

Trabajo de Cátedra: Uso de la placa Discovery para el cálculo e implementación de filtros FIR e IIR

Paz Martin E¹,
Rodriguez Oscar A², y Galasso Christian L³

¹ Estudiante, Cátedra de Teoría de Circuitos II – Universidad Tecnológica Nacional – Facultad Regional Bahía Blanca, Correo Postal 8000,
Buenos Aires, Argentina
pazmartin35@gmail.com
<http://www.frbb.utn.edu.ar/frbb/index.php>

² Profesor, Cátedra de Teoría de Circuitos II – Universidad Tecnológica Nacional - Facultad Regional Bahía Blanca, Correo Postal 8000,
Buenos Aires, Argentina
arodrig@frbb.utn.edu.ar

³ Ayudante, Cátedra de Teoría de Circuitos II – Universidad Tecnológica Nacional - Facultad Regional Bahía Blanca, Correo Postal 8000,
Buenos Aires, Argentina
christian_galasso81@yahoo.com.ar

Resumen. En el presente trabajo se describe un sistema embebido que permite la implementación de distintos tipos de filtros digitales, calculando sus respectivos coeficientes a partir de configuraciones establecidas previamente por el usuario. Por medio de experimentos, se obtuvieron ecuaciones que permiten establecer en forma aproximada, valores máximos de frecuencia de muestreo para un valor determinado de orden del filtro, o un valor de orden máximo para una frecuencia de muestreo establecida, diferenciando si el filtro a utilizar es FIR o IIR. Este diseño se implementó sobre un microcontrolador con conversor analógico digital a la entrada y conversor digital analógico a la salida, para lo cual se tuvo que adaptar la señal de entrada a parámetros aptos para ser introducida al microcontrolador sin dañarlo.

Palabras claves— filtro digital; FIR; IIR; microcontrolador; sistema embebido.

1 Introducción

Este proyecto se desarrolló en el marco de la cátedra Teoría de Circuitos II de la carrera Ingeniería Electrónica de la Facultad Regional Bahía Blanca de la UTN, como una ayuda didáctica, con el fin de que los usuarios puedan comprender el comportamiento de los filtros digitales, considerando el orden, la frecuencia de muestreo, el ancho de banda máximo y los tipos y características de cada filtro.

El trabajo realizado se dividió en dos partes. En la primera se hizo un firmware que implementa filtros tipo FIR o IIR y donde los coeficientes de los filtros son calcula-

dos externamente por medio de un software de análisis matemático, y exportados al software de programación del microcontrolador. El proceso de exportación es tedioso y complejo, ya que la forma en que el programa matemático y el entorno de desarrollo del microcontrolador manejan los valores numéricos es distinta, y exige una traducción cuando se pasa de uno a otro. La traducción se empezó haciendo de forma manual pero luego se optó por utilizar un tercer programa para la misma.

Para el desarrollo del firmware del microcontrolador que implementa los filtros, se utilizaron, en una primera aproximación, librerías de la firma ST Microelectronics. Las funciones provistas por estas son difíciles de analizar y de visualizar cómo lleva a cabo la operación matemática del filtro. Dado que el proyecto tiene una impronta pedagógica se decidió re-escribirlas utilizando las fórmulas provistas por la cátedra de Teoría de Circuitos II – [1], de forma que el alumno que mira el código pueda rápidamente encontrar relación entre el programa escrito en “C” y las fórmulas vistas en la teoría. Se realizaron pruebas con ambas librerías y se obtuvieron los mismos resultados, si bien es probable que el código brindado por ST Microelectronics esté más optimizado.

En la segunda parte se incorporó al firmware del microcontrolador el cálculo de coeficientes para distintos tipos de filtros (pasa bajo, pasa alto, pasa banda y elimina banda) ya sean de tipo FIR o IIR, evitando de esta forma la necesidad de utilizar el programa de análisis matemático y el de traducción de valores numéricos. A diferencia del primer diseño, los coeficientes calculados solo pueden obtenerse depurando el firmware, en caso de que el usuario desee conocerlos. Y como contra partida los datos del filtro a calcular deben introducirse en el código antes de su compilación. Posibles mejoras en esta parte se proponen en la sección VI trabajo futuro.

