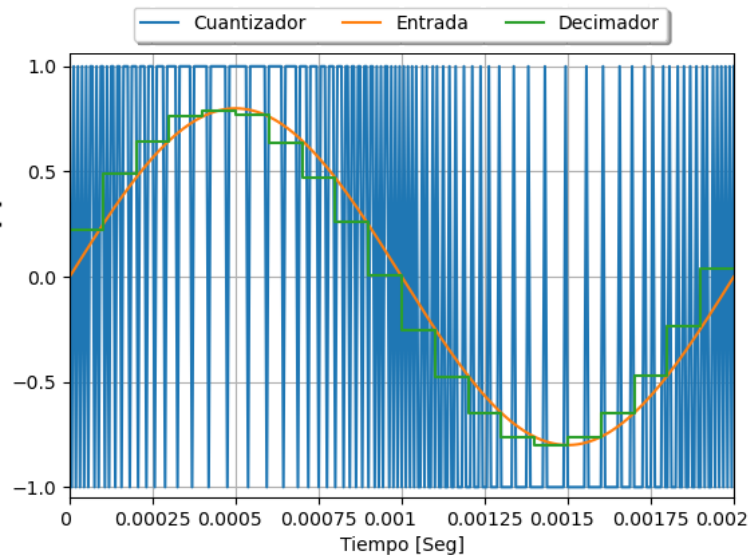


Figure 2.7: Simulación en el tiempo

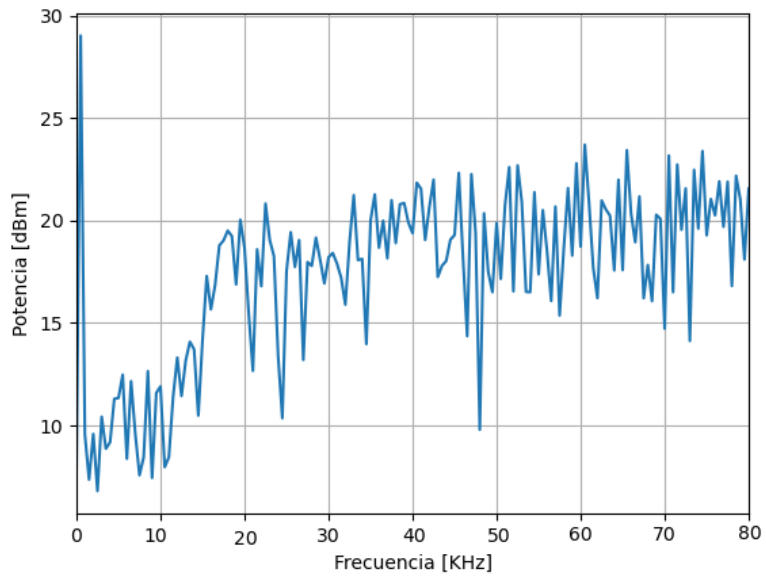


Figure 2.8: Simulación noise shaping

La simulación fue realizada para una señal seno de frecuencia 500 Hz, una frecuencia de oversampling de 320 veces superior a la de la señal de entrada ,160 KHz y se realizo una decimación por un factor de 16. Se utilizaron las librerías scipy, numpy y matplotlib para realizar la simulación.