

Convertidores $\Sigma\Delta$ para microcontroladores

J. Arias

30 de enero de 2007

Índice

1. Introducción.	1
2. Teoría del convertidor $\Sigma\Delta$ [6].	2
3. Implementación. Efectos reales.	4
3.1. Rango de entrada.	4
3.1.1. Resistencias de Pull-Up.	5
3.1.2. Calibración.	6
3.2. Histerésis.	6
3.3. Tonos. Ruido en la alimentación.	7
3.4. Jitter.	8
4. Filtrado y diezmado [5].	8
4.1. Filtro FIR	10
4.2. Filtro IIR	11
4.3. Requisitos computacionales.	11
4.4. Aún más sencillo: Filtro tipo peine (Comb)	12
5. Ampliaciones.	14
5.1. Múltiples convertidores.	14
5.2. Mayor resolución.	15
6. Modulador de segundo orden.	15
6.1. Modulador de segundo orden experimental	17
7. Conclusión.	18

1. Introducción.

En el amplio campo de aplicaciones de los microcontroladores a menudo topamos con la necesidad de medir una tensión analógica con una precisión razonablemente alta, pero con una frecuencia de muestreo baja. Un buen ejemplo podría ser la medida de una temperatura. Los cambios de la temperatura son lentos y por lo tanto no necesitamos una frecuencia de muestreo alta, bastaría con tomar una muestra cada segundo, o incluso cada minuto, pero si queremos una medida con una resolución de décimas de grado necesitamos un convertidor de 10 o más bits.

Muchos microcontroladores incorporan convertidores A/D de aproximaciones sucesivas de 8 o 10 bits pero este tipo de convertidores, además de encarecer el microcontrolador, no es el más adecuado para la aplicación en cuestión. Es difícil obtener 10 bits de resolución sin dotar al circuito de un buen filtro pasabajos en la entrada y un buen apantallamiento. Los convertidores de aproximaciones sucesivas requieren que la señal de entrada permanezca constante durante todo el tiempo que dura la conversión. Si dicho convertidor carece de un amplificador sample & hold los últimos bits de la conversión pueden ser erráticos.

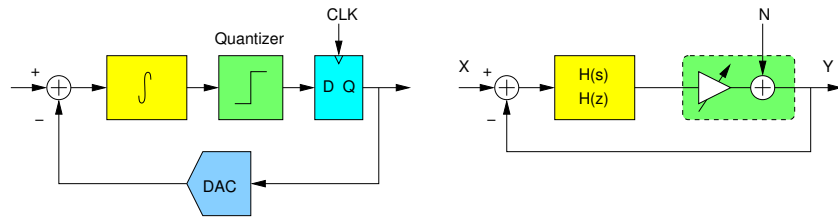


Figura 1: Diagrama de bloques del modulador sigma-delta y su circuito equivalente linealizado.

En este documento se presenta una solución alternativa de muy bajo coste: Un convertidor sigma-delta implementado en su mayor parte por un programa que corre en el microcontrolador. Los únicos componentes analógicos que se necesitan son un par de resistencias y un condensador, y sus valores no son nada críticos. Con este convertidor se pueden obtener resoluciones de 10 o más bits para frecuencias de muestreo del orden del hercio, y una linealidad y monotonicidad garantizadas.

2. Teoría del convertidor $\Sigma\Delta$ [6].

En la figura 1 se muestra el esquema de un modulador sigma delta de primer orden. Este modulador convierte la señal de entrada en un tren de pulsos cuya media temporal es proporcional a la tensión de la entrada. Estos pulsos se procesarán mediante un filtro digital en el microcontrolador para separar la señal de interés del ruido.

La frecuencia del reloj del modulador ha de ser bastante superior al ancho de banda de la señal para obtener resultados satisfactorios. Según el teorema de Shannon la frecuencia de muestreo del convertidor ha de ser el doble del ancho de banda de la señal de entrada. En nuestro caso vamos a considerar un ancho de banda de 1 Hz y una frecuencia de reloj de 256 Hz, lo que nos da un factor de sobremuestreo, OSR, de 128. Un modulador sigma-delta de primer orden puede alcanzar un máximo de relación señal-ruido, SNR, dado por [4]:

$$SNR_{\Sigma\Delta} = 30 \log OSR + 2,6 \text{ (dB)}$$

Lo que nos da un SNR de 65.8 dB. Un convertidor con una frecuencia de muestreo igual al doble del ancho de banda, y N bits de resolución tiene un SNR de:

$$SNR_{Nyquist} = 6,02 \cdot N + 1,76 \text{ (dB)}$$

Esta ecuación nos permite definir el número “efectivo” de bits de un convertidor como:

$$N_{efectivo} = \frac{SNR(dB) - 1,76}{6,02}$$

Que para nuestro modulador sigma delta resulta ser de 10.6 bits. Los 0.6 bits extra quieren decir que si con este modulador queremos hacer un convertidor de más de 10 bits el bit nº 11 está aún algo correlacionado con la señal, pero los siguientes bits son aleatorios. O visto de otro modo: El convertidor es capaz de discernir $2^{10,6} = 1550$ niveles de tensión distintos, que resulta que no son una potencia de 2.

Pero intentemos obtener estos resultados del análisis del modulador de la figura 1. Para ello intentaremos sustituir el cuantizador del modulador de la izquierda por un modelo lineal en el que la salida es la suma de la señal de la entrada, multiplicada por una cierta ganancia, y del ruido de cuantización (N). El cuantizador simplemente mira el signo de la señal de la entrada y genera una salida +1 si la entrada es positiva o -1 si es negativa. Y aquí empiezan las dificultades. En primer lugar el error de cuantización no es propiamente un “ruido”, pues está bastante correlacionado con la entrada. Por otro lado no resulta fácil encontrar una ganancia para la señal. La salida es la misma para señales de poca o mucha amplitud, ya que sólo el signo cuenta. Esto nos lleva a suponer que la ganancia ha de ser variable, dependiente de la amplitud de la entrada, lo que hace no lineal nuestro modelo “linealizado”. El error de concepto que tenemos al plantear la ganancia del cuantizador de esta forma es que pretendemos obtener una ganancia “instantánea”, cuando hemos dicho que la señal de salida es el promedio de un número grande de pulsos del cuantizador. Es mejor usar esta nueva definición

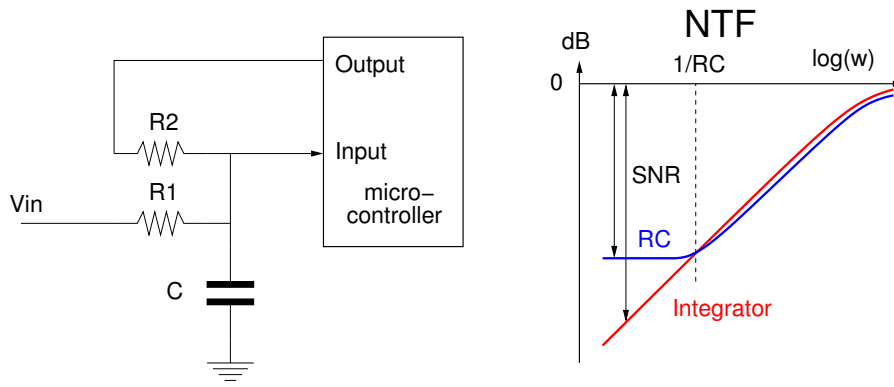


Figura 2: (a) Convertidor propuesto. (b) NTF para moduladores sigma-delta con integrador y con red RC.

de la ganancia:

$$A_Q = \frac{\sum_0^{\infty} O_i}{\sum_0^{\infty} I_i}$$

Donde O_i e I_i son una secuencia de tensiones de salida y de entrada del cuantizador para una entrada constante en el modulador. El problema ahora es que no podemos conocer la ganancia efectiva del cuantizador a priori, ya que ésta dependerá del tipo de modulador en el que se utilice. Por lo tanto resulta necesario hacer una simulación del modulador para obtener un valor para éste parámetro y tener un modelo lineal decente, pero si hacemos una simulación ya no necesitamos el modelo lineal. La simulación es más fidedigna. En cualquier caso el modelo lineal nos puede aportar ideas acerca del comportamiento del circuito, así que continuaremos suponiendo una ganancia, no conocida a priori, A_Q , para el cuantizador. La salida del modulador, Y , será:

$$Y = \frac{A_Q H(s)}{1 + A_Q H(s)} X + \frac{1}{1 + A_Q H(s)} N$$

Donde X es la señal de la entrada y N el ruido introducido por la cuantización. Podemos distinguir dos funciones de transferencia distintas, una para la señal y otra para el ruido. Estas funciones son:

$$STF = \frac{A_Q H(s)}{1 + A_Q H(s)}$$

$$NTF = \frac{1}{1 + A_Q H(s)}$$

La función de transferencia del ruido de cuantización, NTF, tiene una gran importancia en un modulador sigma-delta. Debe ser del tipo filtro pasa-altos, de modo que el ruido no corrompa la señal de interés que está en las frecuencias bajas. La función de transferencia de la señal, STF, debe tener una respuesta plana en la banda de interés.

