

Alberto Pimentel Martínez

Implementación de la transformada rápida de Fourier en un microcontrolador PIC de alto desempeño para el procesamiento de audio.

2011



Universidad Autónoma de Querétaro
Facultad de Querétaro.

Implementación de la transformada rápida de Fourier en
un microcontrolador PIC de alto desempeño para el
procesamiento de audio.

Tesis

Que como parte de los requisitos para obtener el título de
INGENIERO EN AUTOMATIZACIÓN.

Presenta

Pimentel Martínez Alberto

Dirigida por

Dr. Manuel Toledano Ayala

Santiago de Querétaro,
Querétaro, 07/10/2011



Universidad Autónoma de Querétaro

Facultad de Ingeniería

Ingeniería en Automatización

Implementación de la transformada rápida de Fourier en un microcontrolador PIC de alto desempeño para el procesamiento de audio.

TESIS

Que como parte de los requisitos para obtener el grado de

INGENIERO EN AUTOMATIZACIÓN

Presenta:

Alberto Pimentel Martínez

Dirigido por:

Dr. Manuel Toledano Ayala

SINODALES

Dr. Manuel Toledano Ayala

Presidente

Firma

Dr. Genaro Martín Soto Zarazúa

Secretario

Firma

Dr. Edgar Rivas Araiza

Vocal

Firma

M.C Juvenal Rodríguez Reséndiz

Suplente

Firma

Dr. Gilberto Herrera Ruiz

Centro Universitario

Querétaro, Qro.

18/05/2011

México

RESUMEN

El objetivo de este proyecto es procesar señales de audio con un microcontrolador (PIC) de alto desempeño. Primeramente, se diseñó una etapa de potencia, optoaislando perfectamente al microcontrolador de los dispositivos de potencia. Una vez probada la etapa de potencia, se realizó el diseño del acondicionamiento de señales, el cual consiste de una etapa de amplificación, otra de filtrado y finalmente una etapa que consistió en un amplificador sumador, el objetivo de éste sumador es introducir a la señal un voltaje de offset para evitar que lleguen al microcontrolador señales negativas de voltaje. Una vez terminada la parte analógica se comenzó a trabajar con el programa del microcontrolador. La primera tarea que se hizo con el microcontrolador fue programar la interrupción del ADC para muestrear correctamente la señal proveniente del filtro. Después se implementó el algoritmo de la FFT y se procedió a realizar pruebas, los resultados obtenidos del microcontrolador fueron comparados con los resultados obtenidos de Matlab para el mismo vector de datos. Una interfaz gráfica desarrollada en C# obtiene los resultados del microcontrolador y los muestra en pantalla.

(Palabras clave: FFT, microcontroladores, procesamiento de señales, filtro anti-aliasing.)

SUMMARY

The main goal of this research is to process audio signals using a high performance microcontroller (PIC18F4550). First, a power stage was designed opto-isolating the microcontroller from the power devices. After, a signal conditioning stage was designed, which in turn consists of an amplifying module, a filtration step, and finally an adder amplifier with the purpose of adding an offset voltage in order to avoid negative voltage signals in the microcontroller. Once the analogic part was finished, the microcontroller's program was designed. The first task accomplished with the microcontroller was to program the analogical digital converter's interruption in order to scan the signal from the filter. Then, the fast Fourier transform algorithm was implemented and tests were conducted. The results obtained from microcontroller were compared to the ones obtained from the same data vector in the software Matlab. Finally, a graphic interface developed in C# acquires the microcontroller's data and shows them on screen.

Key words: FFT, microcontroller, signal processing, anti-aliasing filter.

DEDICATORIAS

A mis padres y hermano

AGRADECIMIENTOS

Primeramente quiero agradecer a Dios por permitirme llegar hasta este momento.

A mi padre Alberto Pimentel Martínez y Emy Martínez Samperio y a mi hermano Rodrigo Pimentel Martínez, quienes me apoyaron en todo momento durante mi formación académica.

A los profesores que me apoyaron a lo largo de la licenciatura.

A mis amigos que me acompañaron a lo largo de mi formación académica, Moisés, Jorge, Genaro, Roberto y a Fortino, que me ayudaron a superarme y esforzarme en mis trabajos.

INDICE

Contenido

RESUMEN.....	i
SUMMARY	ii
DEDICATORIAS.....	iii
AGRADECIMIENTOS.....	iv
INDICE DE CUADROS.....	vii
INDICE DE FIGURAS.....	viii
1. INTRODUCCION.....	1
1.1 Descripción del problema.....	2
1.2 Justificación.....	2
1.3 Objetivo.....	3
1.4 Hipótesis.....	3
2. REVISIÓN DE LA LITERATURA.....	4
2.1 Antecedentes.....	4
2.2 Marco teórico.....	7
2.2.1 Acondicionamiento de señales.....	7
2.2.2 Sistemas discretos.....	33
2.2.3 Sistema estático.....	34
2.2.4 Sistema dinámico.....	34
2.2.6 Muestreo y cuantización.....	34
2.2.7 Análisis de Fourier.....	38
3. METODOLOGÍA.....	42
3.1 Etapa de potencia.....	42
3.2 Acondicionamiento de señal.....	43
3.3 Implementación de la Transformada rápida de Fourier (FFT).	51
3.4 Interfaz gráfica de usuario (GUI).	52
4 RESULTADOS Y DISCUSIÓN.....	54
4.1 Amplificador no inversor.....	54
4.2 Filtrado.....	55
4.3 Voltaje offset.....	58
4.4 Resultados del procesamiento.	59
4.5 Resultados mostrados en la Interfaz gráfica de usuario.	61

5 CONCLUSIONES	66
Bibliografía.....	67
APENDICE	68

INDICE DE CUADROS

Cuadro		Página
2.1	Factores Q encadenados para filtros pasa bajas de Butterworth.	33
3.1	Valores de R, C_1 y C_2 para las 5 etapas del filtro anti-alias.	49
4.1	Respuesta del sistema a diferentes frecuencias de entrada.	60

INDICE DE FIGURAS

Figura		Página
2.1	Amplificador no inversor de alterna.	9
2.2	Amplificador sumador que utiliza ambos lados del amplificador operacional.	10
2.3	Respuesta ideal del filtro pasa bajas.	11
2.4	Respuesta ideal del filtro pasa altas.	12
2.5	Respuesta ideal filtro pasa banda.	12
2.6	Respuesta ideal de un filtro de banda eliminada.	14
2.7	Respuesta ideal de un filtro pasa todo.	15
2.8	Respuesta de un filtro pasa bajas Butterworth.	17
2.9	Respuesta Chebyshev para filtro pasa bajas.	18
2.10	Respuesta inversa de Chebyshev de un filtro pasa bajas.	19
2.11	Respuesta elíptica de un filtro pasa bajas.	20
2.12	Respuesta Bessel de un filtro pasa bajas.	21
2.13	Filtro LC pasa bajas.	22

Figura		Página
2.14	Efecto de Q en la respuesta de segundo orden.	23
2.15	Frecuencias de resonancia y factores Q en un filtro de orden superior.	24
2.16	Etapas pasa bajas de primer orden.	26
2.17	Etapas pasa altas de primer orden.	28
2.18	Etapas de segundo orden con VCVS para filtros Butterworth y Bessel.	30
2.19	Conexión en cascada de filtros de orden superior.	32
2.20	Representación de un sistema discreto.	33
2.21	Transformación de secuencias.	34
2.22	Operación ideal de muestreo.	35
2.23	Diagrama en bloques de un sistema de muestreo y retención.	36
2.24	Mariposa característica para el algoritmo FFT de reducción en frecuencia.	40
2.25	Mariposa característica para el algoritmo FFT de reducción en el tiempo.	41
3.1	Diagrama de bloque de la etapa de potencia.	42

Figura		Página
3.2	Etapas del procesamiento de señales.	43
3.3	Señal acondicionada.	44
3.4	Señal de audio de la computadora sin amplificar.	45
3.5	Amplificación de la señal de audio.	46
3.6	Medición de la frecuencia de muestreo.	47
3.7	Circuito de offset.	50
3.8	Diagrama de bloques del algoritmo de la FFT.	51
3.9	Diagrama de funcionamiento de la FFT.	52
3.10	Pantalla de la interfaz gráfica de usuario (GUI).	52
4.1	Medición de la señal de audio.	54
4.2	Respuestas del filtro.	55
4.3	Respuesta del acondicionamiento de señal.	58
4.4	Voltaje de offset.	59
4.5	Gráfica de comparación entre la frecuencia de entrada y la frecuencia medida.	60

Figura		Página
4.6	Señal de entrada de 5068Hz.	61
4.7	Resultados desplegados por la interfaz para la señal de entrada de 5068Hz.	62
4.8	Frecuencia de entrada de 4219Hz.	62
4.9	Resultados desplegados por la interfaz para la señal de entrada de 4219Hz.	63
4.10	Señal de entrada de audio 1.	63
4.11	Respuesta a la señal de entrada de audio 1.	64
4.12	Señal de entrada de audio 2.	64
4.13.	Respuesta a la señal de entrada de audio 2.	65

1. INTRODUCCION

El procesamiento digital de audio tiene un gran número de aplicaciones, una de ellas es el famoso espectáculo de las fuentes bailarinas, las cuales reaccionan a cualquier sonido, con luces y aventando chorros de agua dependiendo de la frecuencia del sonido, fue esta aplicación la que motivó este proyecto, sin embargo se puede adaptar sin ningún problema a cualquier otra aplicación que implique procesar audio, simplemente lo que se necesita es adecuarlo a las necesidades de cada aplicación.

El análisis de Fourier es muy complejo, por esta razón cuando el PIC (circuito integrado programable) está realizando el algoritmo de la FFT (Fast Fourier Transform), deja de tomar muestras de la señal para dedicarse únicamente a realizar los cálculos, esto nos lleva a la pérdida de un segmento de la señal, sin embargo al configurar al PIC a 48MHz, se puede realizar el algoritmo de la FFT a una gran velocidad haciendo que el tiempo de duración del segmento de la señal perdido sea muy corto.

Otro aspecto importante es el tratamiento de las señales de entrada, hay que eliminar el ruido de las señales y además amplificarlas, si bien la parte digital no implica ruido y es muy cómoda de trabajar, sí la señal de entrada es defectuosa por ruido o por alguna otra razón la parte digital lo procesara junto con estos defectos y el resultado no será el adecuado.

Existen diferentes tipos de PIC's con diferentes características, al escoger uno se tiene que estar seguro que es el más adecuado para el proyecto a realizar, el PIC más potente que existe es el PIC32, el cual es el ideal para este proyecto, sin embargo otros tipos de PIC's también pueden ser capaces de realizar esto con una adecuada administración de sus recursos.

1.1 Descripción del problema.

En la actualidad el procesamiento digital de señales es una rama importante de la electrónica, tiene múltiples aplicaciones, una de ellas son los analizadores de espectro, los cuales obtienen las diferentes frecuencias que componen a una señal, y esta aplicación se puede derivar en aplicaciones como el de las fuentes bailarinas.

Los PIC's no cuentan con mucha capacidad de memoria, esto limita la cantidad de número de muestras que podemos tomar de la señal, para realizar un analizador de espectro muy preciso, es necesario un muestreo muy grande, sin embargo la aplicación de las fuentes bailarinas no requiere de mucha precisión, por lo que el PIC es una buena herramienta para este propósito.

El objetivo de realizar la FFT para las fuentes bailarinas es que se vea una reacción espontánea para diferentes tipos de música, lo que nos lleva a un mejor espectáculo que si únicamente se programaran secuencias para cada pieza de música.

1.2 Justificación

Existen muchos dispositivos con lo que es fácil realizar el análisis de Fourier, en este caso el objetivo es realizarlo en un microcontrolador, otros dispositivos como los FPGA's y los DSP's, tienen el problema de que la mayoría únicamente se encuentran en empaque de montaje superficial, este empaquetamiento es más complejo de manejar ya que requiere de equipos especiales para soldarlos.

Los microcontroladores también tienen presentaciones de montaje superficial, pero también se pueden conseguir en un empaquetado tipo DIP, el cual permite realizar pruebas antes de implementar el diseño en una tarjeta.

Por otro lado, en el FPGA se requiere más tiempo para cargarle los programas ya que se tienen que sintetizar los códigos, cargar un programa en el PIC es mucho más rápido, en especial si se trabaja con el puerto ICSP.

Uno de los ideales de este proyecto es utilizarlo para fines didácticos, para que los estudiantes de ingeniería puedan observar las aplicaciones que tiene la FFT, en este caso, un microcontrolador está perfectamente al alcance de los estudiantes, debido a que es más fácil de utilizar y económicamente es más accesible que otros dispositivos.

1.3 Objetivo

Procesar señales de audio con un microcontrolador de alto desempeño para realizar una aplicación de software de fuentes bailarinas.

1.4 Hipótesis

El procesamiento de audio en un microcontrolador de alto desempeño utilizando la FFT permite un control adecuado de un proceso interactivo como lo puede ser una fuente bailarina.

2. REVISIÓN DE LA LITERATURA

2.1 Antecedentes

El desarrollo de las series de Fourier comienza a principios del siglo XVIII, cuando el concepto de función era el de una expresión matemática, una sola fórmula que describía su extensión. La forma de onda triangular no habría sido considerada como una función sino dos, y una gráfica arbitraria tal vez no podría haberse representado del todo por una función.

La mitad del siglo XVIII vio el estudio de las cuerdas vibrantes, sujeto a una ecuación diferencial parcial, tres famosos personajes de esa época se enfocaron en éste tema. La solución en forma funcional, suponiendo que la forma inicial de la cuerda está descrita por una expresión, fue por primera vez propuesta por d'Alembert en 1784 y extendida un año después por Euler. En 1753, Bernoulli sugirió por primera vez una serie de seno trigonométrico como solución general para el movimiento de la cuerda cuando parte del reposo. En 1767, Lagrange obtuvo una solución tentadoramente cercana a la que ahora nos es familiar. La solución que Lagrange transformó para ajustarla a la forma dada por Euler involucrada una dependencia del tiempo que, hecha cero, conduciría a una serie de senos que describe el desplazamiento de la cuerda.

Clairaut, en 1757, y Euler, dos décadas después, propusieron la evaluación de los coeficientes de la serie trigonométrica cuya existencia ya había sido demostrada por otros medios. Pero fueron escépticos y creyeron que tales resultados podrían aplicarse sólo a las funciones como ellos las entendían (una expresión analítica).

En el siglo XIX apareció Joseph Fourier quien durante el curso de sus estudios sobre la conducción del calor dio el paso decisivo a sugerir una representación en series trigonométricas para una función completamente arbitraria, sin importar si tal función podría o no describirse por expresiones matemáticas sobre toda su extensión. También se dio cuenta de que, dada una

función sobre un determinado intervalo, no había manera de determinar su valor fuera de dicho intervalo y esto anunció el concepto de extensión periódica, tan importante para la representación en serie. Con estas ideas Fourier afirmó que los estudios de Euler y d'Alembert eran inadecuados debido a que no consideraban el intervalo sobre el que tales desarrollos eran válidos, al mismo tiempo que negaban su existencia para una función arbitraria.

Cauchy dio el primer paso hacia la credibilidad matemática del trabajo de Fourier al proponer en 1823 una definición integral como el límite de una suma y al darle una interpretación geométrica. Pero fue Dirichlet quien verdaderamente parchó los agujeros, santificando los métodos de Fourier con demostraciones rigurosas, y sentando las bases para avances posteriores.

Riemann amplió el concepto de integral de una función continua más allá de la definición de Cauchy como el límite de una suma para incluir funciones que ni son continuas ni contienen un número finito de discontinuidades, y lo más importante, demostró que, para funciones integrables acotadas, los coeficientes de la serie de Fourier tienden a cero a medida que el índice armónico aumenta sin límite, sugiriendo que la convergencia en un punto depende sólo del comportamiento de la función en la vecindad de ese punto.

En 1898 Michelson desarrolló un analizador armónico, pero no pudo obtener la reconstrucción de onda de cuadrado perfecto (sin sobrepaso) al sumar sus 80 armónicos. Un año después, en 1899, Gibbs ofreció una explicación del efecto que ahora lleva su nombre.

En 1904, Fejer demostró que la suma por el método de medias aritméticas da como resultado un efecto de suavización en las discontinuidades y la ausencia del fenómeno Gibbs. Este fue el precursor del concepto de ventanas espectrales que ahora tiene un uso tan común en el análisis espectral.

En el siglo XX, Lebesgue siguiendo los pasos de Borel, quien formuló en 1898 el concepto de medida de un conjunto, Lebesgue reformuló sobre esta base la noción de integral en 1902 y en 1903 extendió el teorema de Riemann para incluir funciones no acotadas, siempre que la integración fuese entendida

en el sentido de Lebesgue. En 1905 presentó una condición suficiente nueva para la convergencia que incluía todas las anteriores. Incluso planteó el molesto problema de la integración término a término de las series de Fourier, demostrando que ya no depende de la convergencia uniforme de las series (Ambardar, 2002).

En el siglo XXI se ha utilizado el método de Fourier para el procesamiento de señales, por esta razón se han desarrollado diversos algoritmos computacionales que son capaces de realizar la transformada de Fourier, generalmente estos algoritmos solo pueden trabajar con un número de muestras que sea potencia de 2, sin embargo, se han desarrollado algoritmos que no dependen de la condición de que el número de muestras sea una potencia de 2, uno de estos algoritmos es el algoritmo Bruun (Murakami, 2002). Jo y Sunwoo (2005), trabajaron con los algoritmos Radix-2 y Radix-4 describiendo las ventajas y desventajas de estos algoritmos.

En los últimos años no solo se han perfeccionado los algoritmos de la FFT, sino que también se han desarrollado circuitos analógicos potentes, los cuales se utilizan para filtrar las señales, esto se hace con el fin de evitar el fenómeno del Aliasing, para evitar este fenómeno se utiliza el filtro pasa-bajas, aunque también el filtro pasa-banda es útil. Konrad y Agniel (2006) usaron tres modelos para encontrar el filtro anti-aliasing ideal para señales de imágenes, estos modelos son: red ortogonal, red no ortogonal y la unión de cambio de red. Los ADC's también han tenido una importante evolución, en 2008 se propusieron diferentes ADC's para utilizarlos en aplicaciones muy complejas, en donde es necesario utilizar un ADC y circuitos analógicos que demanden menos potencia a la fuente de alimentación, un ADC de alto desempeño y bajo consumo de potencia es capaz de realizar conversiones más estables (Kim et al., 2008; Yang et al., 2008).

2.2 Marco teórico

Para analizar una señal analógica, es necesario adaptarla a los rangos de voltaje que soporta el dispositivo digital que se esté utilizando, y además se tienen que filtrar las señales para eliminar señales de ruido y evitar el fenómeno del alias, a este proceso se le llama acondicionamiento de señales.

2.2.1 Acondicionamiento de señales

Los sistemas de adquisición de datos (DAQ) basados en PC y dispositivos insertables son usados en un amplio rango de aplicaciones en los laboratorios, en el campo y en el piso de una planta de manufactura.

Típicamente, los dispositivos DAQ insertables son instrumentos de propósito general diseñados para medir señales de voltaje.

El problema es que la mayoría de los sensores y transductores generan señales que debe acondicionar antes de que un dispositivo DAQ pueda adquirir con precisión la señal. Este procesamiento al frente, conocido como acondicionamiento de señal, incluye funciones como amplificación, filtrado y aislamiento eléctrico. Es así que la mayoría de los sistemas DAQ basados en PC incluyen algún tipo de acondicionamiento de señal además del dispositivo DAQ y la PC.

Los amplificadores operacionales son los dispositivos que se utilizan para diseñar etapas de acondicionamiento de señal. Los amplificadores operacionales tienen diferentes configuraciones que realizan diferentes funciones. Para los acondicionamientos de señal las configuraciones de amplificador operacional que conviene analizar son el amplificador sumador y el amplificador no inversor.

Circuito con amplificador no inversor

El amplificador no inversor es uno de los circuitos básicos con amplificador operacional. Entre las ventajas de éste se incluye la ganancia de tensión estable, la alta impedancia de entrada y la baja impedancia de salida.

Amplificador de alterna acoplado

La Figura 2.1 muestra un amplificador de alterna acoplado no inversor y sus ecuaciones para el análisis. C_1 y C_2 son condensadores de acoplo. C_3 es un condensador de desacoplo. Utilizar un condensador de desacoplo tiene la ventaja de minimizar la tensión de offset de salida. Cuando el amplificador trabaja en la banda de frecuencias medias, el condensador de desacoplo presenta una impedancia muy baja. Por lo tanto, el terminal inferior de R_1 está a tierra de alterna. En la banda de frecuencias medias, la fracción de realimentación se muestra en la ecuación 2.1:

$$B = \frac{R_1}{R_1 + R_f} \quad (2.1)$$

En este caso el circuito amplifica la tensión de entrada como se ha descrito anteriormente.

Cuando la frecuencia es cero, el condensador de desacoplo C_3 es un circuito abierto y la fracción de realimentación B aumenta hasta el valor de la unidad.

Este resultado es válido si definimos ∞ como un valor extremadamente grande, que es el valor de la impedancia a la frecuencia cero. Con B igual a 1, la ganancia de tensión en lazo cerrado es igual a la unidad. Esto reduce la tensión de offset de salida a un valor mínimo, la ecuación 2.2 muestra como calcular la ganancia del amplificador.

$$A_v = \frac{R_f}{R_1} + 1 \quad (2.2)$$

$$f_2 = \frac{f_{unidad}}{A_v} \quad (2.3)$$

$$f_{c1} = \frac{1}{2\pi R_2 C_1} \quad (2.4)$$

$$f_{c2} = \frac{1}{2\pi R_L C_2} \quad (2.5)$$

$$f_{c3} = \frac{1}{2\pi R_1 C_3} \quad (2.6)$$

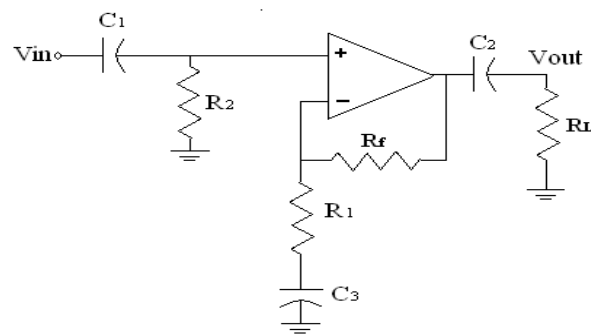


Figura 2.1 Amplificador no inversor de alterna.

Sumador en ambas entradas

La Figura 2.2 muestra un circuito sumador que tiene entradas inversora y no inversora. La parte inversora del amplificador tiene dos canales de entrada y la parte no inversora tiene dos canales de entrada. La ganancia total es igual a la superposición de las ganancias de cada canal, la ecuación 2.7 muestra como calcular la sumatoria total de las señales de entrada y las ecuaciones de la 2.8 a la 2.11 muestran como calcular las ganancias de voltaje de las señales de entrada.

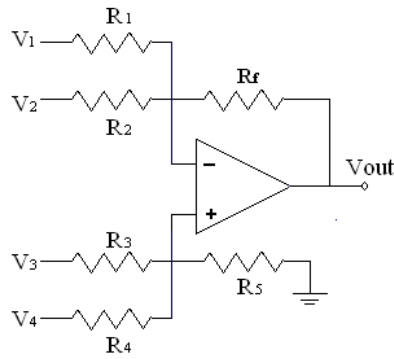


Figura 2.2 Amplificador sumador que utiliza ambos lados del amplificador operacional (Malvino y Bates, 2007).