2 Desarrollo

El microcontrolador utilizado es el STM32F407 de la firma ST Microelectronics [2], el cual puede trabajar a una velocidad máxima de 168 MHz, posee Conversor Analógico Digital (ADC), Conversor Digital Analógico (DAC), temporizadores programables, hardware de Procesamiento Digital de Señal (DSP), y Unidad de Punto Flotante (FPU), permitiendo mayor velocidad de cálculo. El microcontrolador se encuentra montado en un kit de desarrollo STM32F4DISCOVERY, permitiendo múltiples aplicaciones.

El ADC utilizado trabaja con tensión de 3V y una resolución de 12 bits, es decir que el valor mínimo que se puede leer es de 732 uV. Si bien para señales menores a 1 mVpp, estas pueden ser amplificadas, el mayor límite es que la tensión máxima es baja y debe ser de valor positivo. Teniendo en cuenta estas condiciones, el generador utilizado se configuró con una señal senoidal de 1 Vpp y 750 mV de offset.

EL DAC, al igual que el ADC, es de 12 bits y 3V de salida máxima.

El temporizador utiliza como frecuencia de referencia 84 MHz, siempre que el oscilador interno del microcontrolador este configurado para trabajar a 168 MHz, y se configura utilizando dos variables de 16 bits denominadas periodo y prescaler [3]. Su

función es interrumpir a la frecuencia de muestreo establecida y habilitar una bandera que permita el muestreo de la señal

3 Funcionamiento

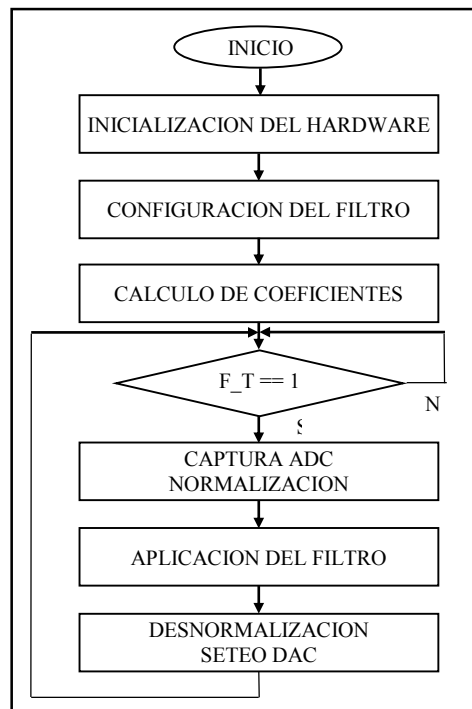


Figura 1 – Diagrama de funcionamiento del microcontrolador.

En la figura 1 se muestra el diagrama general de funcionamiento del microcontrolador, compuesto por tres zonas principales: *inicialización del hardware*, *configuración del filtro*, y *aplicación del filtro*.

a. Configuración del filtro.

Dependiendo si se elige trabajar con un filtro FIR o IIR debe elegirse un código en C u otro. Luego deben establecerse en el mismo las características que definen el filtro. Se introducen el orden, el tipo (pasa bajo, pasa alto, pasa banda o elimina banda), se ingresa una frecuencia de corte única, si el filtro es pasa bajo o pasa alto, o dos frecuencias de corte si es pasa banda o elimina banda, finalmente se determina la ventana a aplicar (Rectangular, Hanning, Hamming o Blackman).