Supongamos ahora que en el modulador tenemos un integrador con una función de transferencia $H(s) = \frac{1}{s/\omega_0}$, y que la frecuencia de muestreo es alta para poder despreciar el efecto del muestreo en el circuito. Nos queda:

$$NTF = \frac{1}{A_Q} \frac{s/\omega_0}{1 + \frac{s}{A_Q \omega_0}}$$

La función de transferencia del ruido tiene un cero en el origen, lo que da la característica de filtro pasa-altos que queremos. La función de transferencia de la señal tiene una ganancia de 1 y un polo en $A_Q \omega_0$.

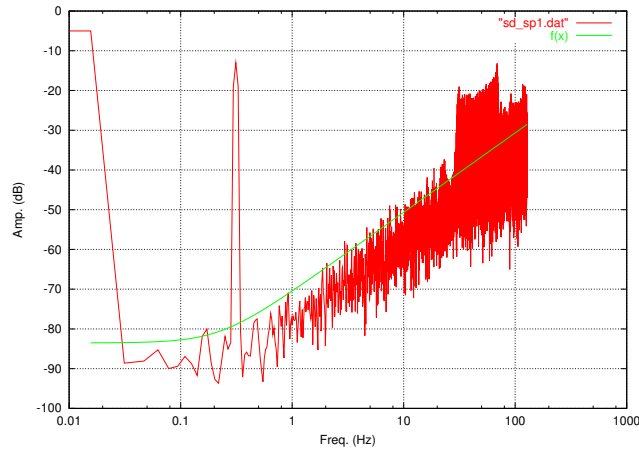


Figura 3: Espectro de la simulación de la salida del modulador. También se muestra la función NTF.

Pero un integrador es un lujo excesivo para nuestro convertidor de extra-bajo coste. En lugar del integrador usaremos una red R-C, que para frecuencias altas tiene un comportamiento parecido. Su función de transferencia es: $H(s) = \frac{1}{1+RCs}$. La función NTF es ahora:

$$NTF = \frac{1}{A_Q + 1} \frac{1 + RCs}{1 + \frac{RC}{A_Q + 1}s}$$

La principal diferencia con la NTF del modulador con integrador es que ahora el cero no está en el origen, sino a la frecuencia $\omega = 1/RC$, de modo que no tenemos una atenuación infinita para componentes continuas. En la figura 2(b) se muestran ambas funciones de transferencia y la conclusión que podemos obtener de este análisis es que si queremos que el ruido de cuantización en la banda de interés tenga un nivel razonablemente bajo la frecuencia de corte del filtro RC ha de ser igual al ancho de banda de la señal o menor. Por otra parte la ganancia de la STF para frecuencias bajas no es 1 sino $\frac{A_Q}{A_Q+1}$, lo que introduce un error de ganancia, que se puede despreciar, ya que la simulación dió $A_Q = 15000$.

La figura 3 muestra el resultado de la simulación del modulador propuesto. En esta simulación la señal de entrada ha sido una senoide pura superpuesta sobre un nivel de continua. El tren de pulsos generado por el modulador se ha convertido al dominio de la frecuencia mediante una FFT. Como se puede comprobar, por debajo de 1 Hz el ruido de cuantización está muy atenuado, mientras que para frecuencias mayores crece a razón de 20 dB por década.

Resumiendo: con una simple red RC y un sobremuestreo de 128 se puede obtener un convertidor sigma delta de más de 10 bits de resolución, ya que la relación señal-ruido, SNR, que se obtiene de la simulación supera los 60 dB. La relación señal-ruido se calcula integrando el espectro desde frecuencias un poco superiores a cero hasta 1 Hz, teniendo cuidado de excluir de dicha integral la frecuencia de la señal de prueba.

3. Implementación. Efectos reales.

La figura 2(a) muestra el esquema de circuito del convertidor propuesto, y este no puede ser más simple. En el microcontrolador se programa una interrupción que tiene lugar 256 veces por segundo (o a una frecuencia similar), y en la rutina de interrupción se lee el valor del bit de entrada, que es la señal cuantizada del modulador, y se escribe su complemento en el bit de salida. La secuencia de bits generada se almacena para su posterior filtrado y diezmado. De cada 128 muestras de 1 bit se generará una única muestra de 16 bits, de los que al menos los 10 más significativos serán correctos.

Hasta aquí la teoría, ahora veamos algunos efectos reales que pueden echar por tierra nuestras pretensiones.

3.1. Rango de entrada.

Dado que la ganancia del cuantizador es muy grande (alrededor de 15000) podemos suponer que la tensión en su entrada es nula (una tierra virtual), pero nula aquí significa que es igual a la tensión umbral del cuantizador. En nuestro circuito el cuantizador es directamente el buffer del pin de entrada del microcontrolador, y la mencionada tensión umbral

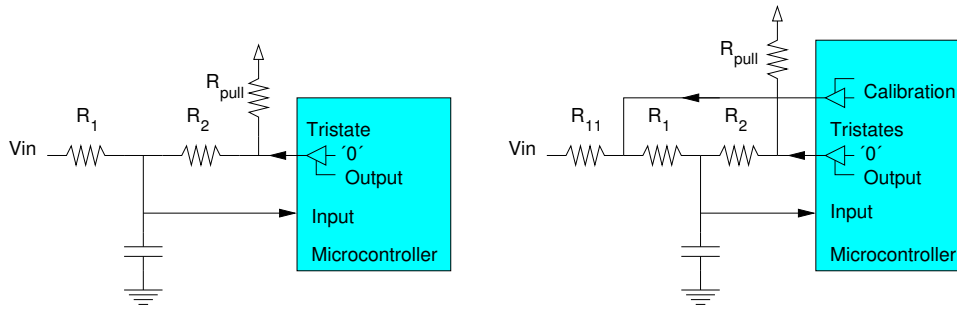


Figura 4: Mejoras en el modulador $\Sigma\Delta$. (a) Inclusión de una resistencia de Pull-up (R_{pull}) para ajustar el rango de entrada. (b) Configuración para la calibración automática del convertidor.

rondará la mitad de la tensión de alimentación, pero puede ser bastante imprecisa. En el microcontrolador PIC16F84 esta tensión es de alrededor de 1.4 Voltios para una tensión de alimentación de 5 Voltios. Llamaremos V_{TH} a la tensión umbral del pin de entrada. Se cumplirá:

$$V_{TH} = V_{IN} \frac{R_2}{R_1 + R_2} + \widehat{V_{OUT}} \frac{R_1}{R_1 + R_2}$$

Donde $\widehat{V_{OUT}}$ es la media temporal de la tensión del pin de salida. Dado que V_{TH} es constante podemos despejar V_{IN} , que será:

$$V_{IN} = V_{TH} \left(1 + \frac{R_1}{R_2} \right) - \widehat{V_{OUT}} \frac{R_1}{R_2}$$

Nos interesa ahora conocer la tensión de entrada para tres casos concretos de la tensión media de salida: $\widehat{V_{OUT}} = [0, V_{DD}/2, V_{DD}]$, que se corresponden con los valores máximo, medio y mínimo del convertidor. Supongamos que nuestro microcontrolador sigue siendo el PIC16F84, con $V_{TH} = 1,4V$ y $V_{DD} = 5V$. Si $R_1 = R_2$, tenemos:

$$V_{INMAX} = 1,4 * 2 = 2,8 \text{ Voltios}$$

$$V_{INmed} = 1,4 * 2 - 2,5 = 0,3 \text{ Voltios}$$

$$V_{INmin} = 1,4 * 2 - 5 = -2,2 \text{ Voltios}$$

Como podemos comprobar el rango de entrada que tenemos no queda precisamente muy bien centrado, es muy probable que las tensiones de entrada negativas no sean de ninguna utilidad, aunque en alguna aplicación concreta sean una ventaja. Por otra parte, el rango de entrada $V_{INMAX} - V_{INmin}$ es $V_{DD} \frac{R_1}{R_2}$, que NO depende de V_{TH} , lo cual es un alivio, ya que la tensión umbral puede variar mucho de un chip a otro o con la temperatura. Las variaciones de V_{TH} pueden cambiar el origen de coordenadas del convertidor, pero no su ganancia. Esto es importante, ya que podremos corregir los errores debidos a V_{TH} simplemente restando un valor de calibración.

3.1.1. Resistencias de Pull-Up.

Cuando V_{TH} es demasiado pequeña el rango de entrada incluye una buena parte de tensiones negativas que normalmente no aparecen en los circuitos. El problema aquí es que la salida del micro puede poner bastante carga en el condensador cuando está en alto, pero sólo puede retirar una poca cuando está en bajo. Esta asimetría se debe a que V_{TH} está por debajo de $V_{DD}/2$, y una posible forma de corregirla es utilizando resistencias de Pull-Up en la salida de realimentación.

En la figura 4(a) se muestra la implementación del modulador con una resistencia de Pull-up. La salida del microcontrolador debe ser de tipo triestado, de modo que cuando queremos escribir un cero programamos el pin como salida y escribimos el cero, pero cuando queremos poner un uno simplemente programamos el pin como entrada. De este modo la resistencia que hay entre el condensador y la tensión de realimentación es menor cuando se escribe un cero, y esto contribuye a descargar más rápido el condensador. El efecto neto de este montaje es el de desplazar el rango de entrada hacia las tensiones positivas.

Para cubrir el rango de entrada que va desde 0 hasta V_{DD} las relaciones entre las resistencias deben ser:

$$\frac{R_{pull}}{R_2} = \frac{V_{DD}^2}{V_{TH}^2} - 2 \frac{V_{DD}}{V_{TH}}$$

$$\frac{R_1}{R_2} = \frac{V_{DD}}{V_{TH}} - 1$$

Si $V_{TH} = V_{DD}/2$ nos da $R_{pull} = 0$ y $R_1 = R_2$.