Ecuaciones:

$$V_{out} = A_{v1}V_1 + A_{v2}V_2 + A_{v3}V_3 + A_{v4}V_4 \quad (2.7)$$

$$A_{v1} = \frac{-R_f}{R_1} \quad (2.8)$$

$$A_{v2} = \frac{-R_f}{R_2} \quad (2.9)$$

$$A_{v3} = \left(\frac{R_f}{R_1 \parallel R_2} + 1 \right) \left(\frac{R_4 \parallel R_5}{R_3 + R_4 \parallel R_5} \right) \quad (2.10)$$

$$A_{v4} = \left(\frac{R_f}{R_1 \parallel R_2} + 1 \right) \left(\frac{R_3 \parallel R_5}{R_4 + R_3 \parallel R_5} \right) \quad (2.11)$$

Filtros

Los filtros pueden ser implementados mediante software o hardware dependiendo del tipo de filtro, estos elementos nos permiten eliminar las señales de ruido que se mezclan con la señal. Los filtros pueden ser analógicos o digitales.

Filtro pasa bajas

Un filtro pasa bajas deja pasar todas las frecuencias desde cero hasta la frecuencia de corte y bloquea todas las frecuencias por encima de la frecuencia de corte.

En un filtro pasa bajas, las frecuencias entre cero y la frecuencia de corte definen la banda de paso. Las frecuencias por encima de la frecuencia de corte definen la banda eliminada. La región comprendida entre la banda de paso y la banda eliminada se denomina zona de transición. Un filtro pasa bajas ideal tiene una atenuación (pérdida de señal) de cero en la banda de paso, una atenuación infinita en la banda eliminada y una transición vertical, la Figura 2.3 ilustra esta idea.

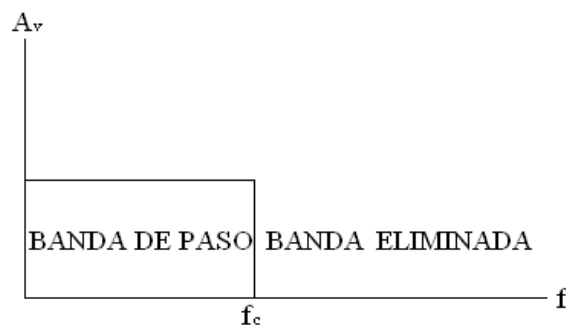


Figura 2.3 Respuesta ideal del filtro pasa bajas (Malvino y Bates, 2007).

Filtro pasa altas

Un filtro pasa altas bloquea todas las frecuencias desde cero hasta la frecuencia de corte y deja pasar todas las frecuencias por encima de la frecuencia de corte.

Con un filtro pasa altas, las frecuencias entre cero y la frecuencia de corte definen la banda eliminada. Un filtro pasa altas ideal tiene una atenuación infinita en la banda eliminada, una atenuación de cero en la banda de paso y una transición vertical, la Figura 2.4 ilustra al filtro pasa altas ideal.

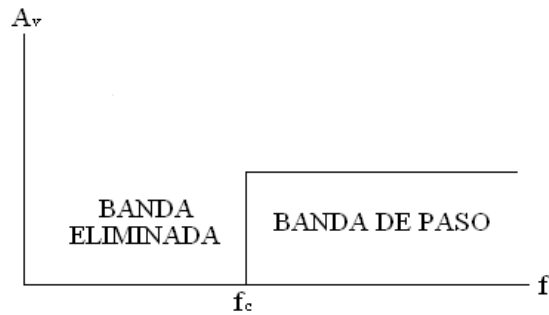


Figura 2.4 Respuesta ideal del filtro pasa altas (Malvino y Bates, 2007).

Filtro pasa banda

Un filtro pasa banda resulta útil cuando se desea sintonizar una señal de radio o de televisión. También es útil en los equipos de comunicaciones telefónicas, para separar diferentes conversaciones telefónicas que se están transmitiendo simultáneamente a través del mismo camino de comunicación. La Figura 2.5 muestra la frecuencia en respuesta ideal de un filtro pasa banda.

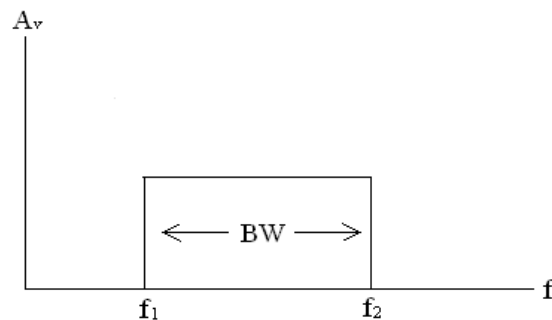


Figura 2.5 Respuesta ideal filtro pasa banda (Malvino y Bates, 2007).

En un filtro pasa banda, la banda de paso comprende todas las frecuencias entre la frecuencia de corte inferior y la frecuencia de corte superior. Las frecuencias por debajo de la frecuencia de corte inferior y por encima de la de corte superior definen la banda eliminada. Un filtro pasa banda ideal tiene una atenuación de cero en la banda de paso, una atenuación infinita en la banda eliminada y dos transiciones verticales.

El ancho de banda (BW, bandwidth) de un filtro pasa banda es la diferencia entre las frecuencias de corte superior e inferior a -3dB, la ecuación 2.12 muestra el cálculo del ancho de banda:

$$BW = f_2 - f_1 \quad (2.12)$$

La frecuencia central se indica con f_0 y queda determinada por la media geométrica de las dos frecuencias de corte tal como se ve en la ecuación 2.13:

$$f_0 = \sqrt{f_1 f_2} \quad (2.13)$$

Para evitar las interferencias entre las distintas conversaciones telefónicas, los filtros pasa banda tienen respuestas que se aproximan a la respuesta en frecuencia mostrada en la Figura 2.5. El factor Q de un filtro pasa banda se define como la frecuencia central dividida entre el ancho de banda, esto se ilustra en la ecuación 2.14:

$$Q = \frac{f_0}{BW} \quad (2.14)$$

Cuando el factor Q es mayor que 10, la frecuencia central puede aproximarse a la media aritmética de las frecuencias de corte, como se observa en la ecuación 2.15:

$$f_0 \cong \frac{f_1 + f_2}{2} \quad (2.15)$$

Si Q es menor que 1, el filtro pasa banda se denomina filtro de banda ancha. Si Q es mayor que 1, el filtro se denomina filtro de banda estrecha.

Filtro de banda eliminada

La Figura 2.6 muestra la respuesta en frecuencia ideal de un filtro de banda eliminada. Este tipo de filtro deja pasar todas las frecuencias desde cero

hasta la frecuencia de corte inferior, bloquea todas las frecuencias comprendidas entre las frecuencias de corte inferior y superior y, por último, deja pasar todas las frecuencias por encima de la frecuencia de corte superior.

En un filtro de banda eliminada, la banda eliminada comprende todas las frecuencias entre las frecuencias de corte inferior y superior. Las frecuencias por debajo de la frecuencia de corte inferior y por encima de la frecuencia de corte superior definen la banda de paso. Un filtro de banda eliminada ideal tiene una atenuación infinita en la banda eliminada, no presenta atenuación en la banda de paso y dos transiciones verticales.

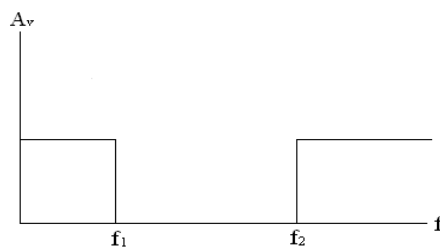


Figura 2.6 Respuesta ideal de un filtro de banda eliminada.

Las definiciones para el ancho de banda, banda estrecha y la frecuencia central son las mismas que las dadas anteriormente.

Filtro pasa todo

La Figura 2.7 muestra la respuesta en frecuencia de un filtro pasa todo ideal. Solo presenta banda de paso y no tiene banda eliminada. Por tanto, deja pasar todas las frecuencias entre cero e infinito. Puede parecer algo raro utilizar un filtro como éste, ya que tiene una atenuación de cero para todas las frecuencias. La razón de emplear este filtro es por el efecto que tiene sobre la fase de las señales que pasan a su través. El filtro pasa todo es útil cuando queremos producir un determinado desplazamiento de fase en la señal que se va a filtrar sin modificar su amplitud.

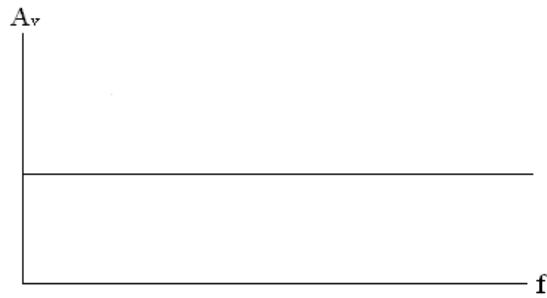


Figura 2.7 Respuesta ideal de un filtro pasa todo.

La respuesta de fase de un filtro se define como la gráfica del desplazamiento de fase en función de la frecuencia.

En la respuesta de fase de un filtro pasa todo cada una de las frecuencias puede desplazarse una cierta cantidad cuando pasa a través del filtro.

Respuestas aproximadas

Las respuestas ideales presentadas anteriormente son imposibles de implementar en circuitos prácticos, pero existen cinco aproximaciones estándar que se usan como solución de compromiso a las respuestas ideales. Cada una de estas aproximaciones ofrece una ventaja que las restantes no tienen.

Atenuación: la atenuación hace referencia a la pérdida de señal, considerando una tensión de entrada constante, la atenuación se define como la tensión de salida para cualquier frecuencia dividida entre la tensión de salida en la banda media de frecuencias, esto se muestra en la ecuación 2.16:

$$Atenuación = \frac{v_{out}}{v_{out(media)}} \quad (2.16)$$

Normalmente, la atenuación se expresa en decibelios utilizando la ecuación 2.17:

$$\text{Atenuación en decibelios} = 20 \log \text{atenuación} \quad (2.17)$$

Atenuación de la banda de paso y de la banda eliminada

En el diseño y análisis de filtros, el filtro pasa bajas se utiliza como un prototipo, un circuito básico que se puede modificar para obtener otros circuitos, normalmente, cualquier problema con un filtro se convierte en el problema equivalente con un filtro pasa bajas y se resuelve; después, la solución se transforma de nuevo al tipo de filtro original.

Una atenuación de cero en la banda de paso, una atenuación infinita en la banda eliminada y una transición vertical no son datos realistas. Para construir un filtro pasa bajas práctico existen diferentes aproximaciones. Dependiendo de la aplicación se escoge alguna de las aproximaciones.

Orden de un filtro

El orden de un filtro pasivo (que se simboliza con n) es igual al número de bobinas y condensadores que contiene el filtro. Si un filtro pasivo tiene dos bobinas y dos condensadores, $n=4$. Si un filtro pasivo tiene cinco bobinas y cinco condensadores, $n=10$. Por lo tanto, el orden nos dice cómo de complejo es el filtro. Cuanto mayor sea el orden más complejo es el filtro.

El orden de un filtro activo depende del número de circuitos RC (denominados polos) que contenga. Si un filtro activo contiene ocho circuitos RC, $n=8$. Normalmente, contar los circuitos RC individuales que contiene un filtro activo es complicado. Por lo tanto, se utiliza el método mostrado en la ecuación 2.18:

$$n \cong \text{número de condensadores} \quad (2.18)$$

Aproximación de Butterworth

La aproximación de Butterworth a veces se denomina aproximación maximalmente plana porque la atenuación de la banda de paso es cero prácticamente en toda la banda y disminuye gradualmente a A_p en el punto de inflexión de la banda de paso. Por encima de la frecuencia de corte, la respuesta disminuye con una pendiente de aproximadamente $20n$ dB por década, siendo n el orden del filtro, la ecuación 2.19 muestra como calcular la pendiente del filtro medida en década y la ecuación 2.20 muestra el cálculo de la pendiente medida en octava:

$$\text{Pendiente} = 20n \text{ dB/década} \quad (2.19)$$

La pendiente equivalente medida en octava es:

$$\text{Pendiente} = 6n \text{ dB/octava} \quad (2.20)$$

Por ejemplo, un filtro de Butterworth de primer orden presenta una pendiente de 20dB por década o 6dB por octava; un filtro de cuarto orden disminuye con una pendiente de 80dB por década, o 24 dB por octava; un filtro de noveno orden presenta una pendiente de 180 dB por década, o 54 dB por octava; y así sucesivamente.

La Figura 2.8 muestra la respuesta de un filtro pasa bajas de Butterworth.

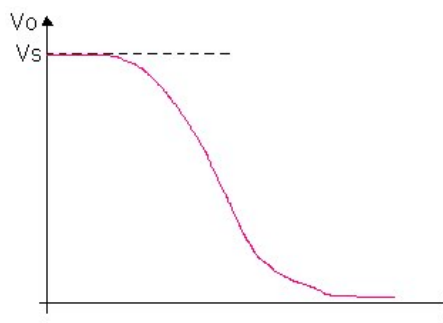


Figura 2.8 respuesta de un filtro pasa bajas Butterworth

La principal ventaja de un filtro de Butterworth es lo plana que es la respuesta en la banda de paso. La principal desventaja es lo relativamente despacio que disminuye en la zona de transición comparado con otras aproximaciones.

Aproximación de Chebyshev

En algunas aplicaciones, una respuesta plana en la banda de paso no es importante. En este caso, puede ser preferible la aproximación de Chebyshev porque decae más rápidamente en la región de transición que un filtro de Butterworth. El precio que hay que pagar por ésta rápida caída es que aparece rizado en la banda de paso de la respuesta en frecuencia.

La Figura 2.9 muestra la respuesta de un filtro pasa bajas de Chebyshev:

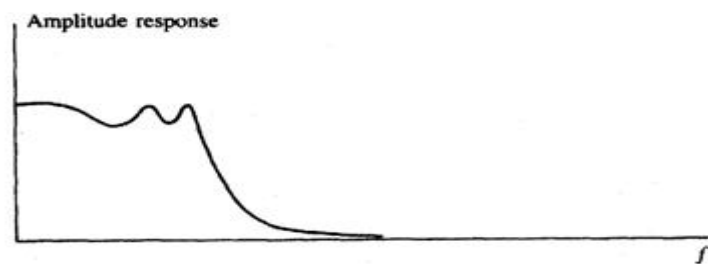


Figura 2.9 Respuesta Chebyshev para filtro pasa bajas.

La atenuación de un filtro Chebyshev siempre es mayor que la atenuación de un filtro de Butterworth del mismo orden.

El número de rizados en la banda de paso de un filtro pasa bajas de Chebyshev es igual a la mitad del orden del filtro tal y como se ve en la ecuación 2.21:

$$\text{número de rizados} = \frac{n}{2} \quad (2.21)$$

Si se tiene un filtro de orden 10, tendrá 5 rizados en la banda de paso; si el orden del filtro es 15, tendrá 7.5 rizados.

Aproximación inversa de Chebyshev

En aplicaciones en las que se requiere una respuesta plana en la banda de paso, así como una caída rápida en la región de transición, el diseñador puede utilizar la aproximación inversa de Chebyshev. Esta aproximación proporciona una respuesta plana en la banda de paso y una respuesta con rizado en la banda eliminada. La velocidad de caída en la región de transición es comparable a la del filtro Chebyshev.

La Figura 2.10 muestra la respuesta de un filtro pasa bajas inverso de Chebyshev:

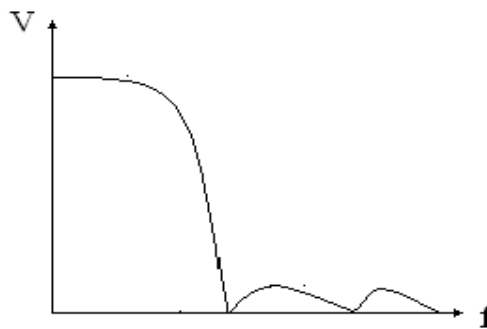


Figura 2.10 Respuesta inversa de Chebyshev de un filtro pasa bajas.

Un filtro inversor de Chebyshev tiene una banda de paso plana, una caída rápida en la región de transición y una banda eliminada con rizado.

Monotónica quiere decir que la banda eliminada no presenta rizado. Con las aproximaciones vistas hasta el momento, filtros Butterworth y de Chebyshev tienen bandas eliminadas monótonicas. La aproximación inversa de Chebyshev tiene una banda eliminada con rizado.

Cuando se especifica un filtro inverso de Chebyshev, debe especificarse la atenuación mínima aceptable en la banda eliminada ya que ésta puede presentar rizados que pueden alcanzar este valor.

Aproximación elíptica

Algunas aplicaciones necesitan en la región de transición, la caída más rápida posible. Si son aceptables una banda de paso y una banda eliminada con rizados, se puede elegir la aproximación elíptica. También conocido como filtro de Cauer, este filtro optimiza la región de transición a expensas de la banda de paso y la banda eliminada.

La Figura 2.11 muestra la respuesta de un filtro pasa bajas elíptico:

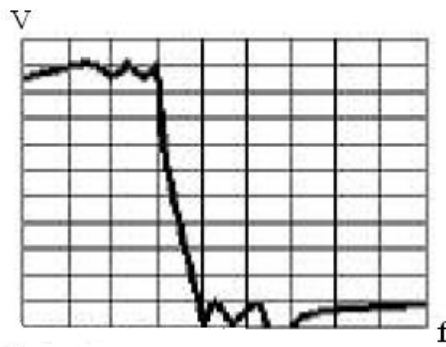


Figura 2.11 Respuesta elíptica de un filtro pasa bajas.

Aproximación de Bessel

La aproximación de Bessel tiene una banda de paso plana y una banda eliminada monótonica similares a las de la aproximación de Butterworth. Sin embargo, para un filtro del mismo orden, la pendiente de caída en la región de transición es mucho menor en un filtro Bessel que en un filtro de Butterworth.

La Figura 2.12 muestra la respuesta de un filtro pasa bajas de Bessel.

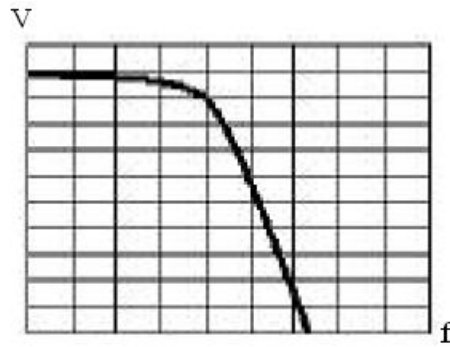


Figura 2.12 Respuesta Bessel de un filtro pasa bajas.

En la Figura 2.12 se puede observar que el filtro Bessel tiene una banda de paso plana, una caída relativamente lenta y una banda eliminada monotónica. Dado un conjunto de especificaciones para un filtro complejo, la aproximación de Bessel siempre proporcionara la menor caída de todas las aproximaciones. Dicho de otra manera: proporciona el filtro de mayor orden o lo que es lo mismo el circuito de mayor complejidad de todas las aproximaciones.

Filtros analógicos

Un filtro analógico se implementa mediante hardware y éste puede ser activo o pasivo.

Filtros pasivos

Los filtros pasivos son aquellos que se implementan únicamente con elementos pasivos, los cuales son resistencias, inductores y capacitores. Estos filtros se basan en la reactancia capacitiva y la reactancia inductiva que es una propiedad de los capacitores e inductores respectivamente.

La reactancia capacitiva e inductiva consiste en que el valor de resistencia en ohms del elemento cambia con las variaciones de frecuencia.

Un filtro LC pasa bajas de segundo orden tiene una frecuencia de resonancia y un factor Q (similar a un circuito resonante serie o paralelo).

Manteniendo constante la frecuencia de resonancia pero variando Q, podemos conseguir rizados en la banda de paso de los filtros de orden superior.

Frecuencia de resonancia y Q

La Figura 2.13 muestra un filtro LC pasa bajas. Se trata de un filtro de orden 2 porque contiene dos componentes reactivos: una bobina y un condensador. Un filtro LC de segundo orden tiene una frecuencia de resonancia definida en la ecuación 2.22 y un factor Q que se define en la ecuación 2.23:

$$f_0 = \frac{1}{2\pi\sqrt{LC}} \quad (2.22)$$

$$Q = \frac{R}{X_L} \quad (2.23)$$

Donde X_L se calcula para la frecuencia de resonancia.

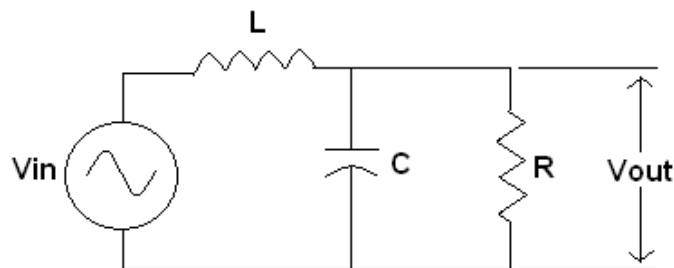


Figura 2.13 filtro LC pasa bajas.

En la frecuencia de resonancia, se forma un pico de ganancia. Cuanto mayor es Q, mayor es el incremento de la ganancia de tensión en la frecuencia de resonancia.

Factor de amortiguamiento

Para explicar la aparición de picos en la frecuencia de resonancia es utilizado el factor de amortiguamiento, que se define en la ecuación 2.24:

$$\alpha = \frac{1}{Q} \quad (2.24)$$

Para $Q=10$, el factor de amortiguamiento es:

$$\alpha = \frac{1}{10} = 0.1$$

De forma similar, un Q de 2 proporciona un factor $\alpha = 0.5$ y un Q igual a 0.707 da $\alpha = 1.414$. Cuanto mayor es el factor de amortiguamiento, menor es la amplitud de pico.

Respuestas de Butterworth y Chebyshev

La figura 2.14 resume el efecto de Q en un filtro de segundo orden. Como se indica en la figura 2.14, un Q de 0.707 produce una respuesta de Butterworth, es decir totalmente plana. Un Q de 2 produce un pico de rizado de 6dB y un Q de 10 produce un pico de rizado de 20 dB. En términos de amortiguamiento, la respuesta de Butterworth tiene una amortiguación crítica, mientras que las respuestas con rizado están subamortiguadas. Una respuesta Bessel (no mostrada) será sobre amortiguada porque su Q es igual a 0.577.

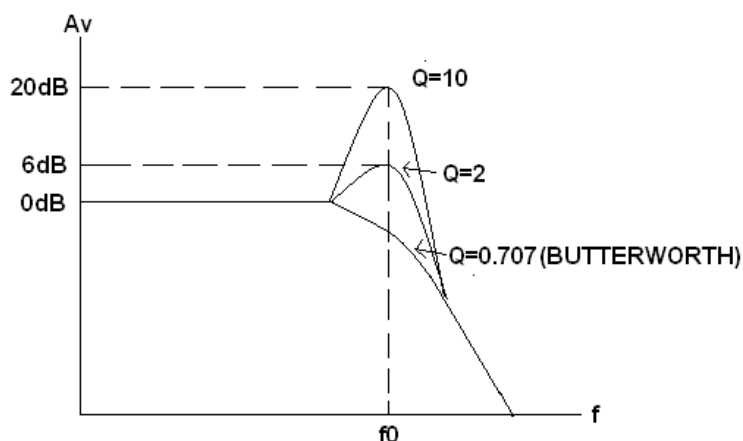


Figura 2.14 Efecto de Q en la respuesta de segundo orden (Malvino y Bates, 2007).

Filtros LC de orden superior

Los filtros de orden superior normalmente se construyen utilizando etapas de segundo orden conectadas en cascada. La Figura 2.15 muestra un filtro de orden superior formado por filtros de menor orden conectados en cascada.