Establecida las características, el algoritmo crea un arreglo de longitud fija (longitud que depende del orden del filtro elegido), de los coeficientes que se utilizaran, de tipo punto flotante de 32 bits. El límite máximo de orden establecido es de 1000 para no sobre pasar la capacidad de almacenamiento del microcontrolador. Para el cálculo de coeficientes, primero se debe determinar qué tipo de filtro se utilizara, a saber:

- Filtro de Respuesta Finita al Impulso: FIR
- Filtro de Respuesta Infinita al Impulso: IIR

La función transferencia de los filtros FIR está dada por la siguiente ecuación:

$$H(z) = \sum_{n=0}^{N-1} b_n * z^{-n} \quad (1)$$

La función transferencia de los filtros IIR está dada por la siguiente ecuación:

$$H(z) = \frac{\sum_{k=0}^{N-1} b_k * z^{-k}}{1 + \sum_{k=0}^{M-1} b_k * z^{-k}} \quad (2)$$

Los filtros FIR ofrecen en general una respuesta de fase más lineal y no entran jamás en oscilación (es decir, no se vuelven inestables), ya que no poseen realimentación. Por otro lado, requieren un gran número de términos en sus ecuaciones y eso les hace más costosos en cuanto a cálculo o carga computacional. Un filtro FIR con un corte muy abrupto (es decir, que tenga una banda de transición muy corta) puede requerir hasta centenas de retardos. En cuanto a los filtros IIR, son muy eficaces y pueden proporcionar pendientes de corte muy pronunciadas. Por otro lado, al poseer características de realimentación (o feedback), tienen tendencia a entrar en oscilación y en resonancia. Y aunque los filtros IIR alcanzan una magnífica respuesta en amplitud, lo hacen a expensas de un comportamiento no lineal en fase.

Decidido el filtro, se calcula los coeficientes según las siguientes características:

- Pasa Bajo
- Pasa Alto
- Pasa Banda
- Elimina Banda

b. Inicialización del hardware

El temporizador utilizado es el Timer 2 en modo contador UP, con interrupción. En este caso, el periodo y el prescaler dependen del valor de la frecuencia de muestreo. El prescaler, se estableció con valor fijo 2, y el valor del periodo sigue la siguiente ecuación:

$$Period = \frac{84000000}{2 * F_s} - 1 \quad (3)$$

Esta ecuación determina que los límites de la frecuencia de configuración del Timer es desde 641 Hz hasta 48 MHz no inclusive, esto sería en el caso de que el ADC pueda capturar y convertir los datos a una frecuencia mayor a 48 MHz, pero en la realidad éste puede trabajar a un máximo de 1MHz. El valor mínimo de frecuencia es porque la variable periodo es de 16 bits (65535 valor máximo en valor decimal) y despejando la variable F_s se obtiene éste valor como resultado. Al entregar un número entero, indica que a mayor frecuencia de muestreo, la variable periodo puede poseer valores iguales para distintas frecuencias. En la sección IV se explica, en base a experimentos, que la frecuencia de muestreo está en función del valor de orden del filtro y del tipo de filtro, ya que el límite máximo para un filtro tipo FIR de orden 1 es 210 KHz, y un filtro IIR de orden 1 es 167 KHz.

- FIR => $641\text{Hz} < F_s < 210\text{ KHz}$
- IIR => $641\text{Hz} < F_s < 167\text{ KHz}$

La interrupción del timer únicamente se utiliza como habilitación de una bandera que es analizada en la función principal.

El ADC se configura en modo independiente a su máxima velocidad de conversión, con una resolución de 12 bits y alineación a la derecha.

El DAC se configura sin disparo, sin generación de onda, a 12 bits y con alineación a la derecha [4].

c. *Aplicación del filtro*

Esta sección de código se encuentra conformada por tres operaciones: *Captura del ADC y normalización, Aplicación del filtro, Desnormalización y Seteo del DAC.*

La primera consiste en capturar la señal y convertir el valor analógico en digital. Debido a que el conversor ADC no puede obtener valores de señales analógicas de tensión negativa, se montó la señal alterna sobre una continua (valor de offset conocido igual a 2048 en valores digitales) que luego es descontado del valor digital de la señal muestreada, permitiendo que la muestra sea un valor entero con signo. Luego se divide por 4096, que es la máxima escala del ADC (12 bits de resolución), estableciendo como resultado un valor en punto flotante entre -0.5 y 0.5.