Por ejemplo, en el caso del microcontrolador PIC16F84, con $V_{TH} = 1,4V$ y $R_2 = 1M\Omega$ nos queda: $R_{pull} = 5,6M\Omega$ y $R_1 = 2,57M\Omega$.

Es conveniente que el rango de entrada real supere de forma holgada al necesario. Las dos razones que nos obligan a ello son por una parte la incertidumbre en el valor de V_{TH} , y por otro el hecho de que los moduladores sigma-delta se vuelven muy ruidosos cuando la tensión de entrada se acerca a sus límites. Para aumentar el rango de entrada haremos un poco más grande el ratio R_1/R_2 .

Cuando un microcontrolador ofrece pines con pull-ups internos no conviene utilizarlos para esta aplicación. El valor de la resistencia del pull-up es muy poco precisa, y por lo general dichos pull-ups se fabrican con transistores MOSFET que tienen un comportamiento muy poco lineal y podrían introducir distorsión en el modulador.

3.1.2. Calibración.

La figura 4(b) muestra las modificaciones necesarias para tener disponibles medidas de calibración del convertidor. La salida de calibración está en alta impedancia durante el funcionamiento del convertidor, y cuando se quiere hacer una medida de calibración se programa como salida. Durante la calibración se pueden medir dos tensiones: la de tierra cuando la salida está en cero y V_{DD} cuando está en uno. Con dichas medidas se puede calcular el valor exacto de V_{TH} y corregir las medidas del convertidor. Si llamamos V_{GND} a la medida de tierra, V_{FS} a la medida de fondo de escala (V_{DD}), V_{AD} a la medida sin corregir del convertidor, y V_{CORR} a la medida corregida, tenemos:

$$V_{CORR} = (V_{AD} - \alpha V_{GND}) \frac{\beta \cdot FS}{V_{FS} - V_{GND}}$$

Donde FS es el código que queremos hacer corresponder al fondo de escala y α, β son dos factores de corrección, que si $R_{11} \ll R_1$ son próximos a uno. El valor de estos factores es:

$$\alpha = \left[\frac{R_{11}}{R_1 + R_{11}} \frac{R_1}{R_1 + R_2} + \frac{R_1}{R_1 + R_{11}} \right]$$

$$\beta = 1 + \frac{R_{11}}{R_1}$$

Por ejemplo, si $R_{11} = 1K\Omega$, $R_1 = 1M\Omega$ y $R_2 = 3M\Omega$ tenemos $\alpha = 0,99975$, $\beta = 1,001$

El pin utilizado como salida de calibración debe ser de tipo CMOS para garantizar que la tensión de calibración es 0 o V_{DD} . Las salidas de tipo TTL no son buenas para este propósito ya que no son capaces de garantizar estos niveles de tensión.

3.2. Histéresis.

Al elegir el pin de entrada del microcontrolador hemos de tener muy presente que dicha entrada NO tenga histéresis. Muchos pines de entrada tienen histéresis para evitar la generación de flancos en presencia de ruido, así que si el pin en cuestión tiene alguna función como entrada de interrupción, temporizador, o similar, es un buen candidato para tener una histéresis intencionada. Estos pines no deben usarse como entrada del modulador sigma delta.

El modulador sigma delta puede tolerar histéresis en el cuantizador siempre que dicho ciclo de histéresis no sea muy ancho. Las simulaciones muestran que para tensiones de alimentación de 5 voltios la máxima histéresis tolerable es de

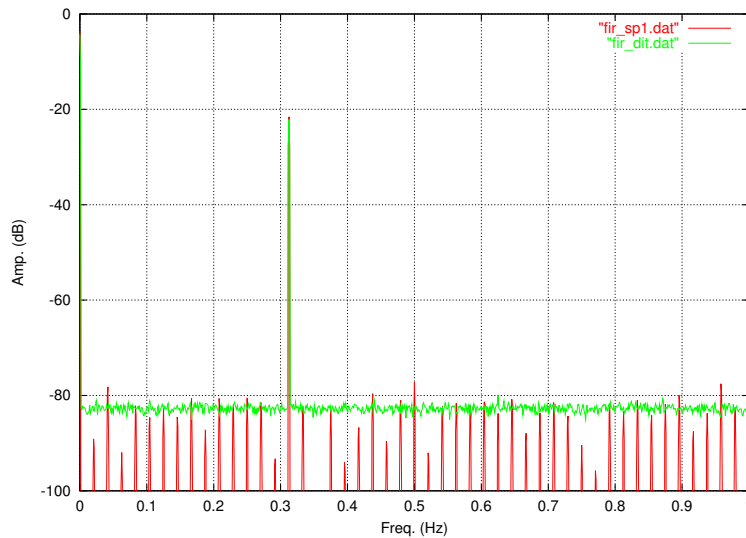


Figura 5: Espectro de salida del convertidor sin y con dithering.

alrededor de 0.1 voltios. Para ciclos de histéresis más anchos el circuito deja de comportarse como un modulador sigma delta, se parece más a un vulgar oscilador de relajación, y la relación señal-ruido se deteriora rápidamente.

3.3. Tonos. Ruido en la alimentación.

Ya hemos comentado antes que el ruido de cuantización no es propiamente “ruido” ya que está correlacionado con la tensión de entrada del cuantizador. A pesar de ello hemos analizado la respuesta en frecuencia y obtenido algunas conclusiones. El efecto de la correlación del ruido de cuantización con la señal de entrada es la aparición de tonos en la respuesta espectral, que son especialmente intensos cuando la señal de la entrada es de poca amplitud o de frecuencia muy baja [6].

Para solucionar este problema los convertidores sigma delta comerciales utilizan la técnica del “dithering”. Esta técnica básicamente consiste en añadir ruido aleatorio a la entrada del cuantizador de modo que la correlación del ruido de cuantización quede enmascarada y desaparezcan o se atenúen los tonos. El ruido de “dithering” tiene la misma función de transferencia que el de cuantización, y es rechazado en la banda de interés, aunque siempre supone la pérdida de algunos dB de SNR.

La figura 5 muestra el espectro simulado de la salida del convertidor sigma-delta sin y con dithering. En el primer caso se aprecian numerosos tonos, aún a pesar que la entrada tiene bastante amplitud. Si a la entrada del cuantizador se superpone una señal aleatoria de 10 mV de amplitud se eliminan los tonos, aunque si se integra el fondo de ruido se tiene una peor SNR. En la mayoría de los casos resulta mejor que todo lo que no sea nuestra señal sea ruido aleatorio en lugar de una colección de tonos, así que intentaremos encontrar alguna forma de incluir dithering en nuestro convertidor.

Una forma poco ortodoxa de conseguir este dithering es aprovechando el ruido que los circuitos digitales provovan en la alimentación. La tensión umbral del pin de entrada es proporcional a la tensión de alimentación, y si la alimentación tiene ruido la variación de V_{TH} es equivalente a la superposición de la deseada señal de dithering. Es importante que el ruido de la alimentación no esté muy correlacionado con la señal del modulador. Para ello, en lugar de un simple condensador de desacoplo, se puede utilizar una red RC (o mejor LC) con una frecuencia de corte bastante superior a los 256 Hz. Se necesitará alguna experimentación hasta conseguir una amplitud de ruido en la alimentación del orden de los 20mV pico a pico.

Notese que la inclusión del ruido de la alimentación tiene un efecto sobre la elección de la constante de tiempo de la red RC. La figura 2 puede hacernos creer que una frecuencia de corte muy baja aproximará más la función NTF de nuestro modulador a la de uno con un integrador. Pero esto también hace que se atenúe la señal a la entrada del cuantizador y que la señal de dithering tenga un peso excesivo reduciendose notablemente el SNR. La figura 6 muestra los resultados de una simulación en la que se ha variado el valor del condensador para varias amplitudes de ruido de dithering para una señal entrada constante. Esta gráfica muestra de forma clara que un exceso de ruido en la entrada del cuantizador reduce la resolución del convertidor. Dado que el fondo de escala del convertidor es ± 32767 la máxima dispersión tolerable si

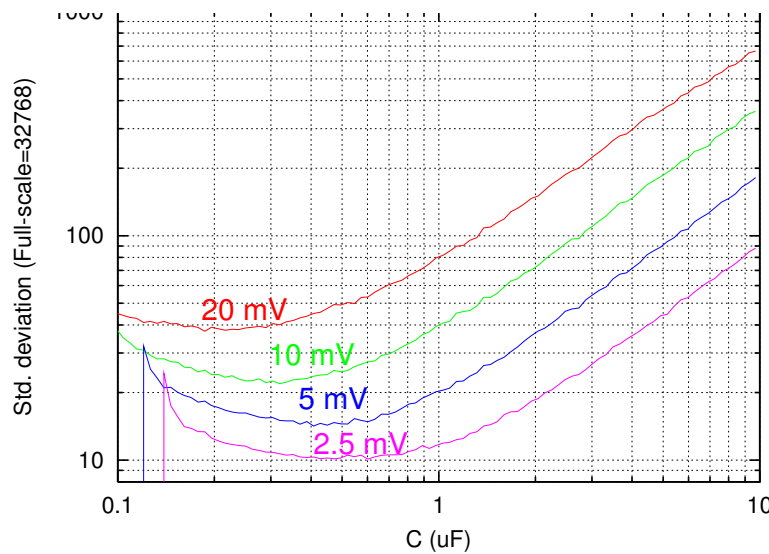


Figura 6: Optimización del valor del condensador de la red RC.

queremos 10 bits de resolución es de 32. Como se puede ver el valor óptimo del condensador depende de la amplitud del ruido de dithering, aumentando al disminuir el ruido. Si suponemos una amplitud entre 5 y 10 mV (el doble pico a pico) tenemos un condensador óptimo en torno a 1/3 pF. Las resistencias han sido: $R_1 = 2,56M\Omega$, $R_2 = 1M\Omega$, $R_{pull} = 5,6M\Omega$. Con un ruido de dithering de sólo 2.5mV se pueden alcanzar casi los 12 bits de resolución (11.6 bits).