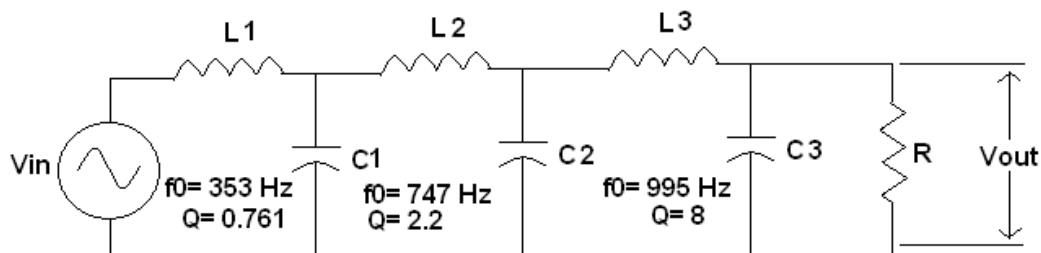


Figura 2.15 frecuencias de resonancia y factores Q en un filtro de orden superior.

Filtros activos

Etapas de primer orden: las etapas de primer orden o de filtro activo de 1 polo sólo tienen un condensador. Por tanto, sólo pueden producir una respuesta pasa bajas o pasa altas. Los filtros pasa banda y de banda eliminada pueden implementarse sólo cuando n es mayor que 1.

Etapas pasa bajas

La Figura 2.16 muestra algunas formas de construir un filtro activo pasa bajas de primer orden. El primer circuito no es más que un circuito RC de retardo y un seguidor de tensión. La ganancia de tensión está dada por la ecuación 2.25:

$$A_v=1 \quad (2.25)$$

La frecuencia de corte a 3 dB está dada por la ecuación 2.26:

$$f_c = \frac{1}{2\pi R_1 C_1} \quad (2.26)$$

Cuando la frecuencia aumenta por encima de la frecuencia de corte, la reactancia capacitiva disminuye y reduce la tensión de la entrada no inversora. Dado que el circuito de retardo $R_1 C_1$ está fuera del bucle de realimentación, la tensión de salida disminuye. Cuando la frecuencia se aproxima al infinito, el condensador se comporta como un cortocircuito y la tensión de entrada es cero. El segundo circuito de la Figura 2.16 muestra otro filtro pasa bajas de primer orden no inversor. Aunque tiene dos resistencias adicionales, tiene la ventaja de la ganancia de tensión. La ganancia de tensión por debajo de la frecuencia de corte está dada por la ecuación 2.27:

$$A_v = \frac{R_2}{R_1} + 1 \quad (2.27)$$

La frecuencia de corte viene dada por la ecuación 2.28:

$$f_c = \frac{1}{2\pi R_3 C_1} \quad (2.28)$$

Por encima de la frecuencia de corte, el circuito de retardo reduce la tensión de la entrada no inversora. Dado que el circuito de retardo $R_3 C_1$ está fuera del lazo de realimentación, la tensión de salida decrece con una pendiente de 20 dB por década.

El tercer circuito de la Figura 2.16 muestra un filtro pasa bajas de primer orden inversor y sus ecuaciones. A bajas frecuencias, el condensador es un circuito abierto y el circuito se comporta como un amplificador inversor con una ganancia de tensión que está dada por la ecuación de 2.29:

$$A_v = \frac{-R_2}{R_1} \quad (2.29)$$

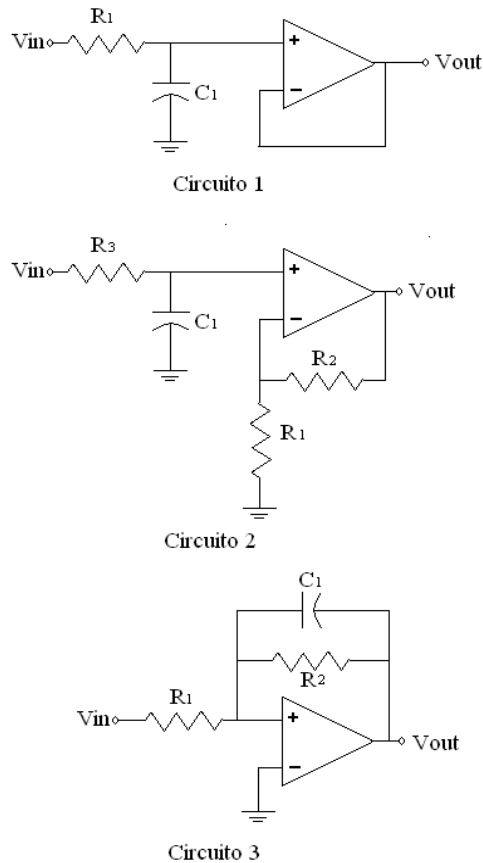


Figura 2.16 Etapas pasa bajas de primer orden.

Quando la frecuencia aumenta, la reactancia capacitiva disminuye y se reduce la impedancia de la rama de realimentación. Esto implica una menor ganancia de tensión. Cuando la frecuencia tiende al infinito, el condensador se convierte en un corto circuito y no hay ganancia de tensión. En la figura 2.16, en el circuito 3, la frecuencia de corte está dada por la ecuación 2.30:

$$f_c = \frac{1}{2\pi R_2 C_1} \quad (2.30)$$

No existe ninguna otra forma de implementar un filtro pasa bajas de primer orden. En otras palabras, los circuitos mostrados en la Figura 2.16 son las tres únicas configuraciones disponibles para una etapa pasa bajas de un filtro activo.

Etapa pasa altas

La Figura 2.17a muestra la forma más simple de construir un filtro activo pasa altas de primer orden. La ganancia de tensión está dada por la ecuación 2.25:

$$A_v = 1 \quad (2.25)$$

La frecuencia de corte a 3dB está dada por la ecuación 2.26:

$$f_c = \frac{1}{2\pi R_1 C_1} \quad (2.26)$$

Cuando la frecuencia disminuye por debajo de la frecuencia de corte, la reactancia capacitiva aumenta y la tensión de la entrada no inversora disminuye. Puesto que el circuito $R_1 C_1$ está fuera de realimentación, la tensión de salida disminuye. Cuando la frecuencia se aproxima a cero, el condensador se comporta como un circuito abierto y la tensión de entrada es cero.

La Figura 2.17b muestra otro filtro pasa altas de primer orden no inversor, la ganancia de tensión por encima de la frecuencia de corte está dada por la ecuación 2.27:

$$A_v = \frac{R_2}{R_1} + 1 \quad (2.27)$$

La frecuencia de corte a 3 dB es mostrada en la ecuación 2.28:

$$f_c = \frac{1}{2\pi R_3 C_1} \quad (2.28)$$

Por debajo de la frecuencia de corte, el circuito RC reduce la tensión de la entrada no inversora. Dado que el circuito de retardo $R_3 C_1$ está fuera del lazo de realimentación, la tensión de salida disminuye con una pendiente de 20 dB por década.

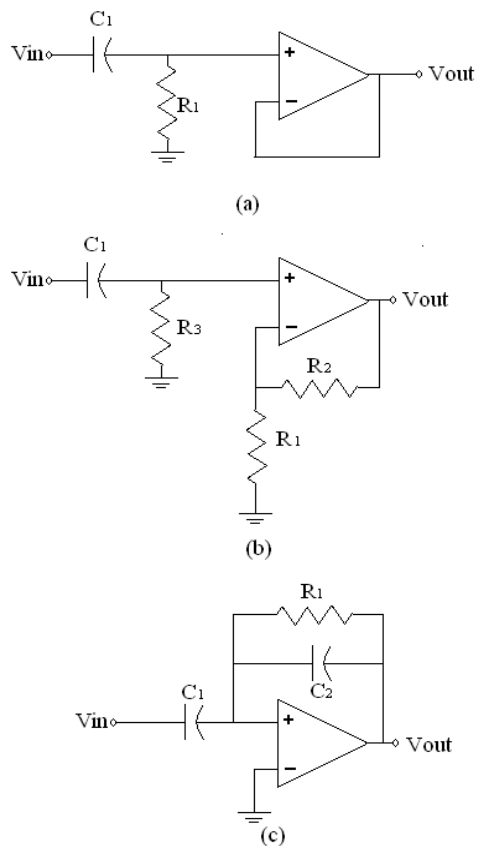


Figura 2.17 Etapas pasa altas de primer orden.

La Figura 2.17c muestra otro filtro pasa altas de primer orden. A altas frecuencias, el circuito se comporta como un amplificador inversor con una ganancia de tensión como el mostrado en la ecuación 2.31:

$$A_v = \frac{-X_{C2}}{X_{C1}} = \frac{-C_1}{C_2} \quad (2.31)$$

A medida que la frecuencia disminuye, la reactancia capacitiva aumenta y hace que la señal de entrada y la realimentación disminuyan, lo que implica una ganancia de tensión menor. A medida que la frecuencia se aproxima a cero, los condensadores se comportan como circuitos abiertos y no hay señal de entrada. Como se muestra en la Figura 3.17c, la frecuencia de corte a 3 dB está dada por la ecuación 2.26:

$$f_c = \frac{1}{2\pi R_1 C_1} \quad (2.26)$$

Filtro pasa banda con realimentación múltiple

Un filtro pasa banda tiene una frecuencia central y un ancho de banda, sus ecuaciones básicas son las siguientes:

$$BW = f_2 - f_1 \quad (2.12)$$

$$f_0 = \sqrt{f_1 f_2} \quad (2.13)$$

$$Q = \frac{f_0}{BW} \quad (2.14)$$

Cuando Q es menor que 1, el filtro presenta una respuesta de banda ancha. En este caso, un filtro pasa banda normalmente se construye conectando en cascada una etapa pasa bajas con una etapa pasa altas. Cuando Q es mayor que 1, el filtro presenta una respuesta de banda estrecha y se utiliza una aproximación diferente.

Filtros pasa bajas de segundo orden y ganancia unidad con VCVS

Las etapas de segundo orden o de 2 polos son las más comunes porque son fáciles de construir y de analizar. Los filtros de orden superior normalmente se construyen conectando en cascada etapas de segundo orden. Cada una de las etapas de segundo orden tiene una frecuencia de resonancia y un factor Q que permiten determinar la cantidad de picos.

Los filtros pasa bajas de Sallen-Key (en honor a sus inventores), también denominados filtros VCVS, porque el amplificador operacional se utiliza como una fuente de tensión controlada con tensión. Los circuitos pasa bajas con VCVS pueden implementar tres de las aproximaciones básicas: Butterworth, Chebyshev y Bessel.

Implementación del circuito:

La Figura 2.18 muestra un filtro pasa bajas de segundo orden de Sallen-Key. Las dos resistencias tienen el mismo valor, pero los dos condensadores son diferentes. Hay un circuito de retardo en la entrada no inversora y existe un camino de realimentación a través del condensador C_2 . Para bajas frecuencias, ambos condensadores se comportan como circuitos abiertos y el circuito tiene ganancia unitaria porque el amplificador operacional está conectado como un seguidor emisor.

Cuando la frecuencia aumenta, la impedancia de C_1 disminuye y la tensión de la entrada inversora disminuye. Al mismo tiempo, el condensador C_2 está realimentando una señal que está en fase con la señal de entrada. Dado que la señal de realimentación se suma a la señal de fuente, la realimentación es positiva. Como resultado, la disminución en la tensión de entrada no inversora causada por C_1 no es tan grande como lo sería si no existiera la realimentación positiva.

Cuanto más grande sea C_2 . Con respecto a C_1 , mayor es la realimentación positiva; esto es equivalente a incrementar el Q del circuito. Si C_2 es lo suficientemente grande como para hacer que Q sea mayor que 0.707, en la respuesta en frecuencia aparecerán picos.

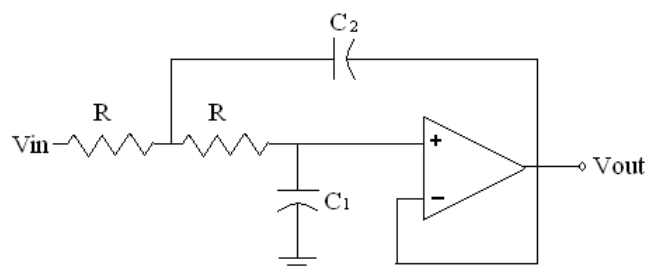


Figura 2.18 Etapa de segundo orden con VCVS para Butterworth y Bessel.

Frecuencia del polo

Las ecuaciones 2.32 y 2.33 son las ecuaciones de la figura 2.18.

$$Q = 0.5 \sqrt{\frac{C_2}{C_1}} \quad (2.32)$$

$$f_p = \frac{1}{2\pi R \sqrt{C_1 C_2}} \quad (2.33)$$

La frecuencia de polo (f_p) es una frecuencia especial utilizada en el diseño de filtros activos.

Respuestas de Butterworth y Bessel

En el circuito de la figura 2.18, se calcula Q y f_p . Si $Q=0.707$, se obtiene una respuesta Butterworth y un valor de K_c de 1. Si $Q=0.577$, tenemos una respuesta de Bessel y un valor de K_c igual a 0.786. A continuación se puede calcular la frecuencia de corte aplicando la ecuación 2.34:

$$f_c = K_c f_p \quad (2.34)$$

En los filtros de Butterworth y de Bessel, la frecuencia de corte siempre es la frecuencia a la que la atenuación es igual a 3dB.

Filtros de orden superior

El método estándar de construcción de los filtros de orden superior consiste en conectar en cascada etapas de primer orden y segundo orden. Si el orden es par, necesitamos conectar en cascada sólo etapas de segundo orden. Cuando el orden es impar, hay que conectar en cascada etapas de segundo orden y una única etapa de primer orden.

Filtros de Butterworth

Cuando las etapas de filtro están conectadas en cascada, podemos sumar la atenuación en decibelios de cada etapa para obtener la atenuación total. Por ejemplo, en la Figura 2.19a muestra dos etapas de segundo orden

conectadas en cascada. Si cada una tiene un Q de 0.707 y una frecuencia de polo de 1 kHz, entonces cada etapa tiene una respuesta de Butterworth con una atenuación de 3 dB a 1 kHz. Aunque cada etapa tiene una respuesta de Butterworth, la respuesta global no es una respuesta de Butterworth porque decae en la frecuencia de polo, como se indica en la Figura 2.19b. Dado que cada etapa tiene una atenuación de 3 dB a la frecuencia de corte de 1 kHz, la atenuación global es de 6 dB a 1 kHz.

Para obtener una respuesta de Butterworth, las frecuencias de polo tienen que seguir en 1 kHz, pero los factores Q de las etapas tienen que sucederse por encima y por debajo de 0.707. La Figura 2.19c muestra cómo obtener una respuesta de Butterworth para el filtro global. La primera etapa tiene un $Q = 0.54$ y la segunda etapa tiene un $Q = 1.31$. El pico en la segunda etapa compensa la caída de la primera etapa para proporcionar una atenuación de 3 dB en 1 kHz. Se puede observar que la respuesta en la banda de paso es maximalmente plana con estos valores de Q (Malvino y Bates, 2007).

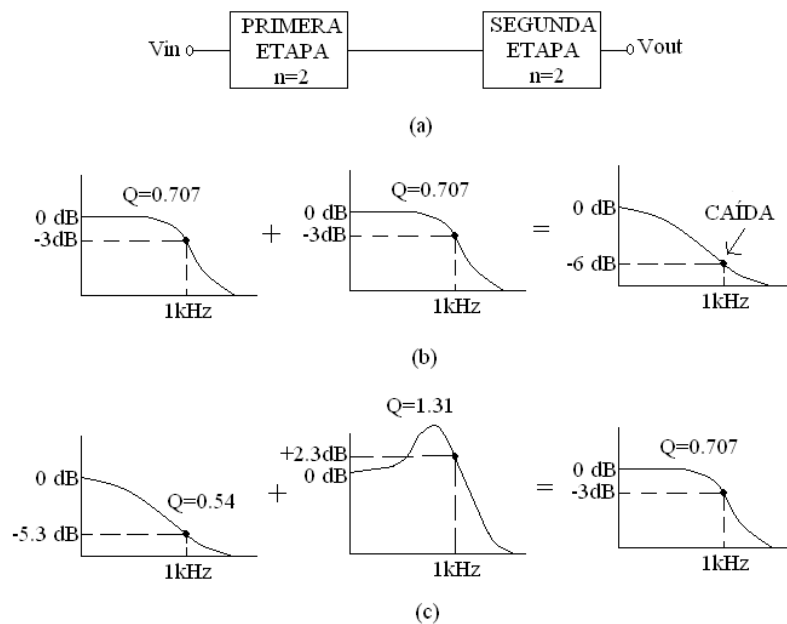


Figura 2.19 Conexión en cascada de filtros de orden superior. (a) Dos etapas conectadas en cascada. (b) Etapas iguales producen una caída en la frecuencia de corte. (c) Etapas de bajo Q y de alto Q se compensan para generar una respuesta de Butterworth (Malvino y Bates, 2007).

En el Cuadro 2.1 muestra la sucesión de valores de Q de las etapas utilizadas en filtros de Butterworth de orden superior.

Cuadro 2.1 Factores Q encadenados para filtros pasa bajas de Butterworth (Malvino y Bates, 2007).

Orden	Etapas 1	Etapas 2	Etapas 3	Etapas 4	Etapas 5
2	0.707				
4	0.54	1.31			
6	0.52	1.93	0.707		
8	0.51	2.56	0.6	0.9	
10	0.51	3.2	0.56	1.1	0.707

2.2.2 Sistemas discretos

Los sistemas discretos se caracterizan por que sus entradas y salidas son secuencias de valores; y una secuencia se puede definir como cualquier conjunto numerado de elementos, es decir, haciendo corresponder a cada número entero un elemento de un conjunto se obtiene una secuencia en dicho conjunto.

La forma general de representar una secuencia es $\{x_k\}$ en donde k , entero asociado al elemento, es el índice del mismo. Este índice podrá ser positivo o negativo.

$$\{\dots, x_{-2}, x_{-1}, x_0, x_1, x_2, \dots\} \Leftrightarrow \{x_k\}$$

Figura 2.20 Representación de un sistema discreto.

El índice 0 es el origen de la secuencia.

Un sistema discreto es un algoritmo que permite transformar una secuencia en otra.

$$\{y_k\} = f\{x_k\}$$

Figura 2.21 Transformación de secuencias.

2.2.3 Sistema estático

Un sistema discreto es estático cuando el elemento de la secuencia de salida de un cierto índice depende únicamente del elemento de la secuencia de entrada del mismo índice.

2.2.4 Sistema dinámico

Es cuando la secuencia de salida de un cierto índice depende de las secuencias de entrada y salida de órdenes distintos al suyo.

2.2.5 Sistema discreto causal

Un sistema discreto es causal si el valor de un elemento de la secuencia de salida depende únicamente de la secuencia de entrada y salida de menor o igual índice. Los sistemas discretos que no verifiquen esta propiedad se denominan no causales.

2.2.6 Muestreo y cuantización

El muestreo de señales consiste en la construcción de secuencias a partir de señales continuas.

Muestreo ideal: el muestreo ideal describe una señal muestreada como una suma ponderada de impulsos, en la que los factores de ponderación son iguales a los valores de la señal analógica en los instantes donde ocurren los impulsos. Una señal muestreada idealmente $x_I(t)$ puede considerarse como el producto de una señal analógica $x(t)$ y un tren periódico de impulsos $i(t)$, como se ilustra en la Figura 2.22.

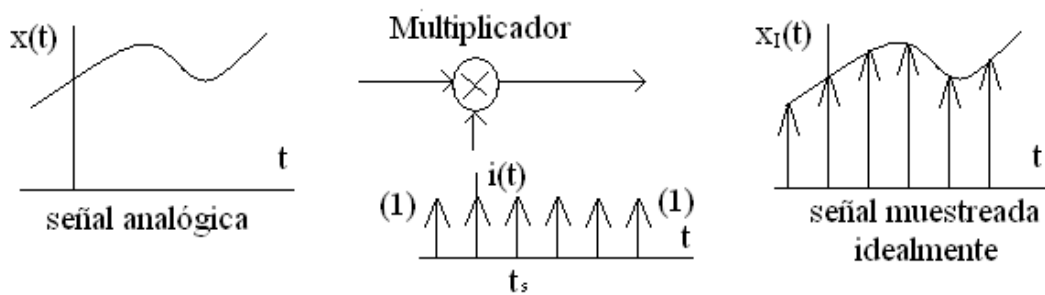


Figura 2.22 Operación ideal de muestreo (Ambardar, 2002).

Ancho de banda de Nyquist

Nyquist afirmó que para poder muestrear una señal es necesario utilizar una frecuencia por lo menos dos veces mayor que la máxima frecuencia de la señal muestreada, de esta manera si la frecuencia máxima de una señal es B , podemos muestrearla con una frecuencia mayor o igual a $2B$, de esta manera se puede evitar el fenómeno de alias.

Fenómeno de Alias

Cuando no se respeta el teorema de muestreo y se muestrea una señal a una frecuencia menor que la frecuencia de Nyquist se obtiene una señal muestreada que al volverla a convertir a una señal analógica y compararla con la señal original, en la señal muestreada componentes de la señal analógica original de altas frecuencias aparecen (por el alias) a bajas frecuencias en la señal muestreada.

Desafortunadamente existen muchas señales que pueden tener un ancho de banda infinito por lo que es imposible muestrearlas con una frecuencia dos veces mayor que la frecuencia máxima, por esta razón se utilizan filtros pasa-bajas para limitar el ancho de banda de la señal y de esta manera poder aplicar el teorema de muestreo.

Cuantización

La importancia de las señales digitales parte de la proliferación de computadoras digitales de alta velocidad para el procesamiento de señales. Debido a las limitaciones de memoria finita de tales máquinas, sólo es posible procesar secuencias finitas de datos. Por tanto, no sólo es necesario muestrear la señal analógica en el tiempo sino también se requiere cuantizar (redondear o truncar) las amplitudes de las señales para reducirlas a un conjunto finito de valores. Puesto que la cuantización sólo afecta a la amplitud de la señal, es posible cuantizar señales analógicas y de tiempo discreto. Las señales cuantizadas de tiempo discreto se conocen como señales digitales.

Consideraciones prácticas relacionadas con el ADC

El convertidor analógico digital (ADC) para la conversión de señales analógicas en señales digitales consiste en un circuito de muestreo y retención, seguido de un cuantizador y un codificador. La Figura 2.23 muestra el diagrama de bloques de un circuito de muestreo y retención.

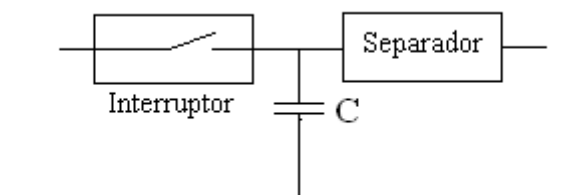


Figura 2.23 Diagrama en bloques de un sistema de muestreo y retención.

Para poder utilizar un ADC de manera adecuada es necesario conocer sus tiempos de retardo:

Tiempo de apertura T_A : este es el tiempo durante el cual se mide la señal.

Tiempo de adquisición T_H : para conmutar del modo de retención al modo de muestreo.

Tiempo de conversión T_C : tiempo del cuantizador.

Un tiempo de apertura finito limita tanto la exactitud con la que puede medirse la señal como la máxima frecuencia que puede manejar el ADC. A la tasa de muestreo máxima que puede ser empleada por un ADC (con un circuito de muestreo y retención) es gobernada por el tiempo de apertura T_A , el tiempo de retención T_H , del circuito de muestreo y retención, así como el tiempo de conversión T_C del cuantizador, esto se muestra en la ecuación 2.35:

$$S \leq \frac{1}{T_A + T_H + T_C} \quad (2.35)$$

Naturalmente, esta tasa de muestreo tiene que ser mayor a la tasa de Nyquist.