Una vez adaptada la señal, el dato ingresa a una función *filtro FIR* o *filtro IIR*, según sea el tipo de filtro a utilizar, en donde se aplican las ecuaciones 1 o 2 según corresponda. Este proceso demanda un tiempo de cálculo proporcional a la numero de orden del filtro, y no al tipo de filtro, sean éstos pasa bajo, pasa alto, pasa banda o elimina banda.

Finalizado el proceso de aplicación del filtro, se obtiene como resultado un valor en punto flotante entre -0.5 y 0.5, que debe ser desnormalizado. El proceso es parecido al de normalización pero en sentido inverso. El valor obtenido es multiplicado por 4096, y sumado al valor de offset establecido de 2048. Este proceso establece el valor resultante entre 0 y 4095 y almacenado en una variable entera sin signo, la cual se utiliza para setear el DAC.

4 Pruebas realizadas

Dado que es difícil determinar en forma analítica, el tiempo de cálculo que demanda la etapa de aplicación del filtro a la señal de entrada, se realizaron pruebas con diferentes configuraciones de filtros, con el fin de determinar ese tiempo a modo experimental.

Se utilizó el siguiente equipamiento:

- Osciloscopio digital Tektronix TBS1102B.
- Generador de señal Keysight 33500B.
- Kit STM32F4 Discovery.

La primera prueba consistió en implementar un filtro FIR pasa bajo con frecuencia de corte en 4 KHz, y frecuencia de muestreo en 20KHz. El orden se varió escalonadamente de 1 a 120.

Con el generador se fijó una señal senoidal de 1Vpp, 0,75V de offset, y una frecuencia de variación automática de 100 Hz a 10 KHz en un lapso de 10 seg.

En una primera prueba, se conectó la salida del generador a un canal del osciloscopio y al ADC del microcontrolador, y la salida del DAC al otro canal del osciloscopio. Se procedió a corroborar el correcto funcionamiento del filtro en las condiciones establecidas y con un orden de 20.

Verificado el filtro, se conectó al osciloscopio dos pines auxiliares del kit (salidas digitales de testeo). Un pin conectado al canal 1 cumple la función de cambiar de estado en el momento en que interrumpe el timer, y el otro pin conectado al canal 2 se coloca en estado alto antes de la captura del ADC, y pasa a estado bajo después del seteo del DAC. El tiempo de transición de un estado al otro del canal 1, determina el periodo de la frecuencia de muestreo, y el periodo de la señal del canal 2 determina el tiempo necesario para aplicar el filtro.

De este experimento se obtuvo como resultado la siguiente tabla:

Tabla 1 – Máxima frecuencia de muestreo aproximada.

Número de Orden	Tiempo de cálculo [useg]	Máxima Fs (aprox)[Hz]
120	42,3	23641
100	36	27778
80	29,6	33784
60	23,3	42918
40	17	58824
20	10,7	93458
1	4,74	210815

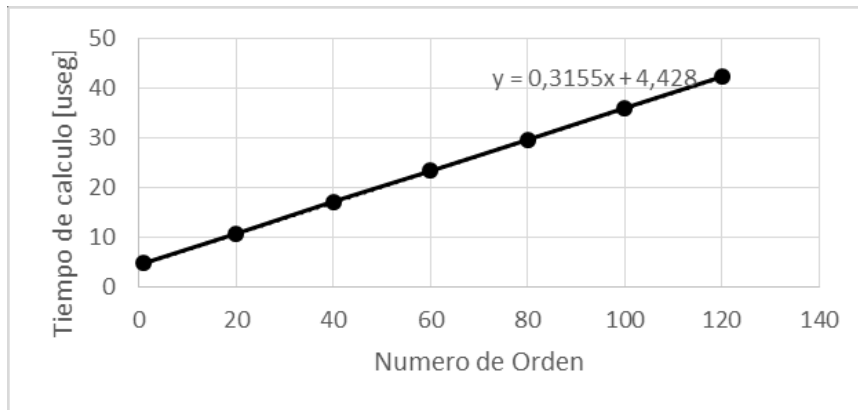


Figura 2 – Grafico número de orden vs tiempo de cálculo.