3.4. Jitter.

Para el correcto funcionamiento del convertidor es importante que el cuantizador muestree su entrada a intervalos regulares de tiempo. La incertidumbre en el instante de muestreo se traduce en un deterioro del SNR. Si este muestreo se realiza mediante una rutina de interrupción se deben tomar algunas precauciones para garantizar que el muestreo se realiza en instantes precisos:

- No deshabilitar nunca las interrupciones.
- Evitar instrucciones que provoquen latencias largas, como multiplicación, manejo de cadenas, etc.
- Si no fuese posible garantizar una latencia de interrupción mínima se puede intentar compensar la incertidumbre consultando algún temporizador hardware del microcontrolador y ejecutando las instrucciones NOP necesarias para equalizar los retardos.

En el caso del microcontrolador PIC16F84 la incertidumbre que tenemos en la latencia de interrupción es de sólo $1\mu s$ ya que las instrucciones sólo tienen dos tiempos de ejecución posibles: $1\mu s$ para la mayoría de las instrucciones o $2\mu s$ para los saltos. La simulación muestra que un error de 1/4096 (suponiendo una frecuencia de interrupción de 244 Hz) tiene un efecto despreciable en el SNR. Por lo tanto en este microcontrolador bastará con garantizar que las interrupciones están siempre habilitadas.

4. Filtrado y diezmado [5].

La salida del modulador es una secuencia de bits donde un 1 representa una amplitud V_{INMAX} mientras que un 0 representa una amplitud V_{INmin} . Esta señal contiene la información relativa a señal de entrada en las frecuencias bajas, mientras que las frecuencias altas contienen el ruido de cuantización y dithering. Para regenerar la señal original habrá que usar un filtro digital que elimine las frecuencias altas. El filtro deberá tener una banda pasante de 0 a 1 Hz, o algo menos para evitar aliasing. La señal filtrada todavía tendría una frecuencia de muestreo demasiado alta, ya que si su ancho de banda es de 1 Hz basta con una frecuencia de muestreo de 2 Hz (teorema de Shannon). Reduciremos la frecuencia de muestreo quedándonos con una muestra de cada 128. Esta operación se llama diezmado.

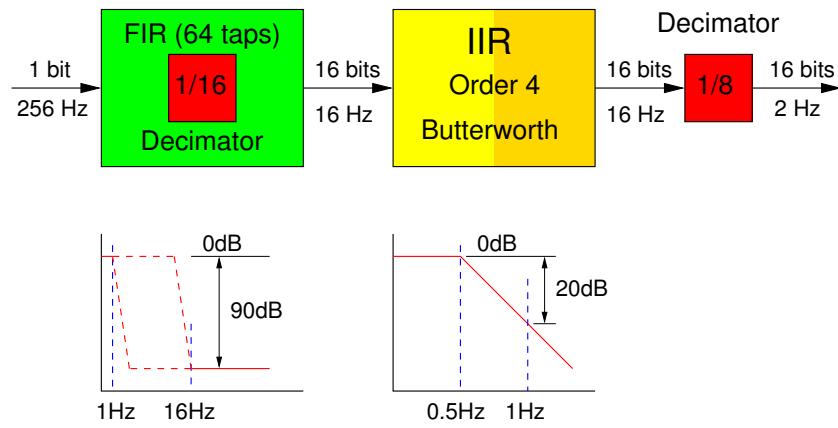


Figura 7: Esquema general del filtro digital y los requisitos de cada bloque.

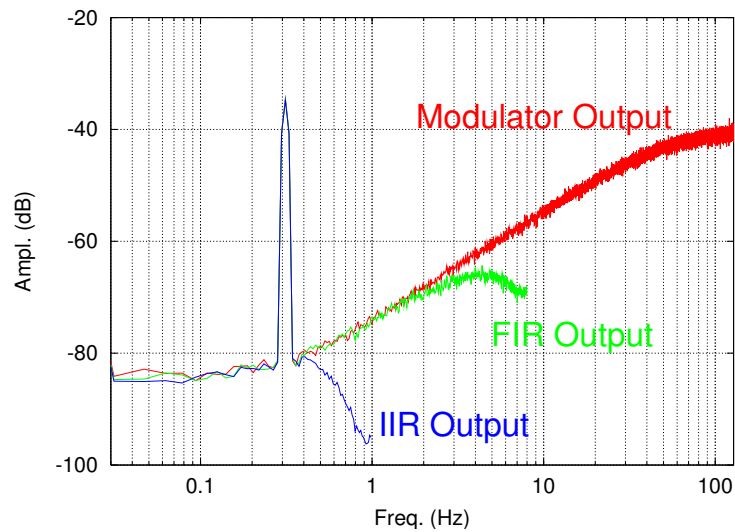


Figura 8: Espectro de la señal de salida en el modulador, en el filtro FIR y en el filtro IIR (salida del convertidor).

Para no realizar demasiadas operaciones aritméticas es una buena idea diezmar lo antes posible la señal para que los filtros que necesiten mayor potencia de cálculo operen sobre señales de baja frecuencia de muestreo. Basándonos en esta idea se ha diseñado un bloque de filtrado y diezmado en dos etapas cuyo esquema se muestra en la figura 7.

En un primer paso usamos un filtro de tipo FIR que se adapta muy bien al diezmado, ya que sólo hay que calcular la convolución del filtro a la frecuencia de muestreo de la salida. Esto es posible porque este tipo de filtros no tiene realimentación. El filtro debe atenuar suficientemente las frecuencias por encima de 15 Hz para que al reducir la frecuencia de muestreo a 16 Hz no nos aparezca el ruido de las frecuencias altas en 0 Hz debido al aliasing. Por otro lado, en las frecuencias bajas el ruido es ya de unos -70dB, por lo que no resulta grave que las componentes alias sean de -90dB. Estos requisitos se cumplen con un filtro FIR de 64 coeficientes. Hay que destacar que la señal de entrada del filtro FIR es de sólo 1 bit, por lo que las multiplicaciones de la convolución se reducen a la suma o resta de coeficientes.

Tras el filtro FIR los datos son de 16 bits y los filtros de tipo FIR requieren muchas multiplicaciones. Si hay que multiplicar que sea poco. Por ello en la segunda etapa utilizaremos un filtro de tipo IIR que es mucho más eficiente desde el punto de vista computacional. El filtro IIR debe operar a la frecuencia de la señal de la entrada ya que la realimentación impide realizar el diezmado de forma simultánea al filtrado. El diezmado final consistirá en quedarnos con una muestra de cada 8 que salgan del filtro IIR.

La figura 8 muestra el espectro de las señales en las salidas del modulador y ambos filtros. Puede parecer que el filtro FIR no mejora la señal, pero sí que lo hace al reducir la frecuencia de muestreo de 256 a 16 Hz sin que todo el ruido de alta frecuencia se “pliegue” hacia las frecuencias bajas por culpa del aliasing. Podemos decir que el principal cometido del filtro FIR es el de evitar que el filtro IIR tenga que operar a la máxima frecuencia. El filtro IIR limita el ancho de banda de la señal a su valor final: 1Hz, permitiendo una frecuencia de muestreo de 2 Hz.

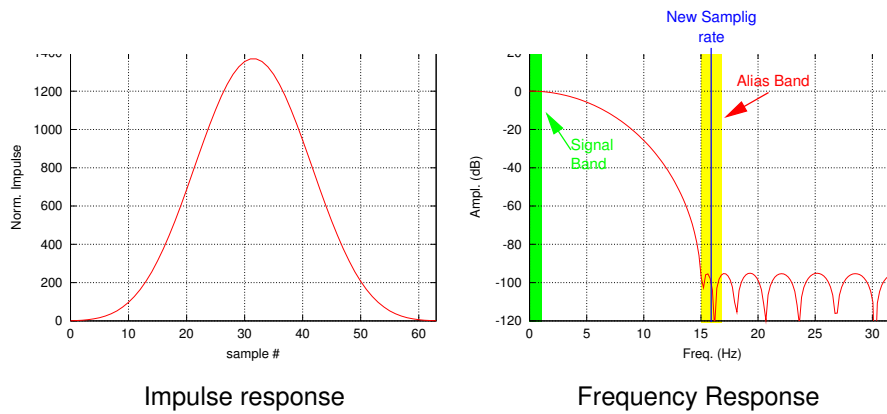


Figura 9: Respuesta del filtro FIR: (a) Respuesta impulsional. (b) Respuesta en frecuencia.