Consideraciones sobre el filtro de anti-alias

La finalidad del filtro de anti-alias es limitar la banda de la señal de entrada. Sin embargo en la práctica no es posible diseñar filtros con características de pared, así que es inevitable tener cierto grado de fenómeno de alias. El diseño de filtro anti-alias debe asegurar que los efectos del fenómeno de alias se mantienen reducidos. Una manera de hacer esto es atenuar los componentes por encima de la frecuencia de Nyquist a un nivel que no pueda ser detectado por el ADC. Por tanto, la selección de frecuencia de muestreo S es dictada no sólo por la mayor frecuencia de interés, sino también por la resolución del ADC. Si el nivel de fenómeno de alias es ΔV y el nivel máximo de la banda de paso es V , se requiere un filtro con una atenuación en la banda de supresión $A_S > 20 \log(V/\Delta V)$ dB. Si el nivel pico de la señal es A y el límite de la banda de paso se define como la frecuencia de potencia media (o de 3dB), y se escoge ΔV como el error rms de cuantización para un cuantizador de B bits, entonces se tiene la ecuación 2.36:

$$A_S > 20 \log \frac{\text{nivel rms máximo en la banda de paso}}{\text{nivel rms mínimo en la banda de supresión}} = \frac{A/\sqrt{2}}{\Delta/\sqrt{12}} = \frac{A/\sqrt{2}}{A/(2^B \sqrt{12})} = 20 \log(2^B \sqrt{6}) \quad \text{dB} \quad (2.36)$$

2.2.7 Análisis de Fourier

Las señales tienen diferentes componentes de frecuencia, la determinación del contenido en frecuencias de muchas señales se puede obtener fácilmente disponiendo de unas buenas tablas de integrales. Cualquier señal periódica se puede expresar como una suma de funciones sinusoidales denominada serie de Fourier, la cual se ilustra en la ecuación 2.37:

$$x(t) = \frac{A_0}{2} + \sum_{n=1}^{\infty} [A_n \cos(2\pi n f_0 t) + B_n \sin(2\pi n f_0 t)] \quad (2.37)$$

Donde f_0 es la inversa del periodo de la señal ($f_0=1/T$). La frecuencia f_0 se denomina frecuencia o armónico fundamental, y los múltiplos de f_0 armónicos. Por tanto, una señal periódica con periodo T estará compuesta por la frecuencia fundamental $f_0=1/T$, más los múltiplos enteros de dicha frecuencia. Si A_0 es distinto de 0, la señal $x(t)$ tendrá componente dc o continua.

Los valores de los coeficientes del desarrollo en serie de Fourier se calculan mediante las siguientes expresiones:

$$A_0 = \frac{2}{T} \int_0^T x(t) dt \quad (2.38)$$

$$A_n = \frac{2}{T} \int_0^T x(t) \cos(2\pi n f_0 t) dt \quad (2.39)$$

$$B_n = \frac{2}{T} \int_0^T x(t) \sin(2\pi n f_0 t) dt \quad (2.40)$$

Este tipo de representación, denominada representación seno-coseno, es la más sencilla de calcular, si bien, presenta el problema de tener dos componentes para cada frecuencia. Otra representación alternativa a la anterior, denominada representación módulo-fase, adopta la siguiente forma que se observa en la ecuación 2.41:

$$x(t) = \frac{C_0}{2} + \sum_{n=1}^{\infty} C_n \cos(2\pi n f_0 t + \theta_n) \quad (2.41)$$

Que se relaciona con la representación seno-coseno mediante las expresiones siguientes:

$$C_0 = A_0 \quad (2.42)$$

$$C_n = \sqrt{A_n^2 + B_n^2} \quad (2.43)$$

$$\theta_n = \tan^{-1} \left(\frac{-B_n}{A_n} \right) \quad (2.44)$$

Transformada de Fourier para señales no periódicas

El espectro de una señal periódica, consiste en un conjunto de componentes en frecuencias discretas a la frecuencia fundamental y sus armónicos. Para una señal no periódica, su espectro consiste en un continuo de frecuencias. Este espectro se puede obtener mediante la transformada de Fourier. Para una señal $x(t)$, con espectro $x(f)$, se verifican las siguientes expresiones:

$$x(t) = \int_{-\infty}^{\infty} x(f) e^{j2\pi f t} dt \quad (2.45)$$

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi f t} dt \quad (2.46)$$

Donde $j = \sqrt{-1}$. La aplicación del número imaginario en las expresiones anteriores es por razones de comodidad. La componente imaginaria tiene una interpretación física relacionada con la fase de la forma de onda (Stalling, 2000).

La Transformada Discreta de Fourier (DFT)

La transformada discreta de Fourier (DFT) con N puntos, $X_{\text{DFT}}[k]$ de una señal con N muestras $x[n]$ y la transformada inversa discreta de Fourier (IDFT), que transforma $X_{\text{DFT}}[k]$ en $x[n]$, se definen como se muestra en la ecuaciones 2.47 y 2.48:

$$X_{DFT}[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N}, \quad k=0,1,2,\dots,N-1 \quad (2.47)$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X_{DFT}[k] e^{j2\pi nk/N}, \quad n=0,1,2,\dots,N-1 \quad (2.48)$$

Cada relación es un conjunto de N ecuaciones. Cada muestra de la DFT se obtiene a través de la suma ponderada de todas las muestras de $x[n]$.

La Transformada Rápida de Fourier (FFT)

La importancia de la DFT se debe a que es apropiada para el cálculo rápido y eficiente mediante el uso de un grupo de algoritmos conocidos como la transformada rápida de Fourier. Los algoritmos rápidos reducen el problema de calcular una DFT de N puntos al del cálculo de una DFT más pequeña. Existen dos algoritmos de la FFT:

Algoritmo FFT de reducción de frecuencia: el algoritmo FFT de reducción en frecuencia (DIF) comienza reduciendo la transformada de N puntos en cada etapa a dos transformadas de N/2 puntos, después a cuatro transformadas de N/4 puntos, y así sucesivamente hasta llegar a N transformadas de 1 punto, que son las que corresponden a la DFT real. Con la secuencia de entrada en orden natural, los cálculos pueden realizarse en el lugar, pero el orden de los bits de la DFT resultará invertido, por lo que será necesario reordenarlos. La Figura 2.24 muestra la estructura de mariposa de este algoritmo:

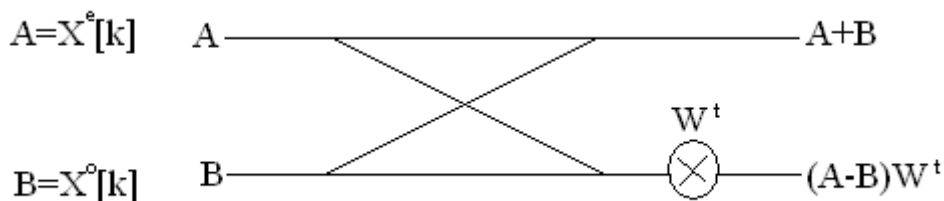


Figura 2.24 Mariposa común para el algoritmo FFT de reducción en frecuencia.

Los factores W^t , denominados factores de giro, aparecen sólo en las esquinas inferiores de las alas de la mariposa en cada etapa. Sus exponentes t tienen un orden definido, descrito de la siguiente manera para un algoritmo fft de $N=2^m$ puntos con m etapas:

1. Número de P de factores de giro distintos W^t en la i -ésima etapa: $P=2^{m-i}$.
2. Valores de t en los factores de giro W^t : $t=2^{i-1}Q$ con $Q=0,1,2,\dots,P-1$.

Algoritmo FFT de reducción en el tiempo: en el algoritmo FFT de reducción en el tiempo (DIT) se comienza con N transformadas de un punto, se combina en transformadas de dos puntos los pares adyacentes en cada etapa sucesiva, luego en transformadas de cuatro puntos y así sucesivamente, hasta llegar a la DFT de N puntos. Con el orden de los bits invertido en la secuencia de entrada, es posible realizar los cálculos en el lugar y la DFT se obtiene en orden natural. En la Figura 2.25 se muestra la estructura de esta mariposa.

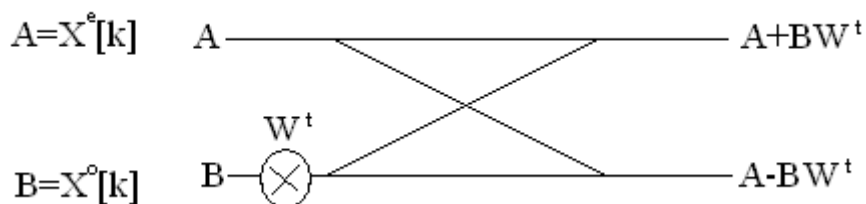


Figura 2.25 Mariposa característica del algoritmo FFT de reducción en el tiempo.

Al igual que con el algoritmo de reducción en frecuencia, los factores de giro W^t en cada etapa aparecen sólo en las alas inferiores de la mariposa. Los exponentes de t también tienen un orden definido (y casi similar) descrito por:

1. Número P de factores de giro distintos W^t en la i -ésima etapa: $P=2^{i-1}$.
2. Valores de t en los factores de giro W^t : $t=2^{m-i}Q$, donde $Q=0,1,2,\dots,P-1$ (Ambardar, 2002)

3. METODOLOGÍA

Primeramente se diseñó un circuito de potencia que se adaptará al material disponible, en este caso electroválvulas de 24VAC.

3.1 Etapa de potencia

La etapa de potencia fue hecha a base de triacs, primeramente dependiendo de las señales de entrada, el PIC producirá pulsos digitales a cada electroválvula, cada salida digital del PIC contiene una etapa de optoacoplamiento, con el fin de que el PIC no esté en contacto directo con los circuitos de potencia. Después de la etapa de optoacoplamiento, la señal es dirigida a un triac el cual es un dispositivo que al recibir una señal en su compuerta (gate) cierra el circuito de la electroválvula, alimentándola con un voltaje de 24VAC. El diagrama de bloques de la etapa de potencia se muestra en la Figura 3.1.

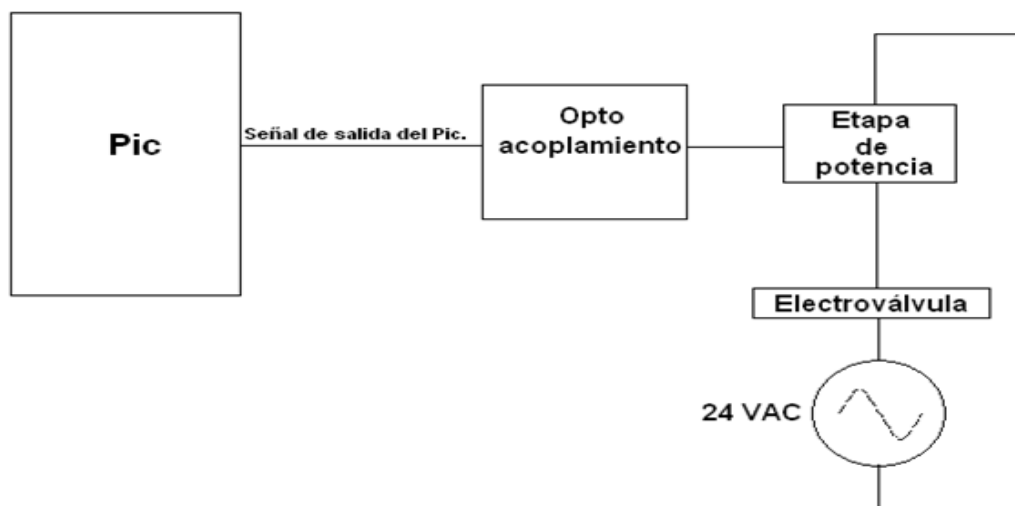


Figura 3.1 Diagrama de bloque de la etapa de potencia.

Cabe señalar que el diagrama de la Figura 3.1 muestra la conexión de una sola electroválvula, para implementar más electroválvulas es necesario realizar esta configuración para cada una.

Las primeras pruebas se realizaron con 10 electroválvulas, una vez armado el circuito, se programaron en el PIC diferentes secuencias para probar tanto la etapa de potencia como el correcto funcionamiento de las electroválvulas, y en las últimas pruebas realizadas se incorporó al programa del PIC el ADC, el cual detecta la variación de voltaje que provoca un potenciómetro y en función del valor de voltaje leído varía la velocidad de las secuencias de las electroválvulas.

Una vez de que la etapa de potencia desplegó buenos resultados, se hizo el diseño del PCB para maquinar la tarjeta.

3.2 Acondicionamiento de señal

Finalizado el diseño de la etapa de potencia, se comenzó a trabajar con la etapa del acondicionamiento de la señal.

Antes de comenzar a trabajar con el análisis de Fourier, se deben trabajar otros módulos los cuales son necesarios para poder llevar a cabo un buen procesamiento de señales, a continuación, la Figura 3.2 muestra un diagrama con los módulos:



Figura 3.2 Etapas del procesamiento de señales.

Al recibir la señal de audio, ésta se tiene que acondicionar, debido a que este tipo de señales generalmente son el orden de milivolts, se tiene que amplificar y evitar que esta señal sobrepase los 5 volts o sea menor que cero volts, estos puntos constituyen el acondicionamiento de señal.

En el filtrado, se tienen que extraer las señales indeseadas que obtiene nuestra señal de audio debido al ruido del medio en el que esté operando el circuito, el filtrado puede ser analógico o digital cualquiera de los dos es efectivo si está bien diseñado. Otra función del filtro es limitar el ancho de la señal para poderla muestrear según el teorema de muestreo, el cual establece que las señales deben ser muestreadas con una frecuencia mínima mayor o igual al doble del valor de la máxima frecuencia contenida en la señal.

Una vez que la señal ha sido acondicionada y filtrada, se utiliza el análisis de Fourier para obtener los armónicos de la señal, después se mandan los resultados obtenidos a la PC y una interfaz gráfica desarrollada en C# se encarga de graficar estos datos.

El acondicionamiento de señal consta de tres etapas: amplificación de la señal de audio, filtrado y voltaje de offset.

La entrada analógica del microcontrolador únicamente acepta señales de 0V a 5V y como la señal de audio tiene valores negativos, se le tiene que introducir un voltaje de offset de 2.5V, de esta manera el cero de la señal será de 2.5V y los voltajes que eran negativos estarán en el rango de 2.5V a 0V y los voltajes positivos oscilaran de 2.5V a 5V en la Figura 3.3 se muestra un ejemplo de cómo tiene que ser la señal de salida del circuito acondicionador de señal.

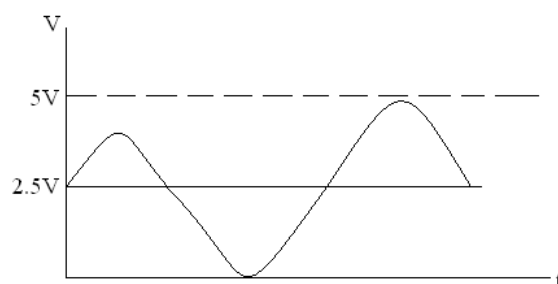


Figura 3.3 Señal acondicionada.

Amplificación de la señal de audio.

La señal de audio se obtiene a través de un cable plug que va de la computadora al circuito. Cada computadora nos brinda una señal de salida diferente en cuanto amplitud, hay computadoras que tienen más volumen que otras y por esta razón el circuito de acondicionamiento de señal se tiene que adaptar.

Primeramente se midió la señal de audio de la computadora, y en base a esa medida se diseñó la etapa de amplificación de la señal, la señal da voltajes picos de hasta 1.5V, de cualquier forma si el circuito se conecta a una computadora que dé más amplitud simplemente se baja el volumen hasta igualar este valor, y una computadora que no alcance el nivel de 1.5V, no excederá el valor límite, así, de esta manera se puede hacer pruebas con cualquier máquina. La Figura 3.4 muestra el resultado del osciloscopio para la señal de audio.

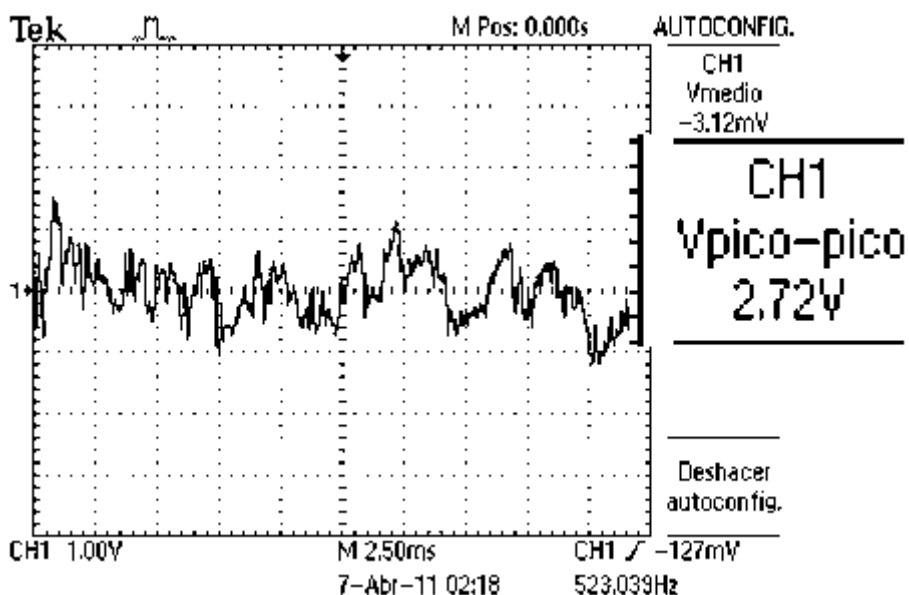


Figura 3.4 señal de audio de la computadora sin amplificar.

Como se puede observar en la Figura 3.4, la señal de audio sin amplificar tiene un voltaje pico-pico de casi 3V, entonces el voltaje pico sería de

1.5V. Para saber la ganancia que debe brindar el amplificador, se toma el valor máximo de amplitud, en este caso 1.5V, el voltaje offset es de 2.5V y la señal oscilara entre 0V y 5V, como la referencia es 2.5V entonces la amplitud pico de la señal tiene que ser de 2.5V, la ganancia se calcula al dividir 2.5V/1.5V, lo que nos da un resultado de 1.67V.

De la ecuación (2.2) del amplificador no inversor, tenemos:

$$A_v = \frac{R_f}{R_1} + 1$$

Nuestra ganancia A_v es igual a 1.6666667, si proponemos que $R_1 = 10k\Omega$ obtenemos que $R_f = 6.67k\Omega$, el circuito con la configuración de amplificador no inversor quedara de la siguiente manera:

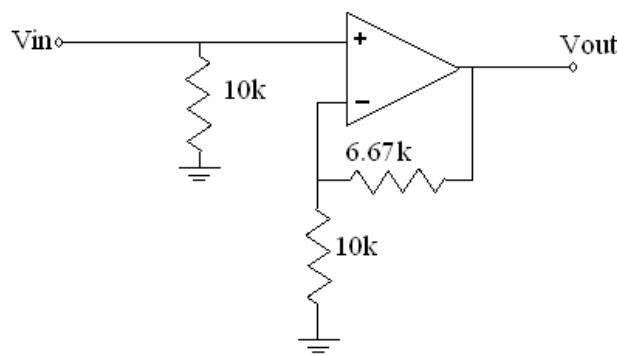


Figura 3.5 Amplificación de la señal de audio.

Filtrado

Para poder limitar el ancho de banda de la señal es necesario un filtro pasa bajas y para proponer una frecuencia de corte del filtro y su orden, es necesario medir la frecuencia de muestreo del microcontrolador. Para poder realizar el muestreo, se implementó en el software del microcontrolador la interrupción del ADC, la cual interrumpe al programa cuando el ADC ha terminado de realizar la conversión analógica-digital del dato, de esta manera nos aseguramos de que el dato que estamos procesando ha sido convertido correctamente, para medir la frecuencia de muestreo, hay que medir el tiempo que hay entre las interrupciones del ADC, para realizar esta tarea, cuando se

ejecuta el código de la interrupción, se manda un pulso a una salida digital del microcontrolador, esto genera un tren de pulsos y con el osciloscopio se mide la frecuencia de esta señal, La Figura 3.6 muestra el resultado de este procedimiento.

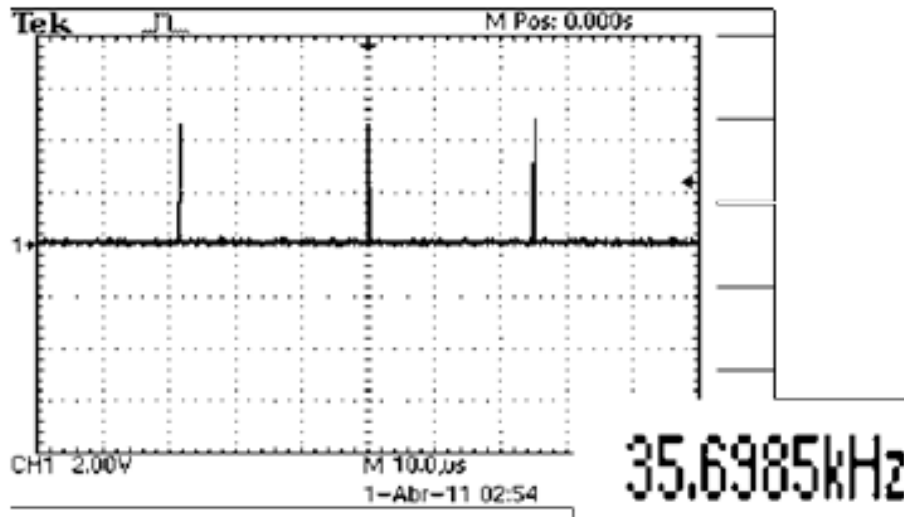


Figura 3.6. Medición de la frecuencia de muestreo.

En la Figura 3.6 se observa la pantalla del osciloscopio, el cual midió una frecuencia de muestreo de 35698Hz, conociendo este dato ahora sabemos que no es posible utilizar como frecuencia de corte 20kHz que es la máxima frecuencia que tiene el sonido audible para el ser humano, ya que con la capacidad del microcontrolador no se puede muestrear al doble de esta frecuencia. Aunque la frecuencia de muestreo del microcontrolador hubiera sido 40kHz, tampoco sería correcto limitar la señal con una frecuencia de corte de 20kHz, a menos que contáramos con una respuesta del filtro ideal como el de la figura 2.3. Como la respuesta real del filtro Butterworth es como la de la figura 2.8 hay que considerar esa rampa de bajada, ya que aunque la magnitud de la señal está disminuyendo, se tiene que llegar a un límite de reducción antes de que el ADC sea incapaz de detectar estas señales, la resolución del ADC determina hasta qué grado de disminución de la señal, el ADC ya no es capaz de detectar a las señales. De esta manera si consideramos que el ADC tiene una resolución de 10 bits, al aplicar la ecuación 2.36 para filtros Butterworth:

$$A_s = 20 \log(2^B \sqrt{6}) = 20 \log(2^{10} \sqrt{6}) = 68dB.$$

El resultado anterior quiere decir que cuando el filtro logra atenuar 68dB la señal, el ADC de 10 bits ya no será capaz de detectar los voltajes atenuados a ésta magnitud.

Si se utiliza el filtro Butterworth de decimo orden, con una frecuencia de corte de 9kHz y utilizando las ecuaciones 3.1 y 3.2, tenemos que:

$$v_s = (10^{0.1A_s} - 1)^{1/2n} \quad (3.1)$$

$$v_s = f_s / f_p \quad (3.2)$$

Para el filtro de décimo orden, con la ecuación 3.1 se obtiene que $v_s = 2.19$.