Como se ve en la figura 2, a medida que aumenta el valor del orden del filtro, aumenta el tiempo de cálculo necesario para aplicar el filtro. De este experimento se puede obtener la siguiente ecuación:

$$T_c = 0,3155 * N_c + 4,428 \quad (4)$$

En donde N_c es el número de orden a utilizar, y T_c es el tiempo que demora el microcontrolador en aplicar el filtro. El valor de la frecuencia máxima de muestreo para un orden dado, se obtiene redondeando hacia arriba la inversa de T_c . Utilizando esta ecuación, se determina que para un filtro de orden 1, es necesario un tiempo de cálculo de 5 useg aproximadamente, es decir que la máxima frecuencia de muestreo es de 210815 Hz para un filtro FIR.

De esta ecuación se puede despejar el número de orden, dejando N_c en función de T_c , permitiendo establecer el número de orden en función de la frecuencia de muestreo, como se muestra en la tabla 2.

$$N_c = 3,1636 * T_c - 13,772 \quad (5)$$

Tabla 2 – Máximo número de Orden.

Frecuencia de muestreo [Hz]	Tiempo [useg]	Máximo Número de Orden. (aprox)
210815	4,7	1,2
100000	10,0	17,9
80000	12,5	25,8
60000	16,7	39,0
40000	25,0	65,3
20000	50,0	144,4
10000	100,0	302,6

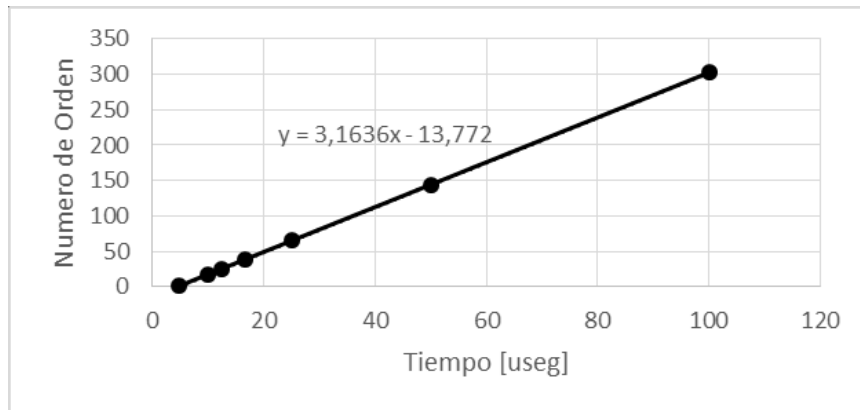


Figura 3 – Grafico Tiempo vs número de orden.

Este mismo procedimiento se realizó con los cuatro diferentes tipos de filtro FIR, determinando que sin importar el tipo de filtro que sea, el tiempo de cálculo requerido para realizar la parte analítica es igual, ya que depende de la cantidad de coeficientes.

En la segunda prueba, se estableció un filtro IIR pasa bajo con frecuencia de corte en 2 KHz y una frecuencia de muestreo de 10 KHz. Las condiciones de la señal de entrada se mantuvieron iguales a las utilizadas en la prueba del filtro FIR. El esquema de conexión fue el mismo al utilizado en la prueba anterior.

La prueba consistió en mantener fijo las características del filtro y variar escalonadamente el número de orden del filtro IIR, obteniendo como resultado el tiempo que requiere el microcontrolador para realizar el cálculo. La tabla 3 muestra el resultado de esta prueba.

Tabla 3 – Máxima frecuencia de muestreo aproximada.