4.1. Filtro FIR

La figura 9 muestra la respuesta impulsional del filtro FIR y su respuesta en frecuencia. La primera gráfica es precisamente la lista de coeficientes del filtro, que consta de 64 elementos. Los coeficientes se han multiplicado por 2^{15} para poder implementar este filtro usando aritmética entera. La suma de todos los coeficientes da muy próxima a 32768 sin llegar a superar este número para evitar desbordamientos aritméticos. La respuesta en frecuencia muestra las características deseadas: Una buena atenuación de las bandas que por aliasing caerían en 0 Hz al reducir la frecuencia de muestreo.

La salida del filtro debe evaluarse cada 16 muestras. Esto puede hacerse de forma eficiente sin utilizar mucha memoria mediante cuatro acumuladores en los que se suman o restan los coeficientes del filtro cada muestra. Cuando en un acumulador ya se tienen 64 muestras su valor se pasa a la salida y el acumulador se pone a cero. De este modo sólo necesitamos 4 variables de 16 bits y un puntero de 8 bits para realizar el filtrado. El siguiente código es un ejemplo de implementación en lenguaje C:

```
// FIR filter (64 taps) & 1/16 decimator
short firout;

void firf(int bit) {
    const short fir[]={
#include "fir.coef"
    };
    static short acc[4]={0,0,0,0};
    static char ix=0;
    int i,j;

    j=ix;
    for (i=0;i<4;i++) {
        if (bit) acc[i]+=fir[j];
        else acc[i]-=fir[j];
        if (j==63) {
            firout=acc[i];
            acc[i]=0;
        }
        j=(j+16)&63;
    }
    ix=(ix+1)&63;
}
}
```

Hay que destacar la simetría de la tabla de coeficientes que permitiría almacenarla como sólo 32 constantes en lugar de 64. En el anterior ejemplo esto no se ha echo para no dificultar la lectura del código.

4.2. Filtro IIR

La figura 10 muestra la forma en la que se ha implementado el filtro IIR. Este esquema obedece a las recomendaciones de la referencia [5]. El filtro es de tipo Butterworth de 4^o orden, pero se ha dividido en dos secciones de 2^o orden para reducir la sensibilidad de los coeficientes. Se ha utilizado aritmética de coma fija que requiere un mínimo de 24 bits, ya que se han reservado 8 bits para la parte decimal en las multiplicaciones. Se precisará de aritmética de 24 y 32 bits. El siguiente es un ejemplo de implementación en lenguaje C:

```
// IIR filter
short iirout;

void iirf(short x)
{
    const int A1=425,B1=178;
    const int A2=467,B2=220;
    static int dl[4]={0,0,0,0};
    int y1,y;

    y1=x+dl[0];

    dl[0]=x+x+((A1*y1)>>8)+dl[1];
    dl[1]=x-((B1*y1)>>8);

    y1>>=7; //Gain~ 1/128

    y=y1+dl[2];

    dl[2]=y1+y1+((A2*y)>>8)+dl[3];
    dl[3]=y1-((B2*y)>>8);

    y>>=7; //Gain~ 1/128
    iirout=(y*305)>>8; //Gain adjust x 1.193
}
```

En este ejemplo se ha cuidado la distribución de la ganancia para evitar alcanzar los límites de la aritmética en cada sección del filtro. En ningún nodo de la figura 10 se alcanzan valores numéricos superiores a 4000000, lo que permite el uso de variables y aritmética de 24 bits sin temor a los overflows, ya que el rango de estas variables es de ± 8388607 . Nótese que el ajuste final de la ganancia es pequeño y podría hacerse de forma externa controlando el ratio $R1/R2$.

4.3. Requisitos computacionales.

El número total de operaciones que hay que realizar cada segundo será:

- Filtro FIR: 4 sumas de 16 bits $\times 256$ Hz
- Filtro IIR: (8 sumas de 24 bits + 4 multiplicaciones de $24 \times 9 \Rightarrow 32$ bits+) $\times 8$ Hz

Las multiplicaciones habrá que realizarlas mediante sumas si el microcontrolador en cuestión carece de instrucciones de multiplicar. En ese caso cada multiplicación equivale a 9 sumas de 32 bits y 9 desplazamientos de 32 bits. Si además consideramos que el ancho de palabra para los microcontroladores usuales es de 8 bits habrá que descomponer las sumas de 16, 24 y 32 bits en 2, 3 o 4 sumas de 8 bits, y lo mismo ocurre con los desplazamientos. Esto nos da un total de:

- Filtro FIR: 2048 sumas de 8 bits.
- Filtro IIR: 1344 sumas de 8 bits, 1152 desplazamientos de 8 bits.

IIR Filter Topology

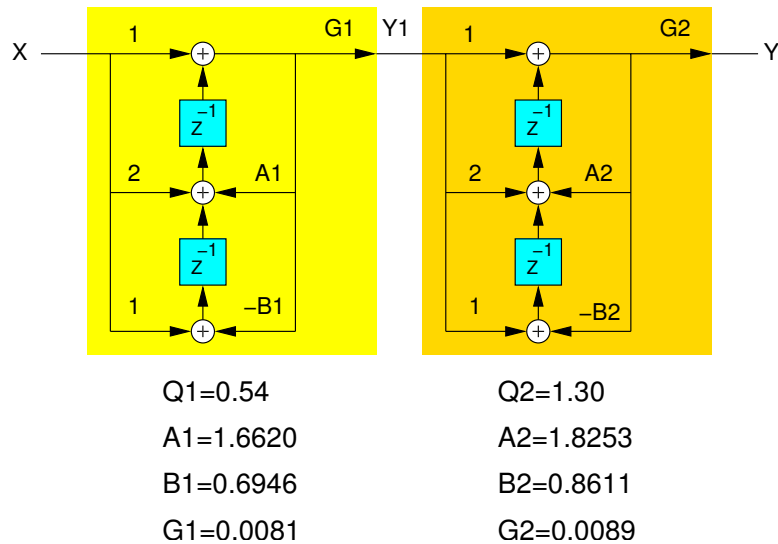


Figura 10: Diagrama de flujo del filtro IIR.

- Total:4544 operaciones de 8 bits

En microcontroladores con arquitecturas orientadas al registro acumulador una suma o desplazamiento supone mover un registro al acumulador, operar en el acumulador y escribir el resultado en el destino. Esto supone unos $3 \mu s$ en un microcontrolador de 1 MIPS, lo que nos da un tiempo de cálculo de 13.6 ms cada segundo o lo que es lo mismo: una carga de CPU mínima en torno al 1.4%. Esto nos lleva a pensar que si dedicamos todo el tiempo de CPU a la conversión A/D podríamos obtener unas 140 muestras de 10 bits cada segundo. La frecuencia de muestreo básicamente está limitada por la aritmética de la CPU.

4.4. Aún más sencillo: Filtro tipo peine (Comb)

La figura 9(b) nos muestra que lo realmente importante del filtro FIR es que las frecuencias próximas a 16 Hz, 32 Hz,... queden suficientemente atenuadas para que el ruido del modulador no aparezca en frecuencias inferiores a 1 Hz debido al aliasing que conlleva el diezmado. Tal vez el uso del filtro FIR sea un tanto excesivo, ya este objetivo puede lograrse con un filtro mucho más sencillo: Un filtro de tipo PEINE (Comb filter).

Los filtros tipo peine se caracterizan por tener una serie de ceros (notch) en frecuencias equiespaciadas. Si hacemos coincidir los ceros del filtro con la nueva frecuencia de muestreo y sus armónicos tenemos resuelto el problema del diezmado.

El filtro peine más sencillo que podemos imaginar es la media móvil. Esto es: la suma de las últimas M muestras. Este filtro presenta ceros en la frecuencia $\frac{f_s}{M}$ y sus múltiplos. Por lo tanto debemos hacer M igual al factor de diezmado. En nuestro caso $M = 16$.

La media móvil es un tipo concreto de filtro FIR en el que todos los 16 coeficientes son uno. Su función de transferencia es por lo tanto:

$$y_i = x_i + x_{i-1} + \dots + x_{i-15}; \quad \frac{Y}{X} = 1 + z^{-1} + z^{-2} + \dots + z^{-15}$$

Una implementación literal nos obligaría a calcular 15 sumas por cada muestra de entrada. Sin embargo podemos usar una estrategia distinta: En una sólo variable sumamos la entrada actual y restamos la entrada que tuvimos 16 muestras antes. La función de transferencia será:

$$y_i = x_i - x_{i-16} + y_{i-1}; \quad \frac{Y}{X} = \frac{1 - z^{-16}}{1 - z^{-1}}$$

Como podemos comprobar esta nueva implementación se corresponde con un filtro de tipo IIR, y sin embargo la

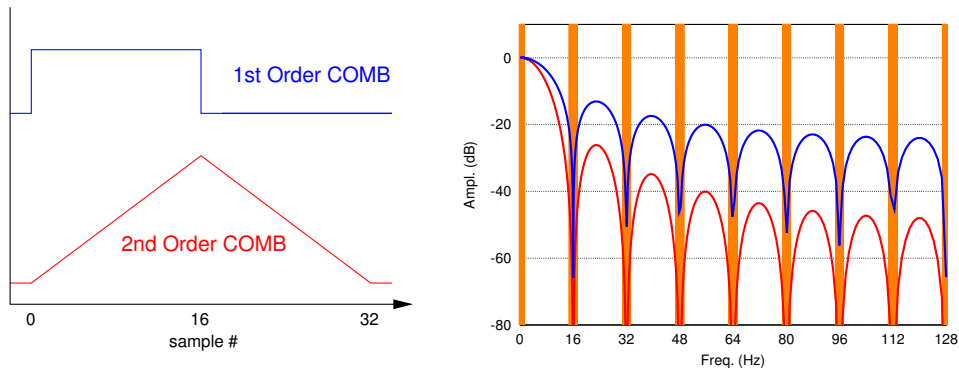


Figura 11: (a) Respuesta impulsional y (b) respuesta en frecuencia de los filtros Peine de primer y segundo orden.

respuesta es la misma. La implementación FIR tiene una función de transferencia con 15 ceros distribuidos alrededor del círculo unidad, mientras que la implementación IIR tiene 16 ceros y un polo, pero el cero extra y el polo están en la misma posición ($z = 1$) y se cancelan.