Despejando f_s de la ecuación 3.2 y utilizando el valor calculado de v_s , se obtiene que $f_s = 19689.85Hz$.

Si queremos evitar el alias, y nuestra máxima frecuencia de interés es 19689.85Hz, tendríamos que muestrear a una frecuencia dos veces mayor que f_s , lo que nos da una frecuencia de muestreo de 39379.7Hz.

La frecuencia de muestreo con la que contamos es 35.6kHz y el valor calculado es 39.4kHz, como podemos observar el valor de la frecuencia de muestreo real es un poco menor que la frecuencia de muestreo calculada, pero si tomamos como máxima frecuencia de interés 8kHz, entonces ya no hay ningún problema, aunque el ADC detecte las frecuencia por encima de 8kHz, en el programa de C# simplemente se toman en cuenta las frecuencias de 8kHz para abajo.

Para diseñar un filtro pasa bajas de orden 10 es necesario colocar en cascada 5 etapas pasa bajas de segundo orden con factores Q como se indica en la tabla 2.1, la configuración de la etapa de segundo orden se muestra en la figura 2.18 y se utilizan las ecuaciones 2.32 y 2.33. En la tabla 3.1 se observan los valores calculados para R, C_1 y C_2 de las 5 diferentes etapas del filtro.

Cuadro 3.1 Valores de R, C₁ y C₂ para las 5 etapas del filtro anti-alias.

Etap	R	C₁	C₂
1	18kΩ	1nf	1nf
2	195kΩ	10pf	410pf
3	71kΩ	220pf	270pf
4	8.2kΩ	1nf	5nf
5	12kΩ	1nf	2nf

Los valores que se muestran en el cuadro 3.1 son valores aproximados a los valores calculados, debido a que se hicieron arreglos con valores de resistencias reales y es muy difícil obtener el valor exacto, sin embargo los valores mostrados en el cuadro 3.1 son muy cercanos a los valores calculados.

Voltaje de offset

Después de la etapa de filtrado, viene la etapa de offset, para esta función se utilizó la configuración del amplificador operacional como sumador, de la figura (2.2) del amplificador sumador, asignamos un valor de $R_1 = R_2 = R_3 = R_5 = 10k$ por lo tanto, R_4 y R_f son las incógnitas.

Partiendo de las ecuaciones (2.10) y (2.11):

$$A_{v3} = \left(\frac{R_f}{R_1 \parallel R_2} + 1 \right) \left(\frac{R_4 \parallel R_5}{R_3 + R_4 \parallel R_5} \right) \quad A_{v4} = \left(\frac{R_f}{R_1 \parallel R_2} + 1 \right) \left(\frac{R_3 \parallel R_5}{R_4 + R_3 \parallel R_5} \right)$$

A_{v3} es la ganancia de la señal de audio, pero como ya se amplificó anteriormente, $A_{v3} = 1$, A_{v4} es el voltaje de offset que se va a introducir a la señal, en este caso se tomó como señal de entrada el voltaje de 5V que alimenta al PIC, por lo tanto si queremos un voltaje offset de 2.5V necesitamos una ganancia de 0.5, por lo tanto $A_{v4} = 0.5$.

Para calcular los valores de R_4 y R_f se hace la igualdad entre la ganancia de la señal de audio y la ganancia de la señal de offset:

$$A_{v3} = 1 \quad A_{v4} = 0.5$$

Después de igualar las ecuaciones (2.10) y (2.11) utilizando los valores de A_{v3} y A_{v4} , se obtiene que:

$$R_4 = 2R_3$$

Esto significa que como a R_3 se le asigna un valor de 10k, entonces $R_4 = 20k\Omega$. Ahora se puede sustituir este valor en las ecuaciones del amplificador sumador y encontrar el valor de R_f .

Con los valores de resistencia propuestos y calculados se obtuvo:

$$R_4 \parallel R_5 = 6666.67\Omega \quad R_3 \parallel R_5 = 5000\Omega \quad R_1 \parallel R_2 = 5000\Omega$$

Sustituyendo estos valores en las ecuaciones de 2.10 y 2.11 se obtuvo que $R_f = 7500\Omega$.

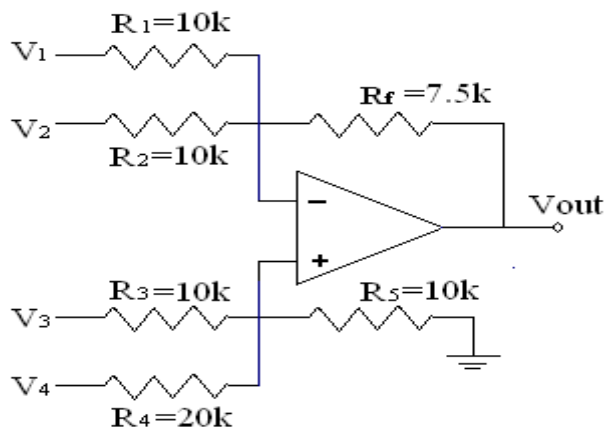


Figura 3.7 Circuito de offset.

En este circuito $V_1 = V_2 = 0v$ esto significa que R_1 y R_2 se conectan a tierra, V_3 es la señal de audio que viene de ser filtrada y $V_4 = 5V$ que con su respectiva ganancia nos dará un offset de 2.5V.

3.3 Implementación de la Transformada rápida de Fourier (FFT).

Una vez que se probó el acondicionamiento de señal y se midió la frecuencia de muestreo, se procedió a implementar el algoritmo de la FFT, el cual se consiguió de un ejemplo de aplicación de C# y se adaptó al lenguaje de ccs. El algoritmo de la FFT se habilitó como una función llamada FFT(), de esta manera podemos llamarla desde el programa principal cuantas veces sea necesario. El diagrama de bloques del algoritmo se muestra en la Figura 3.8.

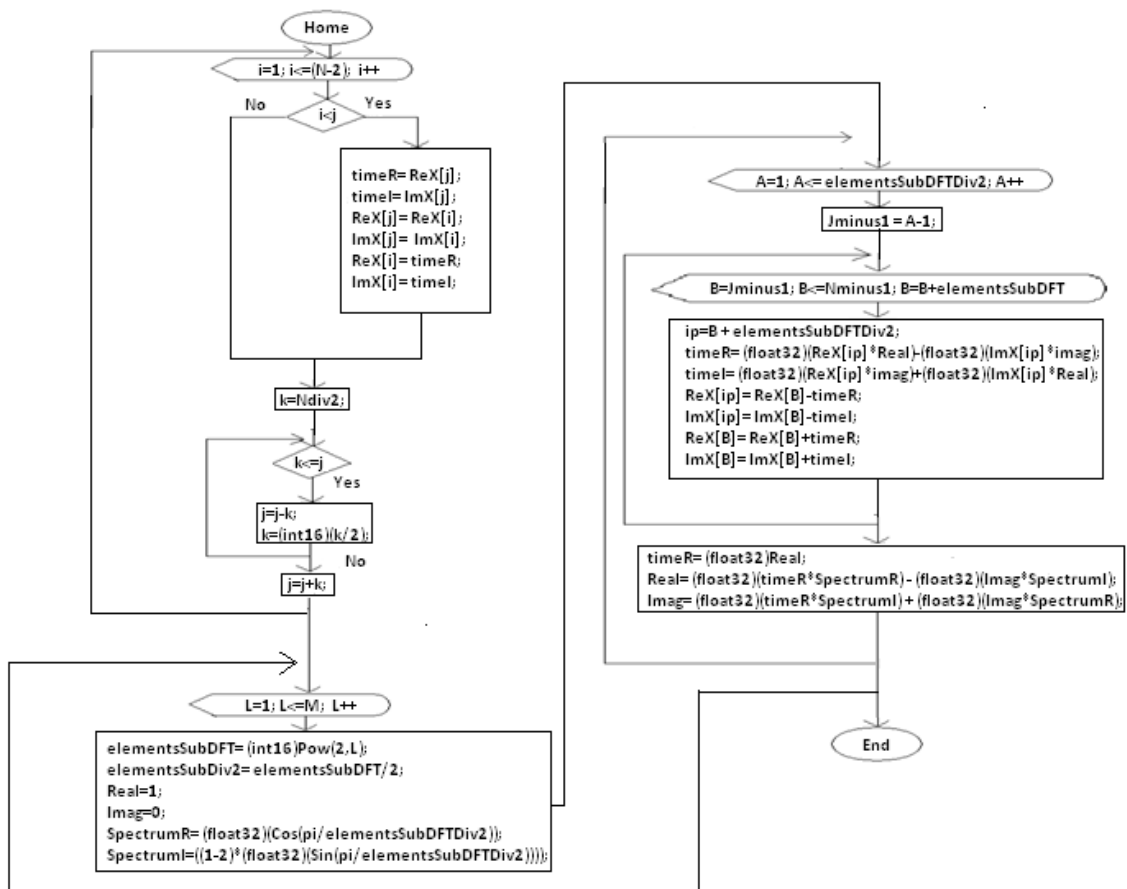


Figura 3.8 Diagrama de bloques del algoritmo de la FFT.

El algoritmo de la FFT es para N puntos, se puede escoger un número N de puntos, se puede cambiar el número de muestras simplemente con cambiar la variable N en las declaraciones del principio. La única condición que se necesita es que N sea una potencia de 2 ya que el algoritmo es el Radix-2.

El algoritmo de la FFT entrega dos resultados de salida para cada número de muestra, éste resultado consta de una parte real y una parte

imaginaria, cuando el algoritmo termina de realizar los cálculos, el microcontrolador tiene la tarea de calcular la magnitud de la parte real y la parte imaginaria de cada una de las salidas y estos son los datos que se mandan a la PC para que se grafiquen contra el vector de frecuencias. Para calcular el vector de frecuencias, se toma la frecuencia de muestreo (f_s) y se divide entre el número de muestras N , el vector de frecuencias se forma desde 0 hasta f_s en intervalos de f_s/N . La Figura 3.9 muestra el diagrama de funcionamiento de la FFT.

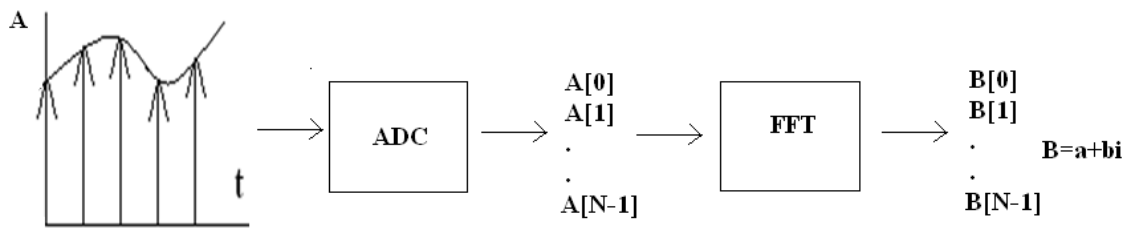


Figura 3.9. Diagrama de funcionamiento de la FFT.

3.4 Interfaz gráfica de usuario (GUI).

La Figura 3.10 muestra la pantalla de la interfaz gráfica de usuario:

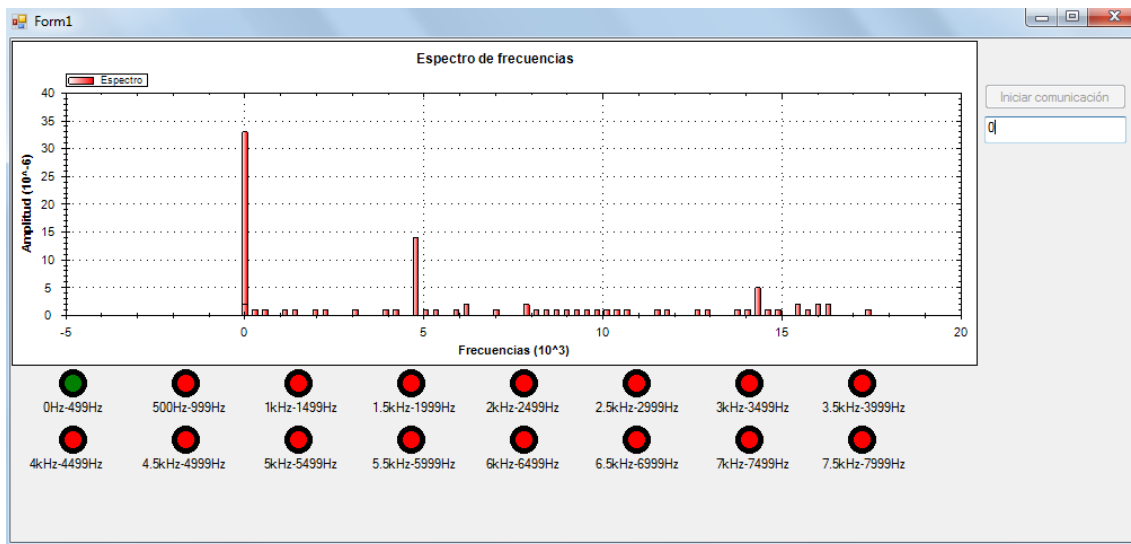


Figura 3.10 Pantalla de la interfaz gráfica de usuario (GUI).

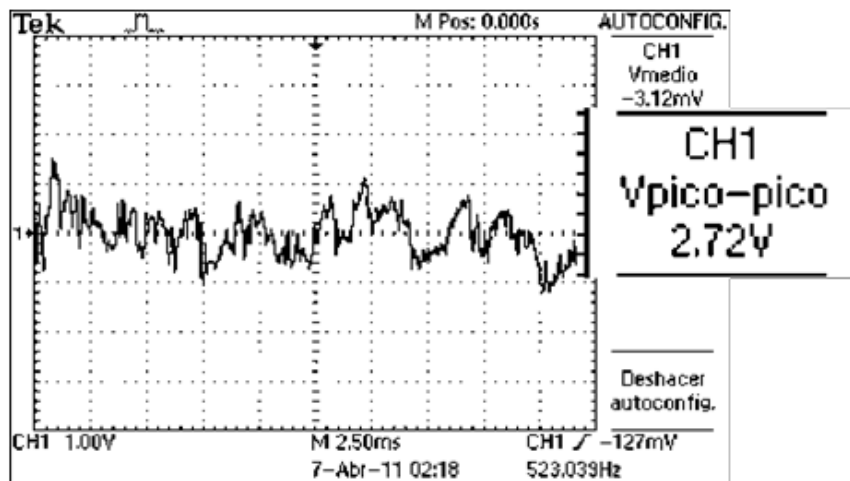
Como se puede observar en la figura 3.10, la interfaz consta de 3 partes, la parte superior izquierda es el espacio en donde se grafican los datos,

debajo de la zona del gráfico hay focos virtuales, los cuales están marcados con un rango de frecuencias, cuando un armónico que ésta en el intervalo de frecuencias del foco, tiene una amplitud mayor a la mínima establecida en el programa, el foco virtual se prende, lo cual indica que hay espectros de mucha amplitud en el rango de frecuencias del foco. Finalmente en la parte superior derecha de la interfaz se encuentra un cuadro de texto y un botón, al oprimir el botón, se inicializa la comunicación y el programa de la interfaz comienza a adquirir datos del microcontrolador y en el cuadro de texto se muestra el resultado de la frecuencia fundamental, la cual se obtiene a partir del armónico que contiene una mayor amplitud.

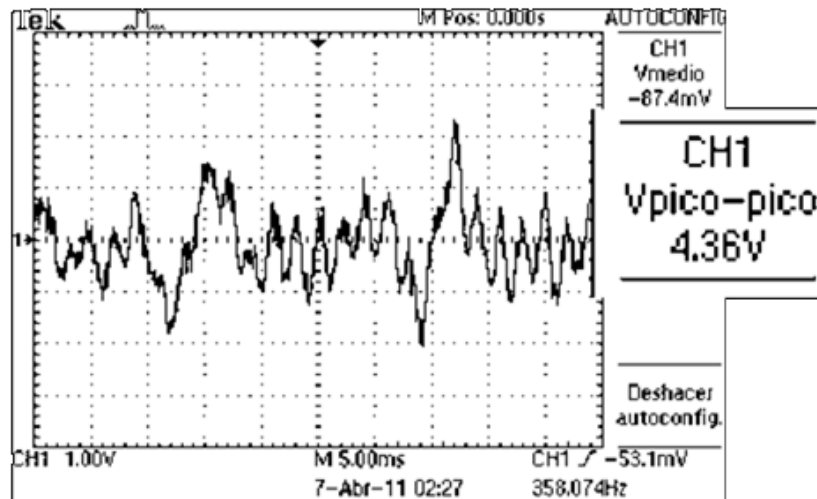
4 RESULTADOS Y DISCUSIÓN

4.1 Amplificador no inversor

Primero se revisó con el osciloscopio la señal del amplificador no inversor. En la Figura 4.1(a) se muestra la señal de audio sin amplificar y la figura 4.1(b) muestra la señal de audio después de haber sido amplificada.



(a)



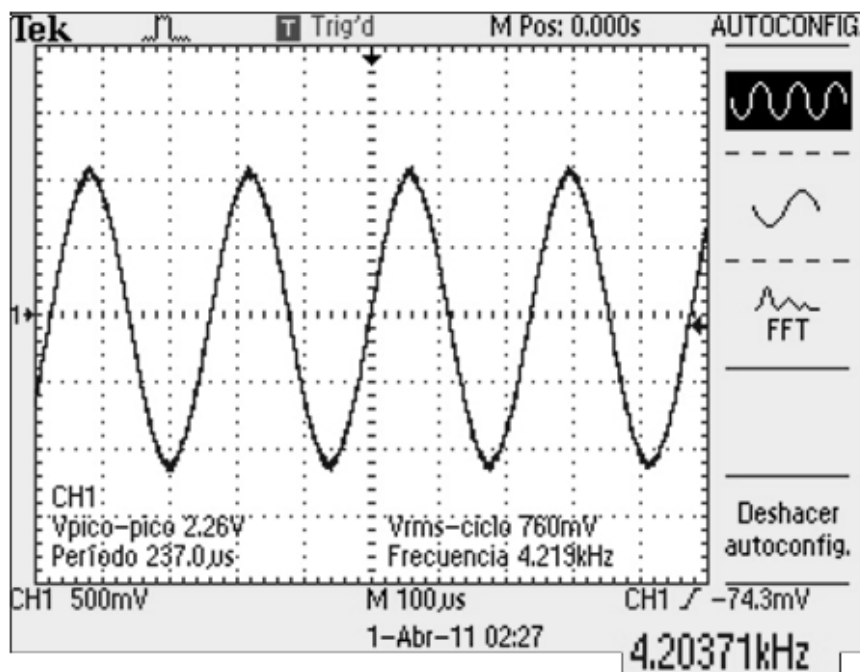
(b)

Figura 4.1 Medición de la señal de audio. **(a)** Señal de audio sin amplificar, **(b)** Señal amplificada de audio.

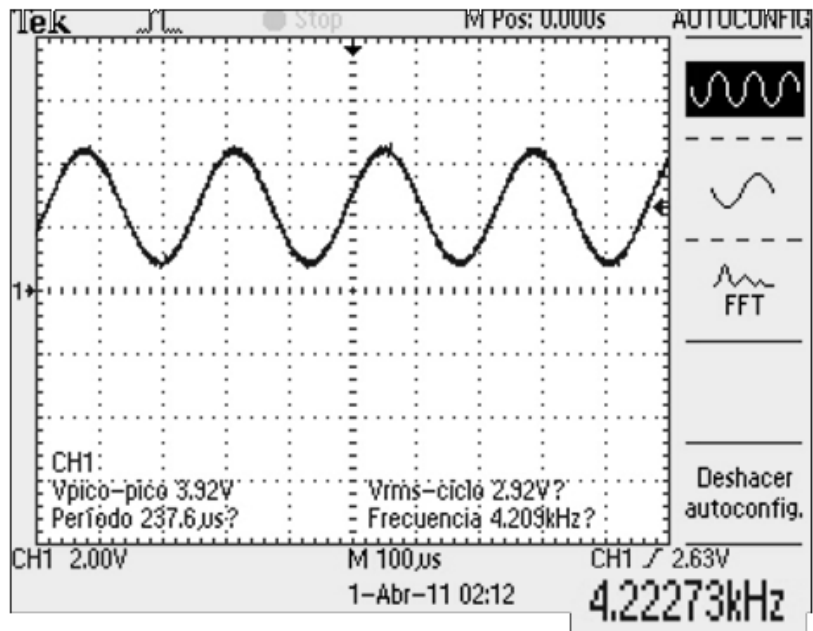
En la Figura 4.1(a) se muestra la señal de audio antes de pasar por la etapa de amplificación, en ésta figura se puede observar que el osciloscopio midió un voltaje pico-pico de 2.72V. La Figura 4.1(b) muestra la señal de audio después de ser amplificada, en esta ocasión el osciloscopio midió un voltaje pico-pico de 4.36V, a diferencia del voltaje pico-pico de 2.72V de la señal sin amplificar que se mostró en la Figura 4.1(a), éste resultado muestra que el circuito amplificador está funcionando correctamente.

4.2 Filtrado

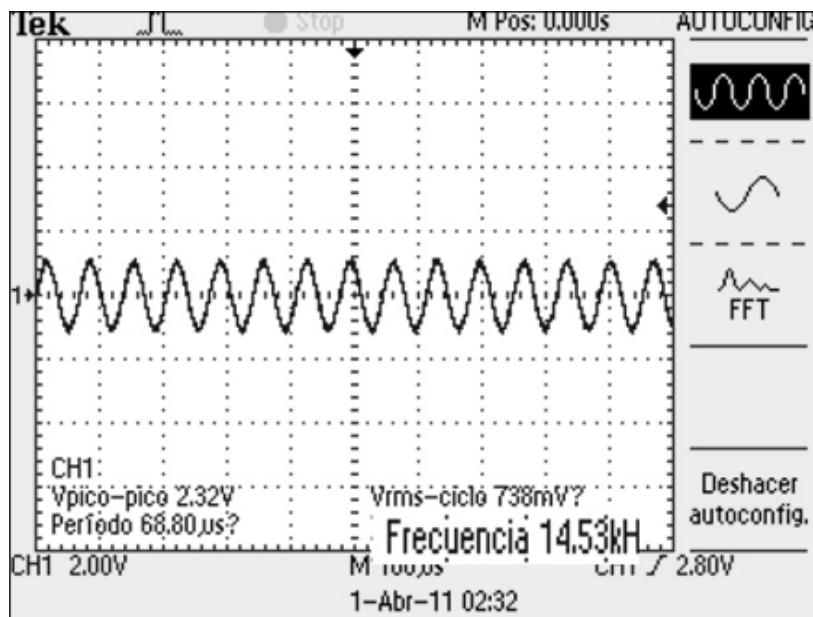
Como se mencionó en capítulo anterior, el filtro diseñado es de orden 10 y consiste en 5 etapas de segundo orden conectadas en cascada, una vez realizados los cálculos, se implementó el circuito y se realizaron pruebas con diferentes señales de entrada, algunos de los resultados de las pruebas se muestran en la Figura 4.2.