Número de Orden	Tiempo de cálculo [useg]	Máxima Fs (aprox) [Hz]
60	83,5	11976
50	70,33	14219
40	57,2	17483
30	44,1	22676
20	30,9	32362
10	17,8	56180
1	5,9633	167692

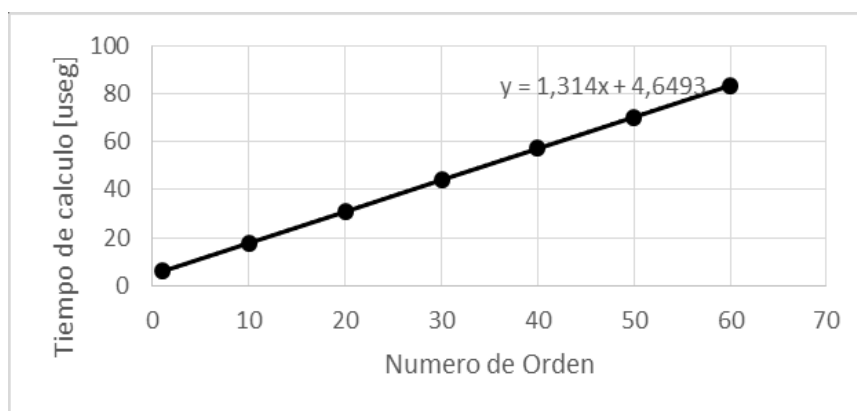


Figura 4 – Grafico núm. de orden vs tiempo de cálculo.

Como se observa en la figura 4, la curva tiene una respuesta lineal, igual a la curva de la figura 2, pudiéndose observar que a medida que se incrementa el número de orden, el tiempo de cálculo se incrementa proporcionalmente. La ecuación correspondiente a este grafico es la siguiente

$$T_c = 1.314 * N_c + 4,6493$$

La fórmula para obtener máxima frecuencia de muestreo es la misma que la hallada en la prueba anterior. Utilizando esta ecuación, se determina que para un filtro de orden 1, es necesario un tiempo de cálculo de 6 useg aproximadamente, es decir que la máxima frecuencia de muestreo es de 167692 Hz para un filtro IIR.

Despejando la variable N_c , del mismo modo que en el caso del filtro FIR se obtiene la fórmula que permite armar la tabla 4.

$$N_c = 0,761 * T_c - 3,5383$$

Tabla 4 – Máximo número de Orden.

Frecuencia de muestreo [Hz]	Tiempo [useg]	Máximo Número de Orden. (aprox)
167692	6,0	1,00
100000	10,0	4,07
80000	12,5	5,97
60000	16,7	9,15
40000	25,0	15,49
20000	50,0	34,51
10000	100,0	72,57

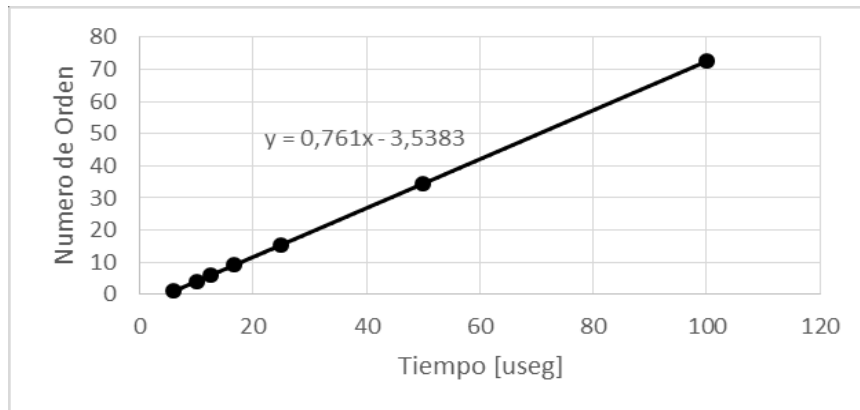


Figura 5 – Grafico Tiempo vs número de orden.

Como se aprecia en la figura 5, sigue una curva lineal al igual que la curva de la figura 3, determinando que a medida que disminuye la frecuencia de muestreo, aumenta el número de orden.

En la figura 6 se muestra la superposición de la figura 2 y la figura 4. Se puede notar que la pendiente de la curva del filtro IIR (línea sólida) es mayor que la pendiente del filtro FIR (línea punteada) estableciendo que para un mismo valor de número de orden, el filtro FIR tiene menor tiempo de cálculo que el filtro IIR.