La media móvil o filtro peine de orden 1 puede que no sea suficiente para atenuar las bandas de aliasing de nuestro modulador. Se pueden obtener filtros peine de mayor orden simplemente conectando en cascada varios filtros de primer orden. Como regla práctica, si el orden del modulador es L , el orden del filtro debe ser $L+1$ o mayor. En nuestro caso bastará con un filtro de orden 2 cuya función de transferencia será:

$$\frac{Y}{X} = \left(\frac{1 - z^{-16}}{1 - z^{-1}} \right)^2$$

En la figura 11(a) se muestra la respuesta impulsional del filtro peine de orden 1 (media móvil) y de orden 2. La respuesta impulsional del filtro de orden 2 dura el doble de tiempo y tiene forma triangular. El filtro peine de orden 2 es por lo tanto una media móvil con ponderación triangular. El filtro de orden 3 sería una media con ponderación parabólica, y así sucesivamente. La figura 11(b) también muestra la respuesta en frecuencia de dichos filtros. Como se puede comprobar el filtro de orden 2 proporciona mayor atenuación tanto en los ceros como fuera de ellos. La curva envolvente decrece a razón de 40 dB/dec. en lugar de los 20 dB/dec del filtro de orden uno. En las bandas de aliasing la atenuación es mayor de 40 dB.

Para implementar el filtro peine de orden 2 dividiremos la función de transferencia en dos partes:

$$\frac{Y}{X} = \frac{1}{1 - z^{-1}} \cdot \frac{1}{1 - z^{-1}} \times (1 - z^{-16})(1 - z^{-16})$$

La primera parte la componen dos integradores del tipo: $y_i = x_i + y_{i-1}$, tremendamente sencillos de implementar por programa. Respecto al numerador tendremos en cuenta que la operación de diezmar una señal por $\frac{1}{M}$ se traduce en las funciones de transferencia en el cambio de z^{-M} por z^{-1} . Por lo tanto realizaremos el diezmo antes de aplicar la segunda parte de la función de transferencia del filtro peine, quedándonos:

$$\frac{Y}{X} = \frac{1}{1 - z^{-1}} \cdot \frac{1}{1 - z^{-1}} \{ \text{diezmado } 1/16 \} (1 - z^{-1})(1 - z^{-1})$$

La segunda parte está compuesta por dos derivadores ($y_i = x_i - x_{i-1}$) que operan a $1/16$ de la frecuencia de muestreo original.

Hay que destacar que ante entradas cuyo valor medio sea distinto de cero los integradores empezarán a aumentar su valor hasta que tarde o temprano se tenga un desbordamiento aritmético. El filtro es por lo tanto inestable, pero esto no supone un problema si tenemos en cuenta el carácter circular de la aritmética entera. Aunque los acumuladores se desborden las diferencias entre muestras siguen siendo correctas si disponemos de suficientes bits. Las variables deberán tener al menos $K + L \log_2 M$ bits, donde K es el número de bits de la entrada, en nuestro caso $K = 1$, L es el orden del filtro ($L = 2$) y M es el factor de diezmo ($M = 16$). En nuestro filtro necesitaremos al menos 9 bits.

A continuación tenemos un ejemplo de implementación del filtro en lenguaje C. Tenemos dos funciones, una para la parte de integración, que se ejecuta por cada muestra del modulador, y otra para el derivador que se ejecuta cada 16

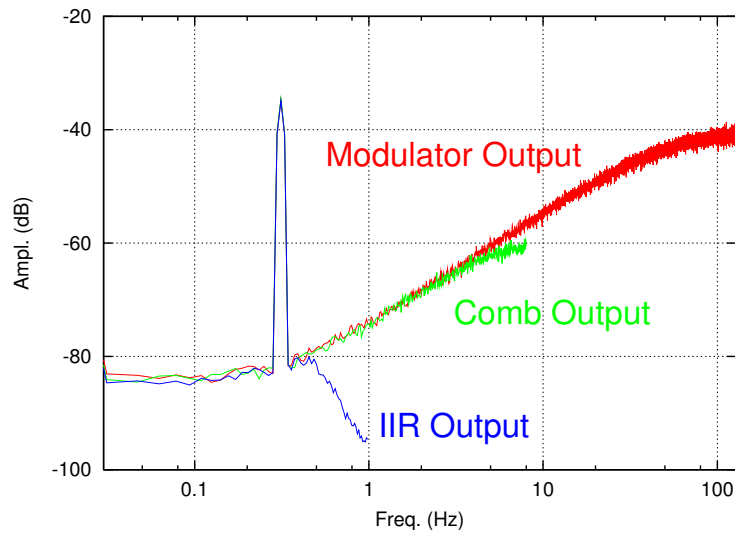


Figura 12: Espectros de las señales a la salida del modulador, del filtro Peine, y del filtro IIR (salida del convertidor).

muestras:

```
// COMB filter (2nd order)
short combiout,combout;

void combint(int bit) {
    static short comb1;

    combiout+=comb1;
    if (bit) comb1++; else comb1--;
}

void combdif() {
    static short comb1,comb2;
    short o;

    o=combiout-comb1;  comb1=combiout;
    combout=o-comb2;  comb2=o;
}
```

El filtro peine requiere menos operaciones que el filtro FIR (1088 sumas de 8 bits por segundo frente a 2048) y menos memoria, ya que no tenemos que almacenar su respuesta impulsional. En la figura 12 se muestran los espectros de las señales del convertidor en el que se ha sustituido el filtro FIR por el filtro peine. Los resultados son prácticamente idénticos en la banda de la señal.

5. Ampliaciones.

5.1. Múltiples convertidores.

La extensión de este diseño a un número mayor de entradas analógicas es directa. Se necesitarán 2 pines de E/S, o 3 si se quiere calibración, por cada convertidor. Los requisitos de cálculo se multiplican por el número de entradas. En cuanto a la memoria, se multiplican las variables de los filtros, pero no las constantes, que son las mismas para todos. Esto es especialmente importante en el caso de los filtros FIR.

Otra posible solución es el uso de un multiplexor analógico externo, que podría incluir entre sus entradas las tensiones de calibración. En este caso hay que tener presente que cuando se conmuta el multiplexor se tardarán varias muestras

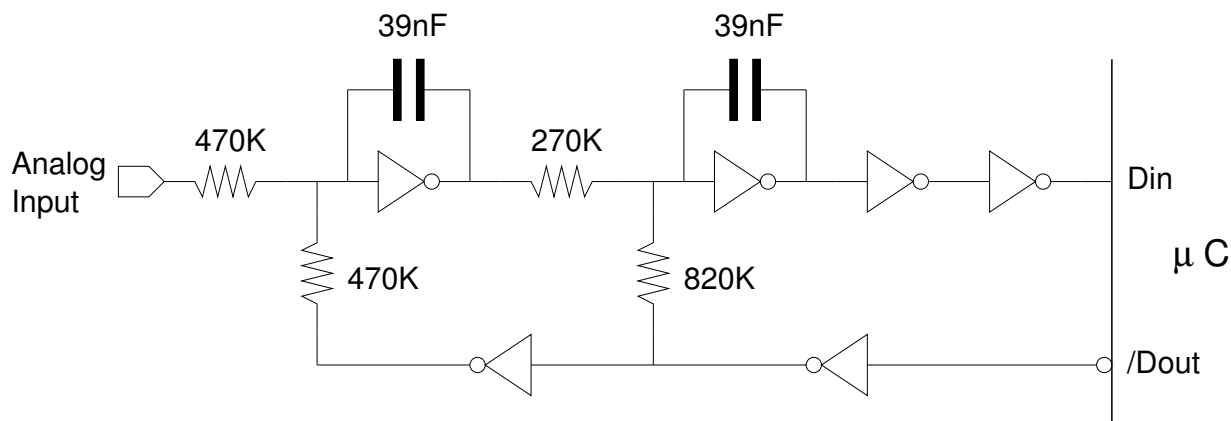


Figura 13: Modulador de segundo orden realizado con inversores CMOS.

en tener una lectura correcta debido a la constante de tiempo del filtro IIR. Con una frecuencia de corte en 0.5 Hz el transitorio de conmutación tardará unos 8 segundos en atenuarse hasta 1/1000 de su valor inicial. Por ello seguramente sería conveniente aumentar la frecuencia de muestreo.