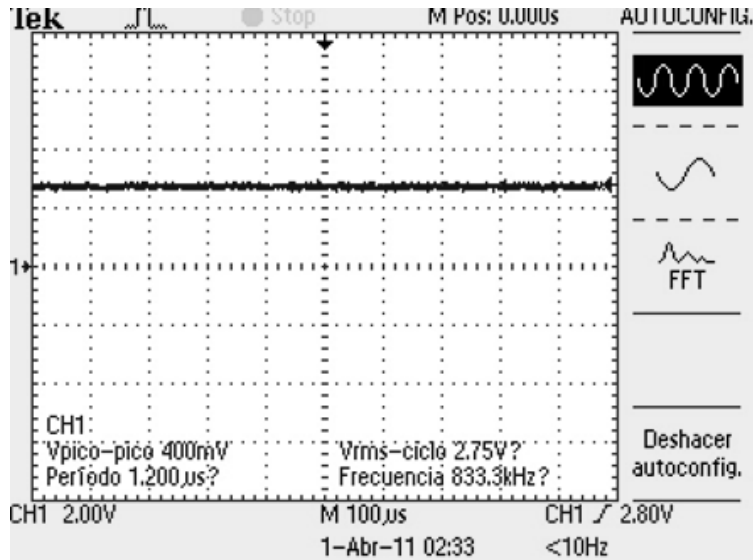
(a)



(b)



(c)



(d)

Figura 4.2 Respuestas del filtro. (a) Señal de entrada al acondicionador de señal de 4.2kHz, (b) Señal de salida del acondicionador de señal de 4.2kHz, (c) Señal de entrada al acondicionador de señal de 14.5kHz, (d) Señal de salida del acondicionador de señal de 14.5kHz.

En las Figuras 4.2(a) y 4.2(c), se muestran las medidas realizadas por el osciloscopio para las señales de entrada al acondicionamiento de la señal, las Figuras 4.2(b) y 4.2(d) muestra las señales medidas por el osciloscopio después de haber pasado por el acondicionamiento de señal (amplificación, filtro y circuito de offset) respectivamente. En la Figura 4.2(a) se puede observar que para una señal de entrada cuya frecuencia es de 4.22kHz, la señal de salida es la misma señal sinodal de entrada con una amplitud más grande debido al amplificador de señal, Figura 4,2(b). Este resultado es correcto ya que la frecuencia de la señal de entrada es menor que la frecuencia de corte del filtro de 9kHz, en la Figura 4.2(c), se observa que para una señal de entrada de 14.53kHz la señal de salida mostrada en la Figura 4.2(d) es una simple señal de 2.5V esto significa que la señal ha sido atenuada correctamente y lo único que se está midiendo es el voltaje de offset.

4.3 Voltaje offset

En la sección anterior, se pudo observar la respuesta del filtro y del circuito de offset, los resultados obtenidos fueron satisfactorios, la última prueba para el sistema de acondicionamiento de señal fue probarlo con una señal de entrada de audio, la señal proveniente de una computadora es de una canción que se está reproduciendo en la máquina, el resultado que se desea obtener es que la señal de audio oscile en un valor entre 0V y 5V, el resultado se observa en la Figura 4.3.

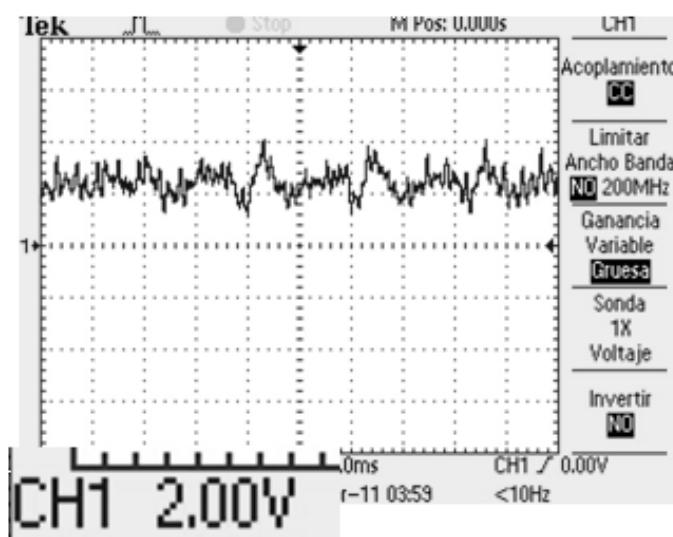


Figura 4.3. Respuesta del acondicionamiento de señal.

En la figura 4.3 podemos observar que la señal de audio ya no contiene voltajes negativos y que los voltajes oscilan perfectamente entre 0 y 5V. La figura 4.4 muestra la respuesta del circuito de acondicionamiento de señal cuando se suspende la señal de audio.



Figura 4.4. Voltaje de offset

Como se puede observar en la Figura 5.4, cuando el circuito no recibe la señal de audio, simplemente hay una señal con un voltaje promedio de 2.45V, lo cual indica que el circuito de voltaje de offset está funcionando correctamente.

4.4 Resultados del procesamiento.

La primera prueba que se realizó fue comprobar que el algoritmo de la FFT nos está entregando resultados correctos, para hacer esto, en el programa del microcontrolador se declaró un vector de 128 elementos, el valor de cada elemento se escogió aleatoriamente, cuando el microcontrolador concluye los cálculos de la FFT manda los resultados a la hiperterminal de Windows, estos resultados impresos en la hiperterminal son comparados con los resultados obtenidos al aplicar la FFT en Matlab para el mismo vector de datos, los resultados obtenidos del microcontrolador fueron exactamente iguales que los obtenidos de Matlab.

Para calcular la exactitud con la que el microcontrolador calcula las frecuencias, se realizó una prueba en donde se varía la frecuencia de entrada

al sistema en 200Hz comenzando desde 0Hz hasta 8000Hz, los resultados medidos se muestran en el Cuadro 4.1.

Cuadro 4.1 Respuesta del sistema a diferentes frecuencias de entrada.

Frecuencia de entrada (Hz)	Medición de frecuencia (Hz)	Frecuencia de entrada (Hz)	Medición de frecuencia (Hz)	Frecuencia de entrada (Hz)	Medición de frecuencia (Hz)	Frecuencia de entrada (Hz)	Medición de frecuencia (Hz)
0	0	2200	2247.19	4400	4494.38	6600	6460.68
200	280.89	2400	2528.09	4600	4494.38	6800	6741.16
400	280.89	2600	2528.09	4800	4775.28	7000	7303.37
600	561.79	2800	2808.99	5000	5056.18	7200	7303.37
800	842.69	3000	3089.88	5200	5056.18	7400	7584.27
1000	1123.59	3200	3089.88	5400	5337.08	7600	7865.17
1200	1123.59	3400	3370.78	5600	5617.98	7800	7865.17
1400	1404.49	3600	3651.68	5800	5898.88	8000	8146.07
1600	1685.39	3800	3932.58	6000	5898.88		
1800	1685.39	4000	3932.58	6200	6179.78		
2000	1966.29	4200	4213.48	6400	6460.68		

La Figura 4.5 muestra la gráfica de los datos de la tabla 4.1, y los compara con los valores que tendría que dar la medición.

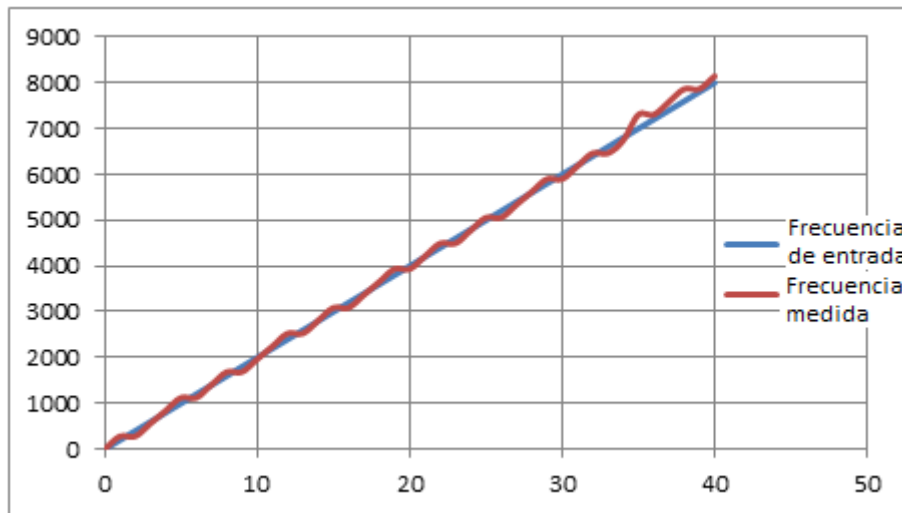


Figura 4.5. Gráfica de comparación entre la frecuencia de entrada y la frecuencia medida.

Como se puede observar de la Figura 4.5, la línea azul son los valores de la frecuencia de entrada y la línea roja son los valores que obtuvo el

microcontrolador, como se puede observar se obtuvo muy buena precisión en la medida de las frecuencias.

Con estos datos se calculó un error máximo del 4% con respecto a la frecuencia de la señal medida.

4.5 Resultados mostrados en la Interfaz gráfica de usuario.

La interfaz gráfica de usuario diseñada en C#, gráfica el espectro de las señales e imprime en pantalla el resultado de la frecuencia fundamental que es la que tiene el armónico de mayor magnitud, la Figura 4.6 muestra una señal sinodal de 5068Hz, después de que esta señal pasa por el acondicionador de señal se dirige a la entrada analógica del microcontrolador, el microcontrolador la procesa y envía los datos a la computadora, en donde la interfaz los recibe y muestra los resultados. La Figura 4.7 muestra los resultados mostrados por la interfaz para esta señal de entrada.

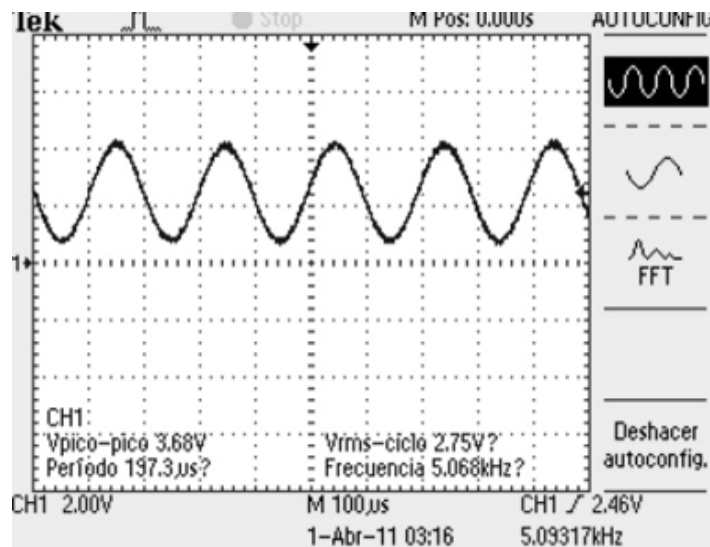


Figura 4.6. Señal de entrada de 5068Hz.

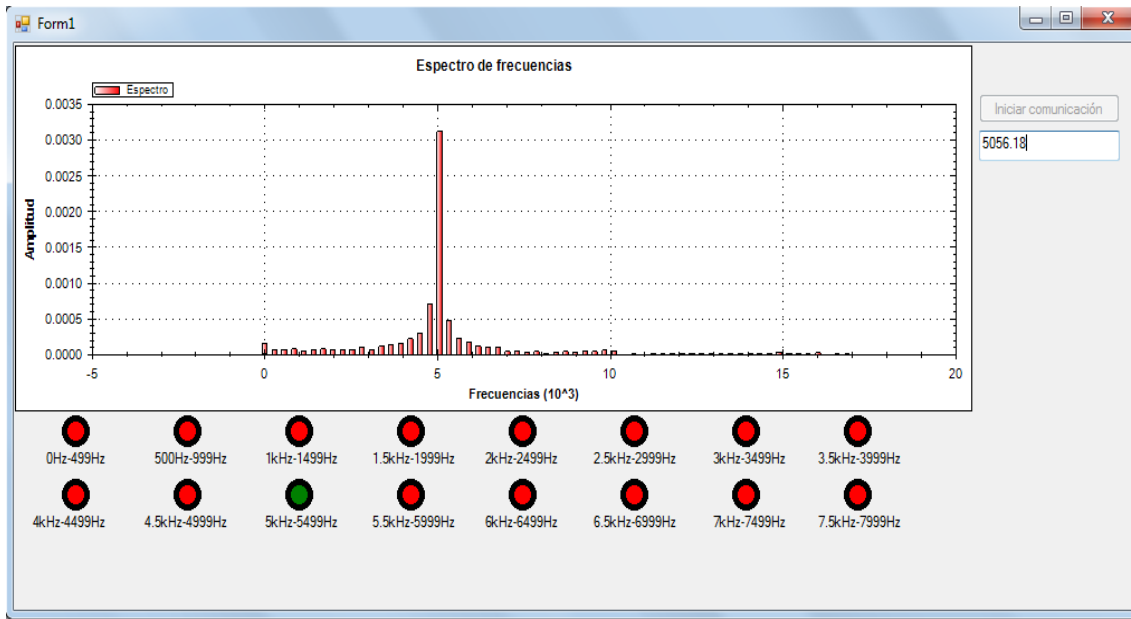


Figura 4.7. Resultados desplegados por la interfaz para la señal de entrada de 5068Hz.

Como se observa en la Figura 4.7, la interfaz marca un valor de 5056.18Hz lo cual está muy cercano al valor real, también se puede observar que la gráfica de los armónicos coincide con el resultado obtenido, al mismo tiempo que el foco virtual que simula la electroválvula del rango de 5kHz a 5.5kHz se prende indicando que el armónico más grande corresponde a su rango de frecuencias. En las Figuras 4.8 y 4.9 se muestran los resultados para otra señal de entrada.

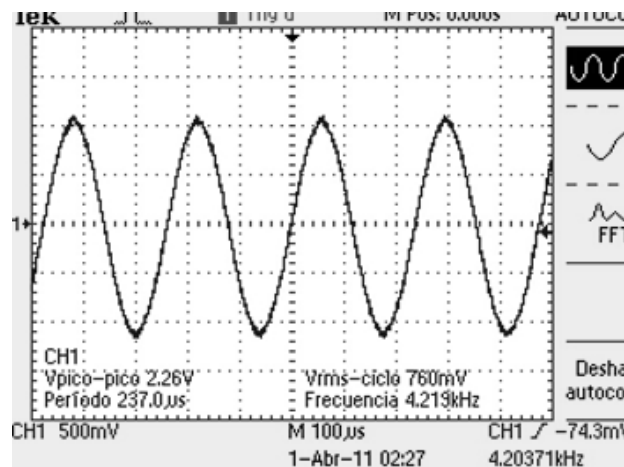


Figura 4.8. Frecuencia de entrada de 4219Hz.

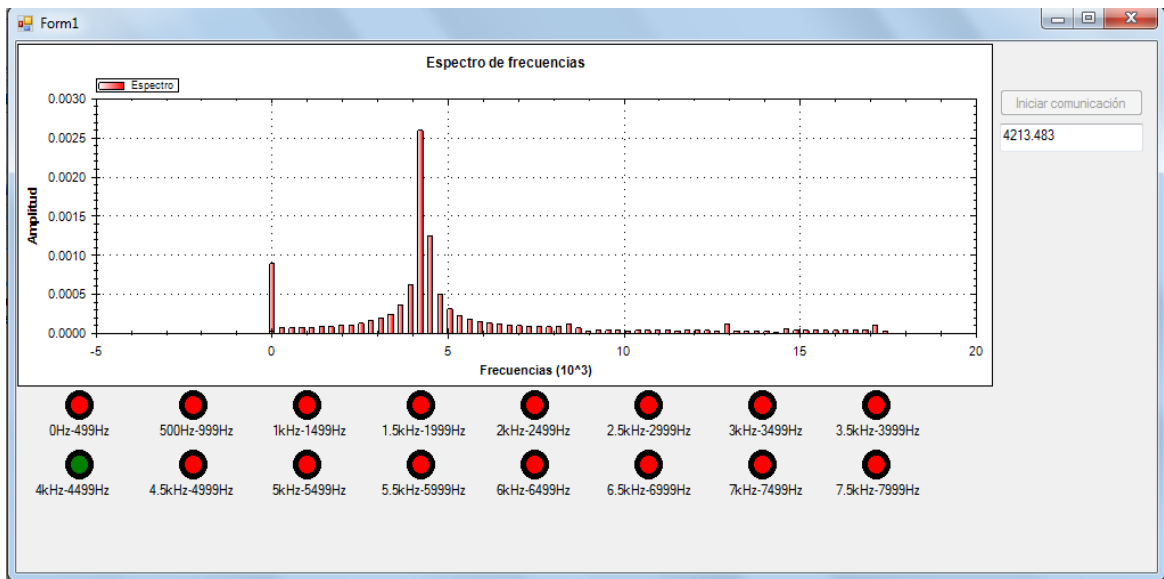
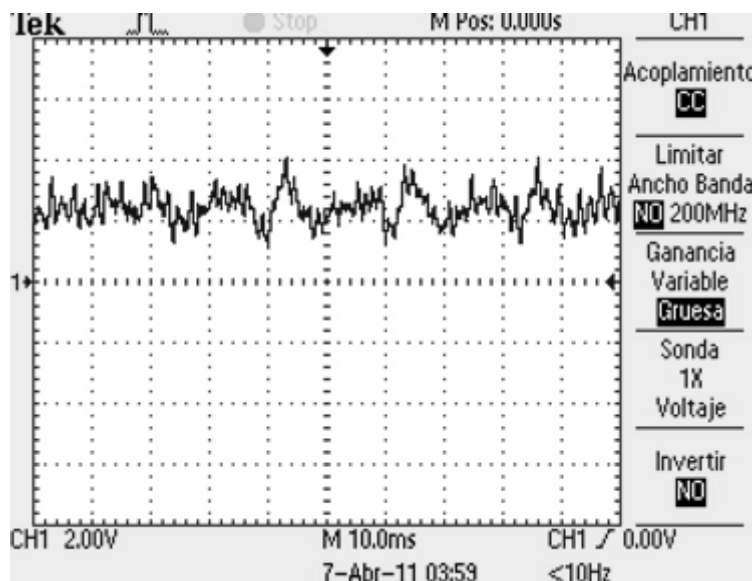
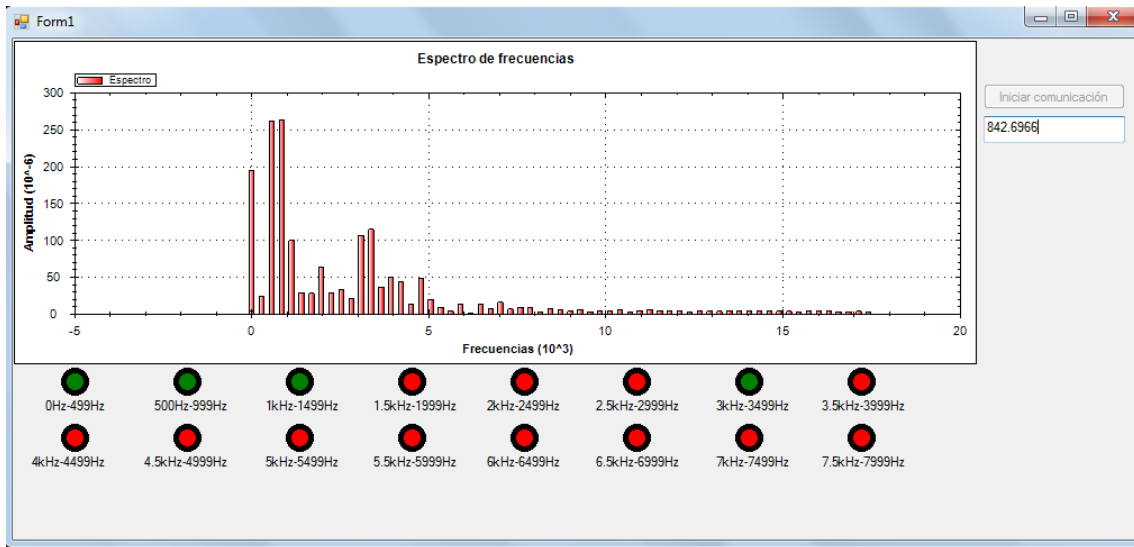


Figura 4.9. Resultados desplegados por la interfaz para la señal de entrada de 4219Hz.

En esta prueba la frecuencia de entrada es de 4219Hz, como podemos observar en la gráfica del espectro, el armónico fundamental se encuentra a la izquierda de la marca de 5kHz cerca de los 4200Hz y la medida de la frecuencia fundamental indica 4213.5Hz lo cual es un valor muy cercano al valor de frecuencia que está entrando al sistema. Las Figuras 4.10 y 4.11 muestran los resultados con una señal de audio a la entrada.



4.10. Señal de entrada de audio 1.



4.11. Respuesta a la señal de entrada de audio 1.

Como se observa en la Figura 4.11, la respuesta a la señal de audio es muy diferente a las señales sinusoidales, en la señal de audio tenemos diferentes armónicos que tienen una amplitud considerable y por esta razón en la simulación de las electroválvulas se observan más focos prendidos. Las Figuras 4.12 y 4.13 muestran la respuesta del sistema de una señal de audio que corresponde a la misma canción de los de las Figuras 4.10 y 4.11 pero en otro instante de tiempo.

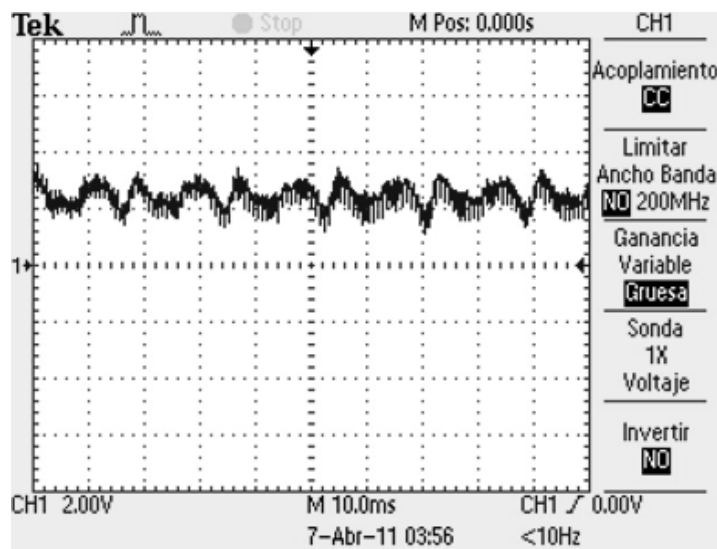
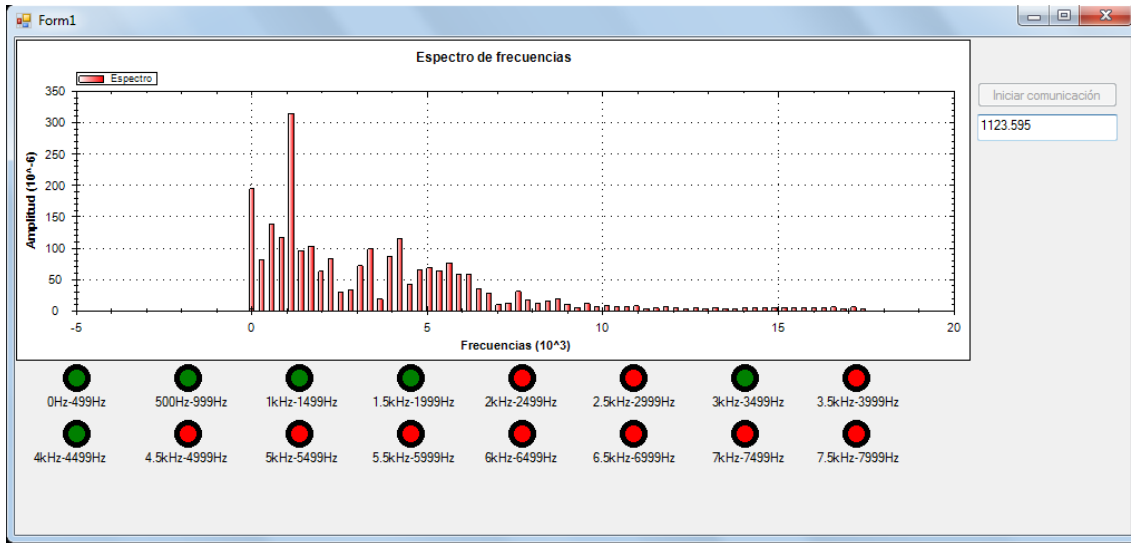


Figura 4.12. Señal de entrada de audio 2.