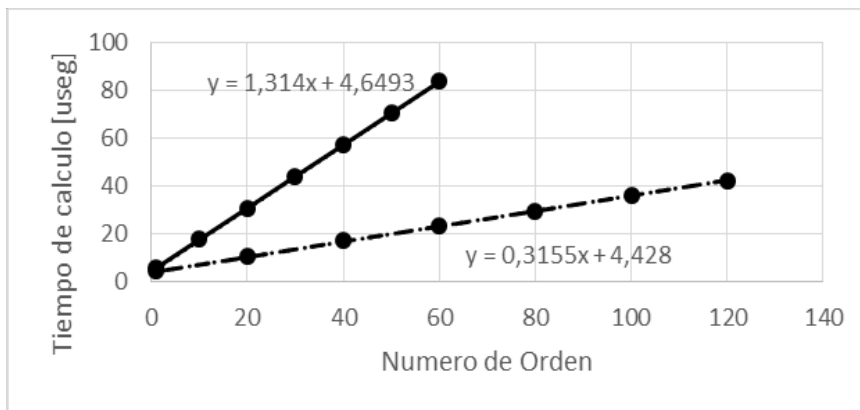


Figura 6 – Grafico número de orden vs Tiempo de cálculo.

5 Conclusión

La utilización de filtros digitales otorga ventaja en cuanto a facilidad de diseño y configuración, proporcionan la misma cantidad de componentes a medida que aumenta el orden ya que solo cuentan con un microcontrolador, y puede configurarse por medio de software su funcionamiento dependiendo de los requisitos del sistema donde

se encuentre, ya sea que se desee utilizar un FIR o IIR, un pasa bajo, pasa alto, pasa banda o elimina banda. La gran desventaja que presentan es el tiempo que requiere el procesamiento de datos cuando son de orden elevado, como por ejemplo un filtro pasa bajo FIR de orden 100, solo puede ser utilizado con una frecuencia de muestreo menor a 25 KHz, y si se desee utilizar un filtro IIR a la misma frecuencia de muestreo, el máximo número de orden es 26.

Otra desventaja que se demostró con este experimento es la gran diferencia que existe entre un filtro FIR con respecto a un IIR en función del número de orden y el tiempo de calculo que requiere el microcontrolador. En el caso del filtro FIR, el número de orden es igual a la cantidad de coeficientes del filtro, no sucede así en el caso de los filtros IIR, en donde la cantidad de coeficientes es 5 (cinco) veces mayor que el número de orden.

El ancho de banda de este tipo de filtros también es una variable a considerar, ya que la máxima frecuencia a la que trabaja el filtro es la frecuencia de Nyquist, equivalente a la mitad de la frecuencia de muestreo. Este es un impedimento en los filtros de tipo pasa alto o elimina banda, ya que éstos deben dejar pasar frecuencias altas a la frecuencia de corte y no comportarse como filtros pasa banda.

También se debe analizar el costo que requiere la realización de este tipo de filtros, ya que si las características del mismo son sencillas, puede ser reemplazado por un filtro analógico de menos costo.

Referencias

- [1]. Tratamiento Digital de Señales: Problemas y ejercicios resueltos. Soria, E.; Martínez, M.; Francés, J.V.; Camps, G. ISBN: 84-205-3559-1. Capítulo 6: Diseño de filtros digitales.
- [2]. ST Microelectronics. Datasheet de la familia de micro-controladores STM32F405xx y STM32F407xx. Junio, 2013.
Disponible en:
<http://www.st.com/web/en/resource/technical/document/datasheet/DM00037051.pdf>
- [3]. ST Microelectronics. Reference Manual RM0090. STM32F405xx, STM32F407xx, STM32F415xx and STM32F417xx. Advanced ARM-based 32-bit MCUs. Octubre, 2012.
Disponible en:
http://www.st.com/web/en/resource/technical/document/reference_manual/DM00031020.pdf
- [4]. ST Microelectronics. User Manual UM1472. STM32F4 DISCOVERY. STM32F4 high-performance discovery board. Enero, 2002.
Disponible en: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/DM00039084.pdf