5.2. Mayor resolución.

Para aumentar la resolución del convertidor se pueden tomar varias medidas:

- Aumentar el factor de sobremuestreo, OSR. Esto supone aumentar la frecuencia de las interrupciones o disminuir el ancho de banda de la señal. En cualquier caso aumentar el OSR nos sirve de poco si se mantiene un nivel de ruido constante en la banda de la señal. Para solucionar este problema habrá que sustituir la red RC por un integrador externo con un amplificador operacional.
- Utilizar un modulador de 2º orden. Si hemos de poner amplificadores operacionales en el circuito podemos implementar un modulador de segundo orden que incorpora dos integradores en su función de transferencia $H(s)$. Consultar las referencias [6, 4] para más información. Este tipo de modulador supone una mejora muy sustancial del SNR, lo que nos permitirá obtener más bits de resolución o en su defecto reducir el OSR consiguiendo un mayor ancho de banda para la señal. A continuación, en la sección 6, se muestra un ejemplo de modulador de segundo orden realizado con unos pocos inversores CMOS.

6. Modulador de segundo orden.

En el mundo de los convertidores sigma-delta rara vez se emplean en la práctica los moduladores de primer orden. Estos moduladores no proporcionan más de 20 dB/dec. de atenuación del ruido de cuantización para frecuencias bajas y tienen la mala tendencia de generar tonos. Los moduladores de orden más alto presentan unas funciones NTF con atenuaciones mucho mayores para el ruido de cuantización, y los tonos no son un problema muy destacado. Por el contrario, los moduladores de órdenes altos pueden ser inestables.

Los moduladores de orden 2 son por así decir el modelo “estandar” de modulador sigma delta. Son notablemente mejores que los de primer orden, todavía con una muy buena estabilidad, y sencillos de fabricar. Estos moduladores proporcionan una NTF con una atenuación del ruido de cuantización que decrece a razón de 40 dB/dec y se implementan con sólo dos integradores.

Hay dos alternativas a la hora de diseñar moduladores sigma-delta. La primera de ellas utiliza circuitos de tiempo discreto en sus filtros. Estos circuitos suelen ser de condensadores conmutados y, aunque son fáciles de integrar y muy precisos, no se adaptan en absoluto a nuestros propósitos. La segunda alternativa emplea filtros de tiempo continuo y tiene su propia problemática [3, 2, 1]. Este será el tipo de modulador que analizaremos a continuación.

El circuito de la figura 13 es una realización de un modulador de orden dos que utiliza inversores CMOS como integradores. Estos 6 inversores pueden ser del integrado 4069 y conviene que al menos los cuatro de la rama de entrada

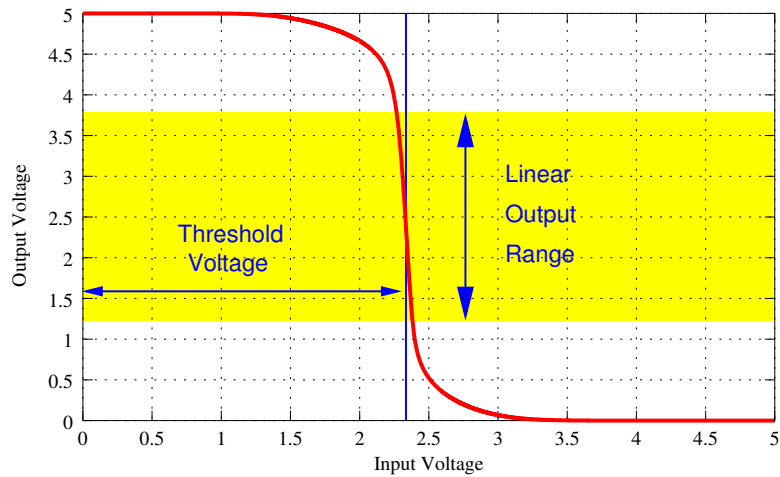


Figura 14: Función de transferencia de un inversor CMOS y su región de funcionamiento lineal.

estén físicamente en el mismo circuito integrado para garantizar que tengan aproximadamente la misma tensión umbral. Los dos primeros inversores realizan las funciones de un amplificador operacional en configuración de integrador, y los dos siguientes se comportan como un comparador que devuelve un cero o un uno dependiendo de si la tensión de la entrada supera o no el umbral de los inversores. La realimentación se hace a través de los dos inversores de la rama inferior. Hay realimentación desde cada integrador y es negativa en ambos casos.

Puede pensarse que un inversor CMOS no llega ni de lejos a ofrecer las características de un amplificador operacional, y no falta razón. Sin embargo, los moduladores sigma-delta son muy tolerantes con las no idealidades de los integradores, y el uso de amplificadores operacionales no estaría justificado. En primer lugar veamos como un circuito, a priori de uso digital, puede suplantantar a otro tan analógico como un amplificador operacional.

En la figura 14 se muestra la característica de transferencia de un inversor CMOS como el 4069. Se puede ver que para tensiones de entrada bajas la salida está con tensiones altas, y viceversa, tal como debe ser en un inversor. Pero para tensiones muy próximas a 2.5 Voltios la tensión de salida decrece de forma lineal con la tensión de entrada. Esta zona de la característica es la adecuada para utilizar el inversor como amplificador. La pendiente de la gráfica es la ganancia del amplificador y en el caso simulado ronda los -40, que, como se puede ver, no es muy elevada (un operacional como el 741 puede tener ganancias de más de 100000) pero para un modulador sigma delta basta que la ganancia en DC de los integradores sea del orden del factor de sobremuestreo.

Uno de los problemas que plantean los inversores es la ausencia de una entrada de tensión de referencia. En el caso ideal la tensión de referencia interna del inversor es de 2.5 V, esto es: la tensión a la que la característica de transferencia baja de forma abrupta. Pero en un inversor real esta tensión puede variar bastante de un integrado a otro o con la temperatura, de modo que seguimos con el problema de no tener un origen bien calibrado para el convertidor. Por el contrario, un amplificador operacional tiene dos entradas y una de ellas se usa como referencia. En este caso el error del origen es de sólo unos milivoltios (el offset de entrada del operacional)

Otro problema de los inversores CMOS es que generan bastante ruido de baja frecuencia. No son por lo tanto nada recomendables para preamplificadores de audio, por poner un ejemplo. Este ruido también pudiera ser un factor limitador si se buscasen resoluciones grandes en el convertidor sigma delta.

Volviendo al modulador de la figura 13, los condensadores y las resistencias de este circuito han sido calculados de acuerdo con las fórmulas de la referencia [2] para una frecuencia de muestreo de 256 Hz, y deben escalarse si esta frecuencia fuese distinta. Así, para una frecuencia de muestreo de 2560 Hz usaremos unos condensadores 10 veces más pequeños o en su lugar unas resistencias 10 veces más pequeñas. La resistencia de 470KΩ de la entrada se puede escalar para adaptar diferentes rangos de entrada. En el circuito de la figura 13 el rango de entrada va desde 0 hasta 5 Voltios, pero es una mala idea convertir amplitudes de más de un 70 % en un modulador como este, ya que se vuelve muy ruidoso con entradas grandes. Para solucionar este problema se puede incrementar la resistencia de entrada en un 43 %, de modo que en ningún caso tendremos entradas excesivamente grandes en el convertidor.

Los resultados de la simulación del modulador de segundo orden de la figura 13 se muestran en la figura 15. En la simulación se ha incluido la característica de transferencia de los inversores (figura 14) con lo que los efectos de su ganancia

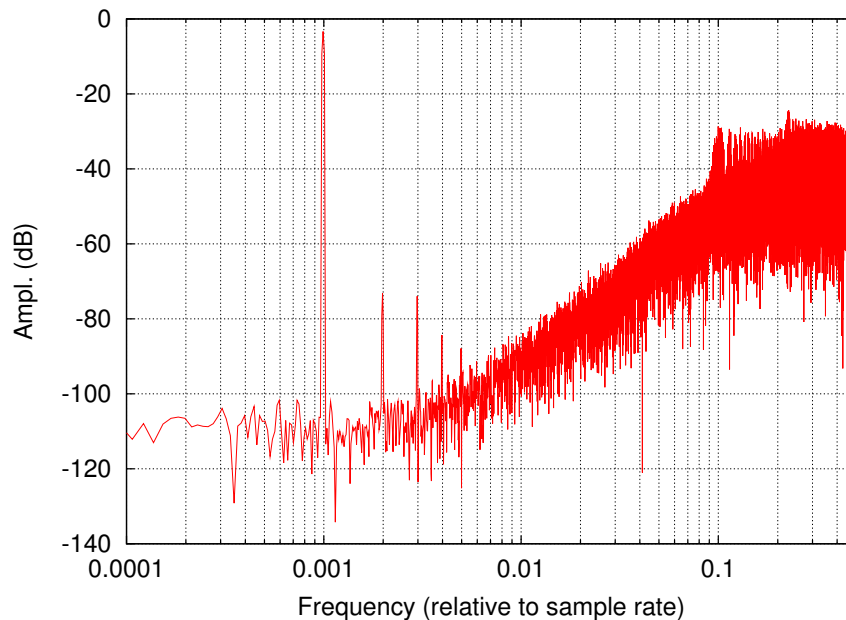


Figura 15: Simulación del modulador sigma-delta de segundo orden.

limitada y no linealidades se pueden apreciar en la gráfica. En primer lugar vemos que el fondo de ruido desciende con la frecuencia a razón de 40 dB/dec, pero a frecuencias bajas se estabiliza en torno a -110 dB/caja_FFT. Este efecto se debe a la escasa ganancia de los inversores, y hace que la relación señal-ruido sea peor de lo esperado, especialmente para factores de sobremuestreo grandes. Para un factor de sobremuestreo de 64 obtenemos una relación señal-ruido de 68 dB, que es prácticamente idéntica a la de un modulador con integradores ideales.