4.13. Respuesta a la señal de entrada de audio 2.

Como se puede observar en estos resultados, en otro instante de la canción, la señal tiene diferentes componentes de frecuencia, la gráfica de la Figura 4.13 muestra una gráfica con una mayor cantidad de espectros de gran amplitud, y por esta razón más focos virtuales se prenden.

5 CONCLUSIONES

Se logró demostrar que el PIC18F4550 tiene los recursos necesarios para llevar a cabo aplicaciones de audio, como es el caso de las fuentes bailarinas, la única limitante del PIC18F4550 es la memoria RAM, debido a esto. Este dispositivo sólo tiene memoria para procesar 128 datos, si aumentáramos el número de muestras, la precisión de la medida de frecuencia sería mayor, pero aun sin contar con esta prestación los resultados obtenidos fueron satisfactorios. Con la frecuencia de muestreo de 35.6kHz, se logró evitar el fenómeno del aliasing para el rango de frecuencias comprendidas debajo de los 9kHz que fue la frecuencia de corte del filtro anti aliasing. Al trabajar el PIC18f4550 a 48MHz se logró obtener una excelente velocidad de procesamiento, lo cual es útil para aplicaciones como fuentes bailarinas.

También se desarrolló una interfaz gráfica en C#. Esta interfaz nos permitió observar los resultados del sistema de una manera clara, lo cual facilitó observar con detalle las respuestas del sistema a diferentes señales de entrada.

En este proyecto aprendió la relación entre la electrónica analógica y la electrónica digital, si la parte analógica (acondicionamiento de señal) tiene errores, la electrónica digital aunque esté trabajando correctamente, obtendrá un resultado erróneo debido a que la señal de entrada es incorrecta. Al trabajar con microcontroladores es importante asegurarse de que el compilador instalado sea capaz de compilar todas las funciones con las que se esté trabajando, ya que esto puede causar muchos problemas a la hora de implementar los programas.

El objetivo del trabajo consistió en realizar una aplicación de software de fuentes bailarinas para demostrar que el microcontrolador puede procesar señales de audio, de esta manera, la etapa de potencia únicamente se diseñó para que los estudiantes de las próximas generaciones, ya tengan el circuito de potencia listo y puedan realizar prácticas con él.

Bibliografía

- [1] Ambardar, A. 2002, "Procesamiento de señales analógicas y digitales", (2da. edición) p 446-472 & 535-581.
- [2] Murakami, H. 2002, "Prime-Length Real-Valued Polynomial Residue Division Algorithms", IEEE Transactions on signal processing, vol. 50, no. 11.
- [3] Jo, B. G. and Sunwoo, M. H. 2005, "New Continuous-Flow Mixed-Radix (CFMR) FFT Processor Using Novel In-Place Strategy, Transactions on circuits and systems—I: regular papers, vol. 52, no. 5.
- [4] Konrad, J. and Agniel, P. 2006, "Subsampling Models and Anti-Alias Filters for 3-D Automultiscopic Displays", IEEE Transactions on image processing, vol. 15, no. 1.
- [5] Kim, M. G., Ahn, G., Hanumolu, P. K., Lee, S., Kim, S., You, S., Kim, S., Temes, G. C., Fellow, L. and Moon, U. 2008 "A 0.9 V 92 dB Double-Sampled Switched-RC Delta-Sigma Audio ADC", IEEE Journal of solid-state circuits, vol. 43, no. 5.
- [6] Yang, Y., Sculley, T. and Fellow, J., A 2008, "A Single-Die 124 dB Stereo Audio Delta-Sigma ADC With 111 dB THD", IEEE Journal of solid-state circuits, vol. 43, no. 7.
- [7] Malvino, A. and Bates, D. J., 2007, "Principios de electrónica", (7ª edición), p 622-666 & 740-796.
- [8] Stallings, W, 2000, "Comunicaciones y redes de computadores", (6ª edición), p 88 & 93-97.

APENDICE

Uno de los logros de este proyecto fue su aprobación para ponencia en el séptimo congreso internacional de ingeniería que organiza la facultad de ingeniería de la UAQ anualmente, a continuación se muestra el artículo que fue aceptado en dicho congreso. En el artículo se logran distinguir algunas diferencias respecto a lo mostrado en los capítulos de la tesis, una de ellas es que la frecuencia de muestreo mostrada en el artículo es de 16.75kHz mientras que el capítulo 3 de la tesis se muestra una frecuencia de muestreo de 35.69kHz, esto indica el incremento de la eficiencia del proyecto en los meses posteriores a la redacción del artículo. Otra diferencia importante es que cuando se redactó el artículo, la interfaz gráfica de usuario no contaba con la ventana donde se muestra la gráfica del espectro, simplemente se mostraba la medida de la frecuencia fundamental.

Procesamiento de audio en un microcontrolador de alto rendimiento utilizando la Transformada Rápida de Fourier

Audio processing in a high performance-microcontroller using the Fast Fourier Transform

Alberto Pimentel Martínez, Manuel Toledano Ayala.

Facultad de Ingeniería de la Universidad Autónoma de Querétaro.

RESUMEN. El objetivo de este trabajo es procesar señales de audio con un microcontrolador (PIC) de alto desempeño 18F45550. Primeramente, se diseñó una etapa de acondicionamiento de señales, el cual consiste en una etapa de amplificación, otra de filtrado (de orden 20) y finalmente una etapa que consistió en un amplificador sumador cuyo objetivo es introducir a la señal un voltaje de offset para evitar que lleguen al microcontrolador señales negativas de voltaje. Una vez terminada la parte analógica se comenzó a trabajar con el programa del microcontrolador. La primera tarea que se hizo con el microcontrolador fue programar la interrupción del Convertidor Analógico Digital (ADC) para muestrear la señal proveniente del filtro, después se implementó el algoritmo de la FFT y se procedió a realizar pruebas. Los resultados obtenidos del microcontrolador fueron comparados con los resultados obtenidos

de Matlab para el mismo vector de datos y los resultados se muestran al final del trabajo. Finalmente, una interfaz gráfica desarrollada en C# adquiere los resultados del microcontrolador y los muestra en pantalla.

Palabras clave: FFT, microcontrolador, procesamiento de señales, filtro anti-aliasing.

1. INTRODUCCION

En el procesamiento de señales es muy importante el concepto de filtro anti-aliasing, cuya función es limitar el ancho de banda de la señal, generalmente se utiliza el filtro pasa-bajas, aunque también el filtro pasa-banda es útil. Konrad y Phillipie Agniel usaron tres modelos para encontrar el filtro anti-aliasing ideal para señales de imágenes, estos modelos son: red ortogonal, red no ortogonal y la unión de cambio de red. Para evitar el fenómeno del aliasing es necesario contar con un ADC que sea capaz de muestrear como mínimo a una frecuencia dos veces mayor que la máxima frecuencia detectable por el ADC después de que la señal ha sido atenuada por el filtro. Es importante asegurarse que el ADC este entregando resultados confiables, por esta razón se utilizó la interrupción del ADC del microcontrolador, la cual se explica con detalles en la sección 2. La aplicación que se expone en este artículo es sencilla y tanto el ADC interno del microcontrolador como los amplificadores operacionales en el diseño del filtro satisfacen nuestra necesidad, sin embargo, en aplicaciones más complejas es necesario utilizar un ADC y circuitos analógicos que demanden menos potencia a la fuente de alimentación (Min Gyu Kim, et Al., 2008). YuQing Yang, et Al., (2008), propusieron un ADC de alto rendimiento y bajo consumo de energía, el cual permite conversiones más estables en las aplicaciones que requieren mucha exactitud.

El procesamiento de señales tiene muchas aplicaciones, cada aplicación tiene características diferentes, como por ejemplo, un analizador de espectro que se implementa para un instrumento de medición requiere de un análisis más preciso y cuidadoso que un analizador de espectro aplicado en unas fuentes bailarinas. Existen diversos algoritmos que realizan en cálculo de la transformada rápida de Fourier, los cuales son muy utilizados en el procesamiento de señales. La FFT requiere una gran cantidad de cálculos, los cuales demandan una gran cantidad de tiempo de procesamiento, por esta razón se han propuesto varios algoritmos que realizan la FFT (Byung G. Jo et al., 2005). Algunos ejemplos de estos algoritmos son el Radix-2 y el Radix-4, El algoritmo Radix-2 requiere más tiempo de procesamiento que el Radix-4, sin embargo el Radix-4 no puede procesar un número de puntos cuya potencia sea 2, solamente puede con un número de puntos que sean potencia de 4 (Byung G. Jo et al., 2005), a diferencia del Radix-2 que puede realizar el cálculo cuando el número de muestras es potencia de 2, la limitante de los algoritmos mencionados anteriormente es que únicamente pueden trabajar con un número de muestras que es potencia de dos y de cuatro, por esta razón se han propuesto algoritmos que puedan trabajar con un número de muestras que no sea potencia de dos o de cuatro, una de estas propuestas es el algoritmo Bruun, está diseñado para trabajar con un número de muestras cuya potencia es diferente de dos. (Hideo Murakami, 2002). En el caso de las fuentes bailarinas se utilizó el algoritmo Radix-2, aunque cualquiera de los otros algoritmos también son útiles para esta aplicación.

La interfaz gráfica de usuario, permite visualizar los resultados de una manera clara y sencilla, en la sección 2 de éste artículo se muestra el diseño de la pantalla y en la sección 3 se muestran los resultados que despliega al recibir datos del microcontrolador, en la sección 3 se presentan los resultados y finalmente, en la sección cuatro se presentan las conclusiones.

2. DESARROLLO E IMPLEMENTACION

El sistema requiere de una etapa de acondicionamiento de señal, esta etapa se constituye por tres módulos los cuales son: amplificación de la señal de audio, filtro anti-alias y finalmente un módulo que le agrega a la señal un voltaje de offset. El cerebro del sistema es un microcontrolador, el cual recibe la señal del circuito de acondicionamiento y se encarga de muestrear la señal con uno de sus canales analógicos, para muestrear la señal se implementó la interrupción de ADC la cual interrumpe al programa cuando el ADC termina de convertir un dato, de esta manera sabremos que el dato que estamos procesando está correctamente cuantificado por el ADC. La comunicación de la PC con el microcontrolador se realiza mediante RS-232 el cuál es un protocolo muy utilizado en la industria.

La interfaz gráfica está diseñada en C#. Se eligió este lenguaje porque es fácil de utilizar y permite hacer interfaces gráfica usuario (GUI's) muy amigables para los usuarios. En la figura 1 se muestra el diagrama de bloques del sistema.

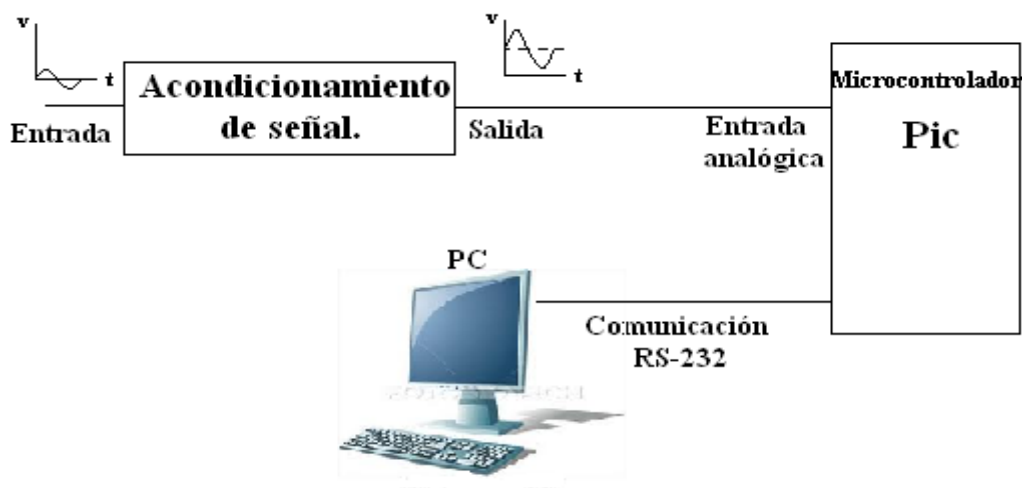


Figura 1. Diagrama a bloques del sistema.

2.1 Acondicionamiento de señal

Generalmente las señales de audio que provienen de la conexión plug de una computadora tienen un nivel de voltaje muy pequeño, esto nos origina la necesidad de amplificarla hasta un nivel de voltaje ideal para el dispositivo que recibirá la señal, en el caso del pic18 los límites de la señal tendrán un rango de 0 a 5V.

Después de que la señal ha sido amplificada, es necesario limitar el ancho de banda para evitar el fenómeno alias, esta tarea la realiza un filtro pasa bajas, la frecuencia de corte y el orden del filtro dependen de la resolución del ADC y de la máxima frecuencia a la que podamos muestrear con el microcontrolador.

La entrada analógica del microcontrolador no puede recibir voltajes negativos, si queremos analizar una señal de audio o simplemente una señal senoidal, es necesario agregar a las señales un

voltaje de offset, de esta manera se le agrega a la señal un offset de 2.5V, ya que el microcontrolador acepta un rango de voltaje de 0 a 5V, entonces la señal tendrá como máximo un voltaje pico de 2.5V después de pasar por la etapa de amplificación y su referencia serán los 2.5V del offset.

2.1.1 Amplificación de la señal.

Para diseñar el amplificador de la señal, se midió con el osciloscopio el voltaje de la señal de audio de la computadora, la figura 2 muestra el resultado de la medición.

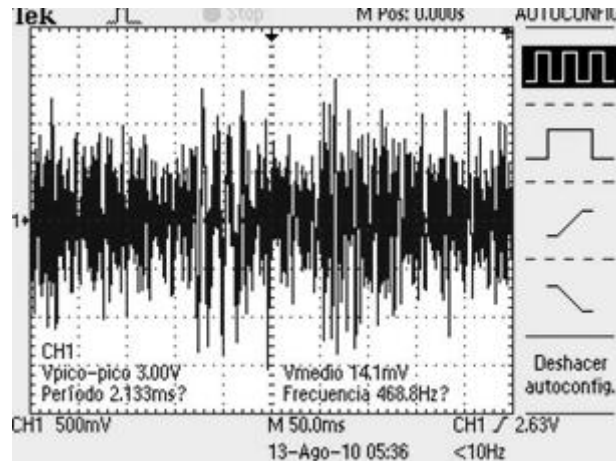


Figura 2. Señal de audio sin amplificar proveniente de una PC.

En la figura 2 se puede observar que la señal tiene picos que alcanzan 1.5Vp. Para saber la ganancia del amplificador, se toma el valor máximo de amplitud, en este caso 1.5V el voltaje offset es de 2.5V y la señal oscilará entre 0 y 5V, como la referencia es 2.5V entonces la amplitud pico de la señal tiene que ser de 2.5V. Para obtener la ganancia del amplificador se aplicó la siguiente operación:

$$Ganancia = \frac{2.5v}{1.5v} = 1.6666667$$

Se eligió la configuración del amplificador no inversor, cuya ecuación es la siguiente:

$$A_v = \frac{R_f}{R_1} + 1 \quad (2.1)$$

Nuestra ganancia A_v es igual a 1.6666667, si proponemos que $R_1 = 10k\Omega$ obtenemos que $R_f = 6.67k\Omega$, el circuito con la configuración de amplificador no inversor quedara de la siguiente manera:

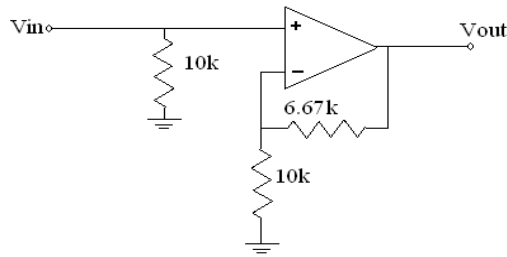


Figura 3. Esquemático para la amplificación de audio.

2.1.2 Filtrado

Para poder encontrar el orden adecuado del filtro, es necesario saber en qué instante de la atenuación del filtro el ADC ya no es capaz de leer los voltajes, para realizar dicho cálculo se utiliza la ecuación:

$$A_s = 20 \log(2^B \sqrt{6}) \quad (2.2)$$

Donde B es la resolución del ADC y A_s la atenuación en decibeles. Al aplicar ésta fórmula para una resolución del ADC de 10 bits se obtuvo el siguiente resultado:

$$A_s = 20 \log(2^{10} \sqrt{6}) = 20 \log(2^{10} \sqrt{6}) = 68dB$$

El resultado anterior quiere decir que cuando el filtro logra atenuar 68dB la señal, el ADC de 10 bits ya no será capaz de detectar los voltajes atenuados a ésta magnitud.

Se utilizó un filtro Butterworth de orden 20 para evitar el fenómeno del aliasing. Del resultado obtenido al aplicar la ecuación (2.2), sabemos que el filtro tiene que atenuar la señal a 68db antes de que la frecuencia de la señal sea mayor o igual a la frecuencia de muestreo, la frecuencia de muestreo es de 16744Hz

Debido a que la frecuencia de muestreo es de 16744Hz la frecuencia de corte del filtro tiene que ser menos de 16744Hz/2, según el teorema de muestreo de Nyquist que establece que se debe muestrear a una frecuencia 2 veces mayor que la frecuencia más grande de la señal muestreada, para hacer que la frecuencia de corte sea lo más grande posible se trabajó con un orden del filtro igual a 20. Este filtro fue construido con 10 etapas de segundo orden conectadas en cascada, la configuración de la etapa de segundo orden de filtro Butterworth se muestra a continuación:

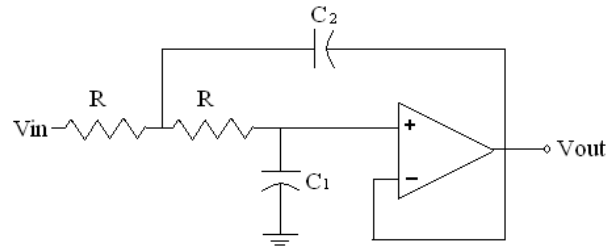
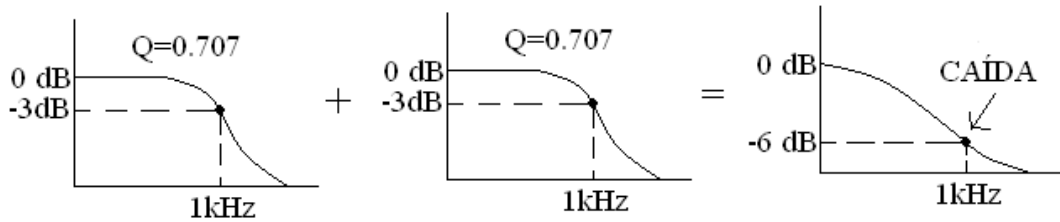


Figura 4. Configuración de segundo orden para filtro Butterworth.

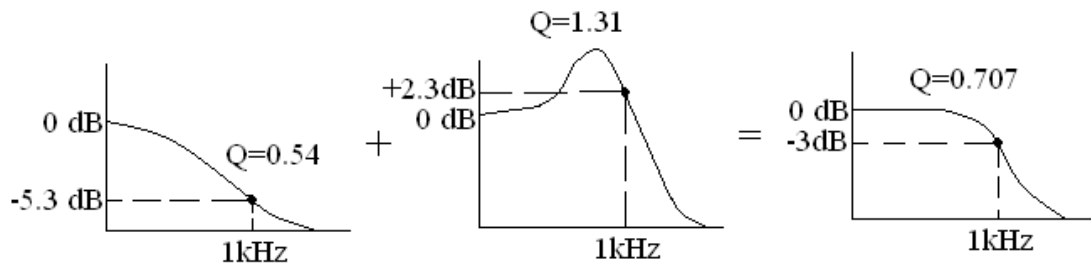
Quando se conectan filtro Butterworth en cascada, la respuesta no es exactamente Butterworth debido a que las atenuaciones de las diferentes etapas se suman y hacen que la atenuación en la frecuencia de corte aumente, para ello se tiene que variar la atenuación en las diferentes etapas, la figura 5 se muestra éste concepto:



(a)



(b)



(c)

Figura 5. (a) Dos etapas conectadas en cascada. (b) Etapas iguales producen una caída en la frecuencia de corte. (c) Etapas de bajo Q y de alto Q se compensan para generar una respuesta de Butterworth.

Para poder utilizar una frecuencia de corte de 9kHz, se conectaron des etapas de décimo orden en cascada, cada una de ellas con respuesta de Butterworth normal. La respuesta es parecida a la de la figura 5b. De esta manera estamos reduciendo el fenómeno del aliasing y muestreando un mayor rango de frecuencias. Los valores de Q para las diferentes etapas de los filtros de décimo orden se muestran en la tabla 2.1.

Tabla 1. Valores Q de las 5 etapas para los filtros de décimo orden.

Etapa 1	Etapa 2	Etapa 3	Etapa 4	Etapa 5
Q=0.5	Q=3.2	Q=0.56	Q=1.1	Q=0.707

En la siguiente sección se muestran las respuestas del filtro, como se podrá observar, la atenuación es mucho mayor a 6dB a frecuencias poco menores a la frecuencia de corte, lo cual se debe a la tolerancia de los componentes y como se trata de 10 etapas de segundo orden, los errores por tolerancia aumentan, pero al atenuar mucho más que 6dB en la frecuencia de corte resulta muy favorable en la lucha contra el aliasing.

2.1.3 Voltaje offset

Para evitar que el microcontrolador reciba voltajes negativos, se le agregó a la señal de audio un voltaje offset de 2.5V, esto es logrado mediante la configuración del amplificador sumador no inversor, cuyas ecuaciones son las siguientes:

$$A_{v3} = \left(\frac{R_f}{R_1 \parallel R_2} + 1 \right) \left(\frac{R_4 \parallel R_5}{R_3 + R_4 \parallel R_5} \right) \quad (2.3)$$

$$A_{v4} = \left(\frac{R_f}{R_1 \parallel R_2} + 1 \right) \left(\frac{R_3 \parallel R_5}{R_4 + R_3 \parallel R_5} \right) \quad (2.4)$$

El diseño final del circuito se muestra en la figura 4.

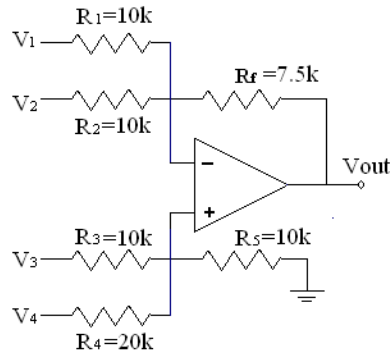


Figura 6. Esquemático para el ajuste de offset.

En este circuito $V_1 = V_2 = 0v$ esto significa que R_1 y R_2 se conectan a tierra, V_3 es la señal de audio que viene de ser filtrada y $V_4 = 5V$ que con su respectiva ganancia nos dará un offset de 2.5V.

1.2 Adquisición de datos

Usando la interrupción #INT_AD, se interrumpe al programa cuando la conversión analógica digital ha sido completada, de esta manera obtenemos un muestreo confiable, una vez que la interrupción entra, se lee el resultado de la conversión y enseguida se inicia la adquisición de un dato nuevo. Otra ventaja de utilizar esta interrupción es que en cada interrupción se manda un pulso a una terminal del microcontrolador y con ayuda del osciloscopio se mide el tiempo que hay entre interrupciones, y de esta manera se obtiene la frecuencia a la cual estamos muestreando.

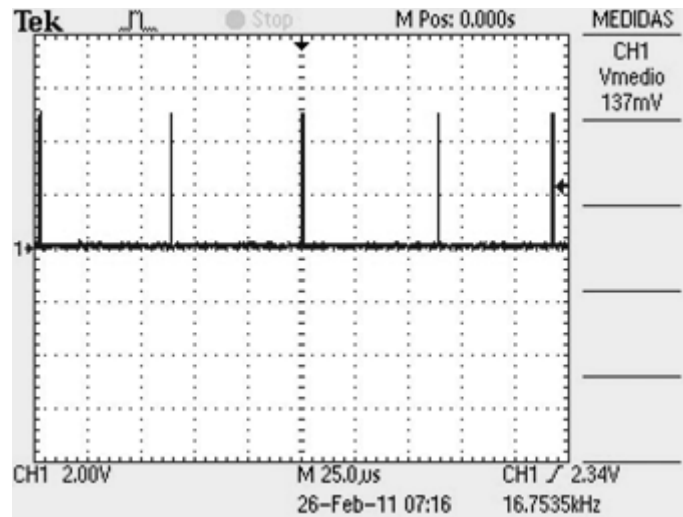


Figura 7. Frecuencia de muestreo de 16.7535 KHz.

Como se muestra en la figura 7, la frecuencia de muestreo es de 16.753kHz, sin embargo cabe señalar que la frecuencia oscila entre 16.75kHz y 16.76kHz.

1.3 Implementación de la Transformada Rápida de Fourier (FFT)

El algoritmo utilizado fue el Radix-2, el cual fue adquirido del manual de aplicaciones de C# y adaptado para utilizarlo en el software PIC C Compiler. El algoritmo trabaja con dos vectores de entrada: ReX, el cual contiene los N datos muestreados e ImX el cual es un vector de tamaño N con todos sus elementos iguales a 0, una vez que el algoritmo de la FFT modifica a ReX e ImX, ReX contiene a la parte real del resultado e ImX contiene la parte imaginaria del resultado, una vez que se obtiene la parte real e imaginaria, el programa calcula el espectro de frecuencias, para hacer esto el microcontrolador calcula el módulo de la parte real e imaginaria de los elementos, esto se hace calculando el módulo de ReX[0] con ImX[0] hasta ReX[N] con ImX[N], éste nuevo vector de módulos es el que se envía por RS-232 y C# lo compara con el vector de frecuencias.

El algoritmo de la FFT es para N puntos, siempre y cuando el número de puntos sea una potencia de dos, la memoria del microcontrolador es el factor que limita el número de muestras que pueden ser analizadas, en el caso del pic18f4550 se pueden analizar 128 muestras.

Cuando el microcontrolador termina de obtener las 128 muestras mediante el ADC, éste deja de obtener más muestras y se dedica a aplicar el algoritmo de la FFT a las 128 muestras obtenidas, cuando el algoritmo termina de hacer los cálculos, los resultados son enviados por RS-232 a la PC, en donde la interfaz de usuario mostrará los resultados.

Para comprobar que el algoritmo estuviera realizando cálculos correctos, se compararon los resultados del microcontrolador con los resultados arrojados por Matlab para el mismo vector de datos. El microcontrolador produjo los mismos resultados.

El programa principal llama al algoritmo de la FFT mediante la función FFT(), ésta función no recibe parámetros de entrada ni regresa parámetros de salida, por lo tanto todas las variables declaradas en el programa son globales, en la figura 8 se muestra la función del algoritmo de la FFT.

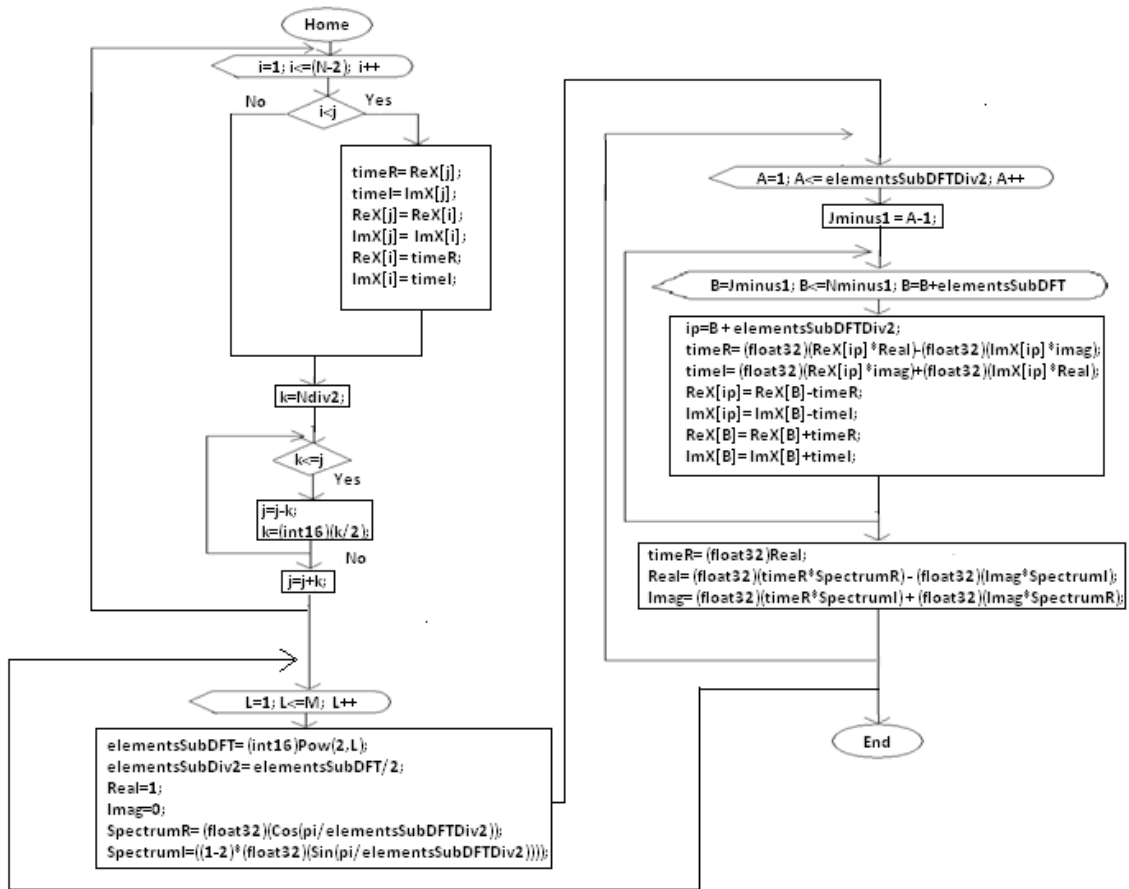


Figura 8. Algoritmo de la FFT.

1.4 Interfaz gráfica de usuario (GUI)

Las primeras pruebas de comunicación se realizaron utilizando la Hyperterminal de Windows. Posteriormente se implementó una interfaz en Matlab para su análisis y utilizar sus herramientas gráficas para después realizar una interfaz gráfica en C#.

El programa de la interfaz calcula el vector de frecuencias y lo asocia con el vector de resultados que recibe del microcontrolador. La principal función de la interfaz es mostrar en un cuadro de texto la frecuencia fundamental de las señales en caso de ser periódicas y mostrar en que rangos de frecuencias están los espectros más altos de la señal.

Para calcular el vector de frecuencias, el programa toma la frecuencia de muestreo (f_s), que es de 16744Hz y la divide entre número de muestras (N), el vector de muestras se forma desde 0 hasta f_s en intervalos de f_s/N .

La pantalla de la interfaz se compone por un editor de texto, en donde se muestra la frecuencia fundamental, el recuadro rojo de la figura 3 señala ésta parte, el recuadro verde señala un conjunto de focos que corresponden a diferentes rangos de frecuencias, los cuales se prenden cuando en sus rangos de frecuencia hay espectros cuya amplitud es cercana a la amplitud del espectro más grande, finalmente el

recuadro azul señala los botones de inicio y paro, cuando se presiona el botón de inicio, el programa comienza a adquirir los datos del microcontrolador, cuando se presiona el botón detener se cancela la comunicación, al presionar el botón inicio otra vez, se reinicia la comunicación.

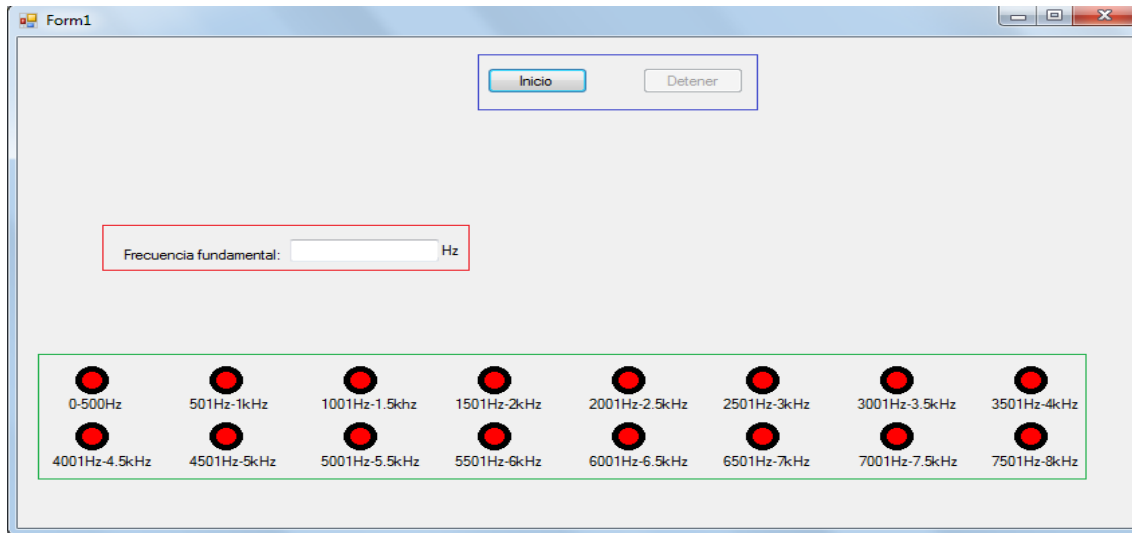


Figura 9. Diseño de la interfaz gráfica de usuario (GUI).

3. PRUEBAS Y RESULTADOS

La figura 10 muestra los resultados obtenidos del osciloscopio al capturar las señales de salida del circuito de acondicionamiento de señal, tras recibir señales de entrada de diferentes frecuencias. A continuación se muestran las señales medidas en el osciloscopio.

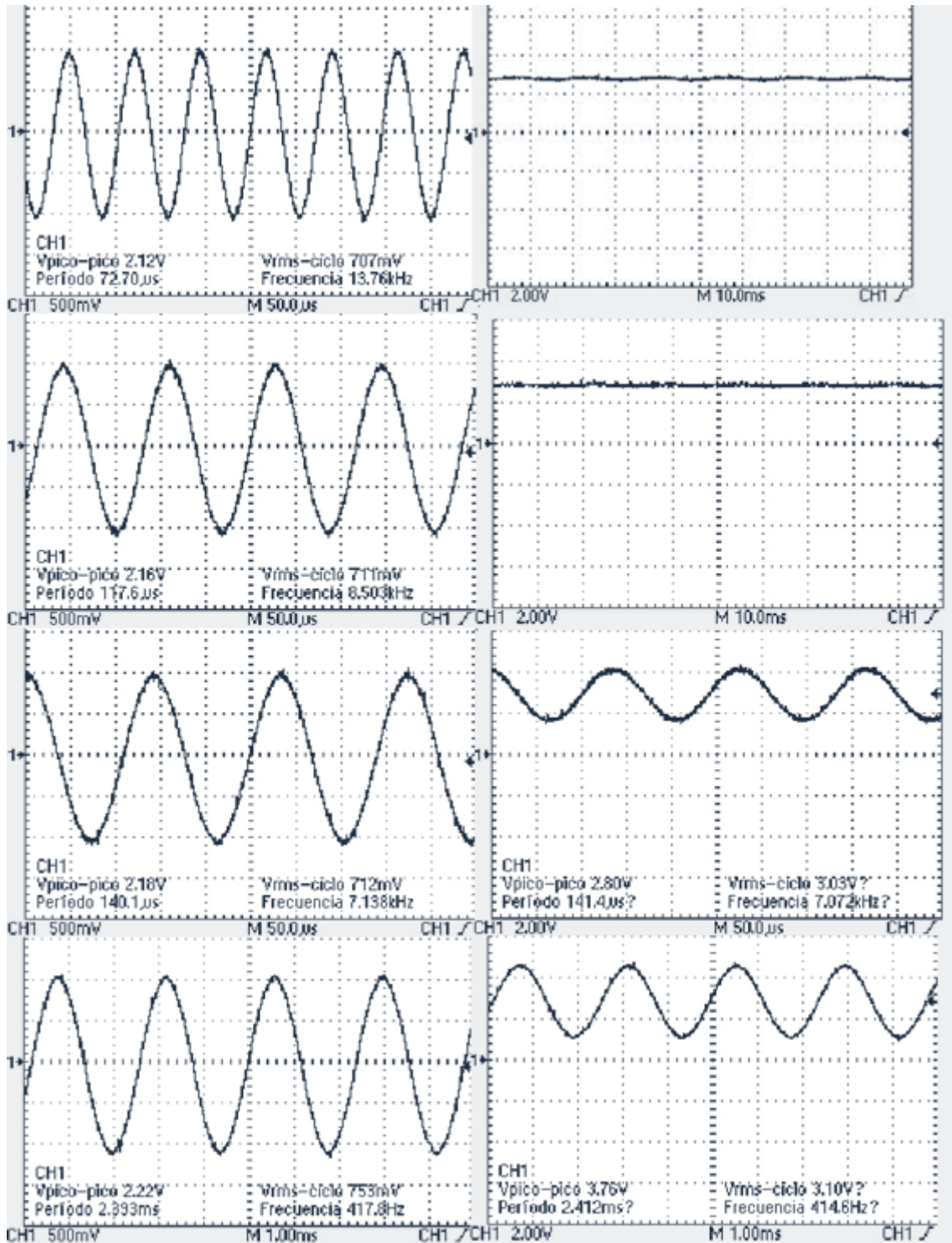


Figura 10. Respuestas del acondicionamiento de señal.

Las pantallas del lado izquierdo de la figura 4, son las señales de entrada al circuito, las pantallas del lado derecho representan la salida del circuito, las cuales están a lado de la pantalla que muestra su respectiva entrada, las pantallas de hasta arriba muestran la respuesta del acondicionador de señal a una frecuencia de 13700Hz, como se puede observar la señal de salida únicamente muestra una línea de voltaje a 2.5V, lo cual indica que la señal del seno ha sido filtrada por completo, las siguientes pantallas muestran la respuesta del circuito a una frecuencia de 8500Hz. Como se puede observar, la señal de salida

sigue igual que la respuesta anterior, lo que significa que el muestreo a una frecuencia de 16744Hz es adecuado y el acondicionamiento es capaz de reducir el fenómeno del aliasing, las pantallas de abajo muestran la respuesta del circuito a frecuencias de 7138Hz y 417Hz, como se puede observar las señal del seno comienza a aparecer en las pantallas que muestran las señales de salida.

Para comprobar que el microcontrolador está procesando los datos correctamente, se realizaron pruebas con señales senoidales aplicadas a la entrada del circuito de acondicionamiento de señal y la señal de salida conectada al canal analógico del microcontrolador con una frecuencia de 712Hz.

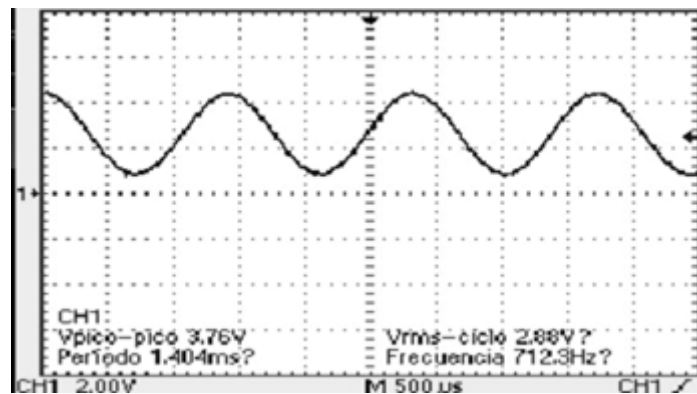


Figura 11. Señal de entrada al ADC del microcontrolador.

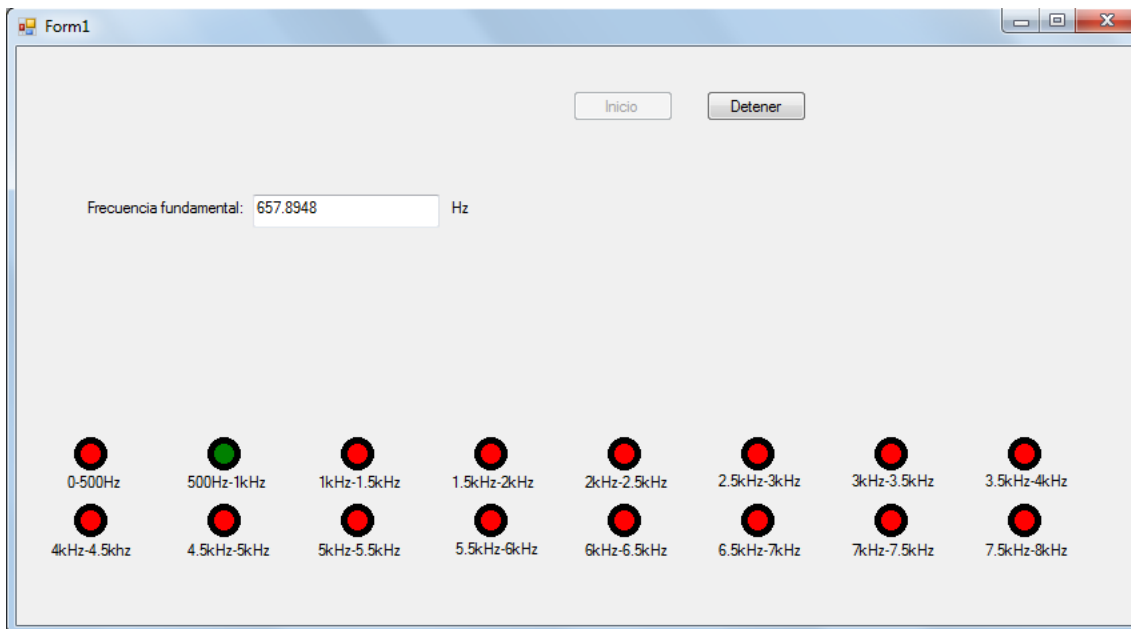


Figura 12. Resultados de la señal de 712Hz

Como podemos observar en la figura 12, la medición tiene un error de 54Hz debido a errores ocasionados por el número de muestras tomadas no es suficiente, sin embargo este número no puede ser elevado ya que el microcontrolador necesitaría más capacidad de procesamiento. En la siguiente figura se muestra una frecuencia de entrada de 4120Hz.

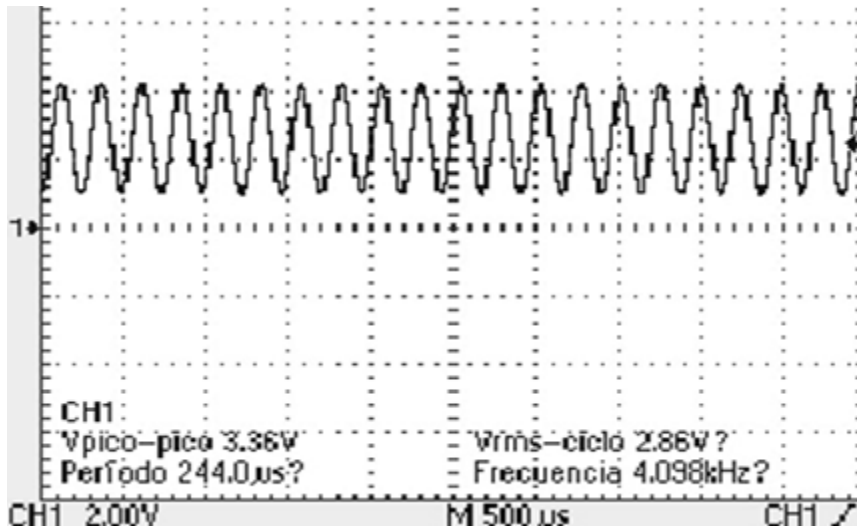


Figura 13. Señal de entrada al ADC del microcontrolador.



Figura 14. Resultados de la señal de 4120Hz

4. CONCLUSIONES

Se diseñó un analizador de espectro con un microcontrolador de alto rendimiento, sin embargo, se recomienda para aplicaciones donde el tiempo de respuesta no sea crítico, ya que algoritmo de la FFT demanda mucho tiempo de procesamiento.

El microcontrolador arrojó resultados cercanos al valor real, al aumentar el número de muestras se obtendría una mejor precisión, sin embargo la memoria del microcontrolador nos limita. Se puede observar de las dos pruebas realizadas, que cuando aumenta la frecuencia de entrada el error de la medición también aumenta.

Otra causa del error en la medición, se debe a que los tiempos de conversión de ADC varían en unos pocos microsegundos, esto hace que el tiempo entre muestras no sea uniforme y causa un error.

5. REFERENCIAS

- [1] Min Gyu, Member, IEEE, Gil-Cho Ahn, Member, IEEE, Pavan, Member, IEEE, Sang-Hyeon, Student Member, IEEE, Sang-Ho, Seung-Bin, Jae-Whui Kim, Gabor, Life, IEEE, and Un-Ku, Senior Member, IEEE, "A 0.9 V 92 dB Double-Sampled Switched-RC Delta-Sigma Audio ADC", (Kim, Ahn, Kumar, Lee, Kim, You, Kim, Temes, Fellohw and Moon IEEE Journal of solid-state circuits, vol. 43, no. 5, may 2008).
- [2] YuQing, Terry, and Jacob Abraham, IEEE, "A Single-Die 124 dB Stereo Audio Delta-Sigma ADC With 111 dB THD", (Yang, Sculley and Fellow, IEEE Journal of solid-state circuits, vol. 43, no. 7, July 2008).
- [3] Hideo, Member, IEEE, "Prime-Length Real-Valued Polynomial Residue Division Algorithms", (Murakami, IEEE Transactions on signal processing, vol. 50, no. 11, November 2002).
- [4] Janusz, Senior Member, IEEE, and Philippe, "Subsampling Models and Anti-Alias Filters for 3-D Automultiscopic Displays", (Konrad and Agniel, IEEE Transactions on image processing, vol. 15, no. 1, January 2006).
- [5] Byung G. and Myun., Senior Member IEEE, "New Continuous-Flow Mixed-Radix (CFMR) FFT Processor Using Novel In-Place Strategy", (Jo and Sunwoo, IEEE Transactions on circuits and systems— I: regular papers, vol. 52, no. 5, may 2005).
- [6] Amardar Ashok, "Procesamiento de señales analógicas y digitales", (2da. Edición).
- [7] Stallings William, "Comunicaciones y redes de computadores", (6th. Edición).
- [8] Malvino Albert, J. Bates David, "Principios de electrónica", (7th edición).
- [9] David Báez López, "Matlab con aplicaciones a la ingeniería Física y finanzas", (1ª. Edición).
- [10] Jeff Ferguson, et Al, "La biblia de C#".
- [11] <http://csharp.simpleserial.com/> (28/02/11).