Por otra parte, la gráfica de la figura 15 también muestra la presencia de armónicos. Estos armónicos aparecen debido a la no-linealidad de los inversores y se pueden considerar como un error del convertidor, de manera similar a como hacemos con el ruido. Teniendo en cuenta la amplitud de los armónicos el rango dinámico libre de espúreos (SFDR) del convertidor se reduce a unos 64 dB.

La simulación nos muestra que se pueden obtener 11 bits de resolución con un factor de sobremuestreo de sólo 64 en un modulador que emplea inversores CMOS como amplificadores. En un modulador ideal se obtendrían 13 bits de resolución con un sobremuestreo de 128 y 16 bits con un sobremuestreo de 256. En un convertidor ADC con un modulador de segundo orden ideal se ganan 3 bits de resolución cada vez que se duplica la frecuencia de muestreo. Sin embargo, con el modulador propuesto, la no linealidad del convertidor no sería aceptable para resoluciones de más de 11 bits de modo que habría que sustituir los inversores por otros amplificadores mejores.

También hay que destacar que si el modulador es de orden 2 el filtro de tipo peine, que se realiza por programa en el microcontrolador, debería de ser de orden 3 para atenuar suficientemente el ruido de alta frecuencia.

6.1. Modulador de segundo orden experimental

Hemos realizado una implementación del modulador de segundo orden mencionado anteriormente, aunque en este caso se ha diseñado para funcionar con una frecuencia de reloj de 256KHz. El esquema básico se muestra en la figura 16. La realimentación se obtiene directamente de un flip-flop. No hay ningún microcontrolador procesando los datos. En su lugar, el flujo de bits de salida se agrupan en grupos de 8 bits gracias a un registro de desplazamiento (no mostrado) y se almacenan en una memoria FIFO. Los datos de la FIFO se leen en un ordenador personal a través de su puerto paralelo y allí se procesan para obtener los espectros de salida del modulador.

Los resultados obtenidos se muestran en las figuras 17(a,b). En la figura 17(a) se muestra un espectro de salida en el que se puede apreciar el tono de la señal de entrada junto con alguno de sus armónicos, la pendiente del fondo del ruido de cuantización, que es de +40dB/dec para las frecuencias medias, el fondo plano del ruido de cuantización en las frecuencias bajas que es debido a la baja ganancia de los inversores, y un débil tono en 50Hz que indica que estamos en Europa :)

Suponiendo un factor de sobremuestreo de 64, lo que nos da un ancho de banda para la señal de 2KHz, se obtienen las

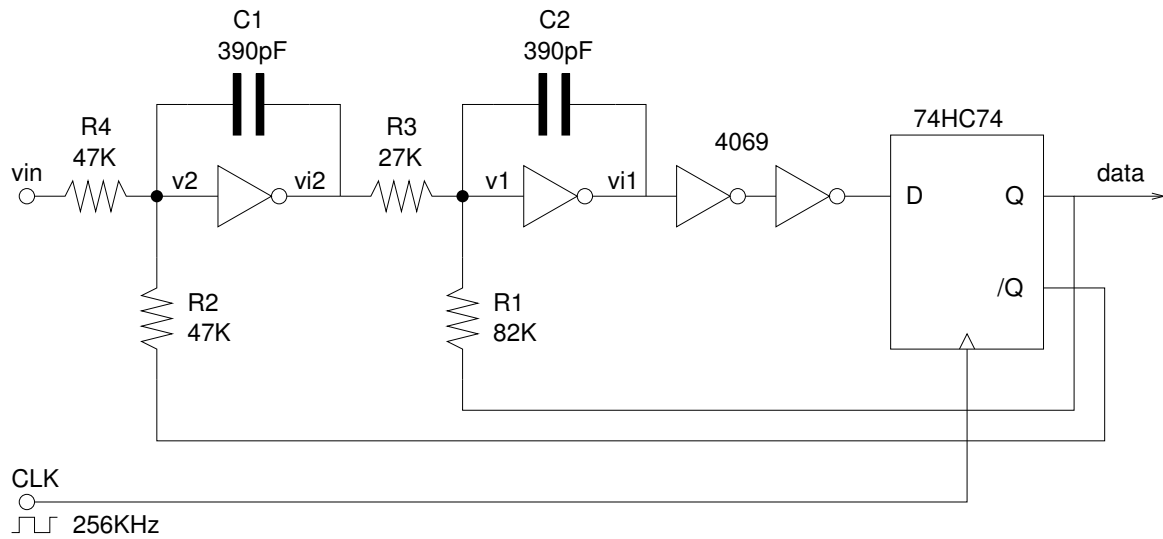


Figura 16: Montaje experimental de un modulador $\Sigma\Delta$ de segundo orden basado en inversores CMOS.

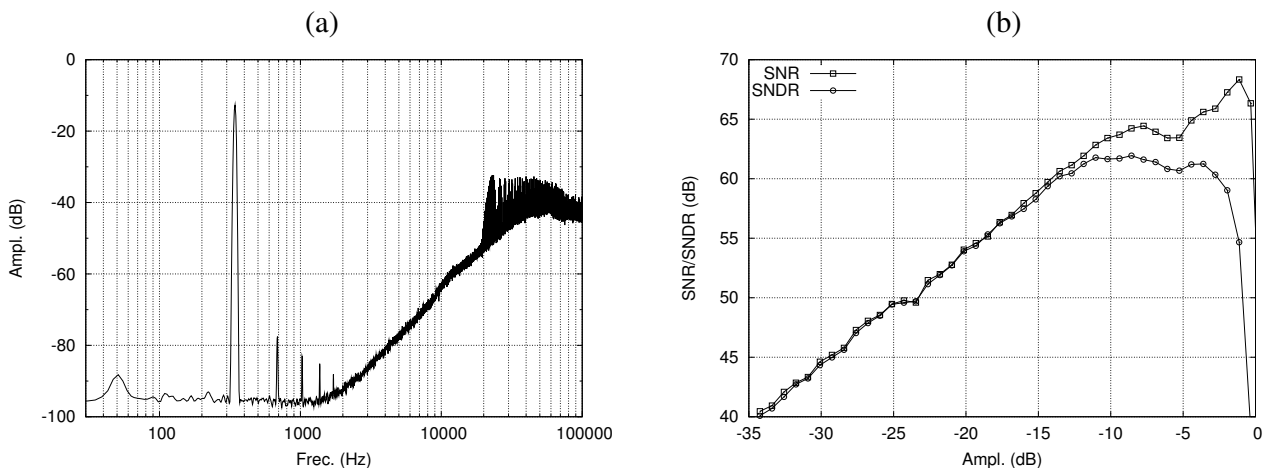


Figura 17: Resultados experimentales del modulador basado en inversores CMOS. (a) Espectro de salida para una entrada sinusoidal. (b) Dependencia de la relación señal-ruido (SNR) y relación señal-ruido+distorsión (SNDR) con la amplitud de la señal de entrada.

curvas de la figura 17(b). Como se puede observar, se superan los 60dB de SNR, lo que equivale una resolución efectiva de más de 10 bits. Para amplitudes de entrada grandes la distorsión domina sobre el ruido de cuantización y es el efecto que limita la resolución efectiva de convertidor ya que si extrapolamos la recta del SNR que tenemos para amplitudes de entrada bajas se obtendrían más de 70dB de SNR para una amplitud de -3dB (máxima amplitud que mantiene un modulador ideal estable).

7. Conclusión.

Los cálculos realizados, junto con numerosas simulaciones, muestran que es posible la implementación de un convertidor A/D $\Sigma\Delta$ usando tan sólo dos pines de entrada-salida digital y una red R-C. Con este convertidor se pueden conseguir resoluciones de 10 bits, y para frecuencias de muestreo bajas el tiempo de cálculo es totalmente asumible, aún a pesar de los escasos medios computacionales disponibles en los microcontroladores de bajo coste. Con un poco más de electrónica, también de muy bajo coste, se puede realizar un modulador de segundo orden, cuyas prestaciones son aún mejores.

Referencias

- [1] Hassan Aboushady. *Design for reuse of current-mode continuous-time sigma-delta analog-to-digital converters*. PhD thesis, University of Paris VI, Dep. Electronics, Communications and Computer Science, January 2002.
- [2] H. Aboushady, E. L. Mendes, M. Dessouky and P. Loumeau. A current-mode continuous-time delta-sigma modulator with delayed return-to-zero feedback. *IEEE ISCAS*, II:360–363, June 1999.
- [3] J. A. Cherry and W. M. Snelgrove. *Continuous-Time Delta-Sigma Modulators for high speed A/D conversions*. Kluwer Academic Publishers, 2000. ISBN: 0-7923-8625-6.
- [4] Peter Kiss. *Single-Loop delta-sigma ADCs Overview (Chapter 2)*. PhD thesis.
- [5] Kurt L. Kosbar. *DSP Pages*. <http://www.siglab.ece.umr.edu/ee301/dsp/>, 1998.
- [6] Norsworthy, Schreier, Temes. *Delta-Sigma Data Converters*. IEEE Press, 1997.