

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DISEÑO E IMPLEMENTACIÓN DE UN FILTRO DIGITAL ADAPTABLE CON LÓGICA DIFUSA

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO ELÉCTRICO ELECTRÓNICO

P R E S E N T A N :

DANIEL CALDERÓN REYES
ROBERTO FERNANDO HUERTA TREJO



DIRECTOR DE TESIS: ING. ROBERTO MACÍAS PÉREZ

MÉXICO, CIUDAD UNIVERSITARIA, 2009

AGRADECIMIENTOS

A mis padres:
Muchas gracias por haberme formado como una persona de principios, por mostrarme su apoyo incondicional, tanto en los momentos de dicha como en los difíciles, y darme las bases necesarias para afrontar la vida con una visión de progreso. Gracias por mostrarme el camino y por impulsarme siempre.

J. David Calderón Calderón:
Gracias papá por el gran esfuerzo que realizas día con día para mostrarme que puedo alcanzar mis metas que me he fijado, siempre aspirando a ser mejor, por brindarme grandes momentos de felicidad con tu compañía y por enseñarme a ver el lado positivo de cada situación. Te quiero mucho papá.

Aurora Reyes Villarreal:
Aunque ya no te encuentras entre nosotros físicamente, agradezco diariamente tus enseñanzas siempre valiosas, por formarme con cariño, dedicación y sabiduría para ser una persona responsable y consiente, por los sacrificios que hiciste y por ser la mejor guía que siempre estará con sus hijos. Te quiero mucho mami.

A mis hermanos:
Gracias por compartir tantas experiencias, por estar a mi lado en los momentos más importantes, por su gran contribución en mi vida y por ser quienes son.

Carlos David Calderón Reyes:
Gracias por toda la ayuda y el apoyo que me has dado, por comprenderme y por tomarme en cuenta siempre.

Karina Aurora Calderón Reyes:
Gracias por la confianza que me has dado, por ser cómplice de innumerables grandes ideas y grandes acciones y por estar al pendiente de mí.

A mis amigos:
Gracias por las experiencias que hemos vivido juntos, por brindarme su amistad sincera, ya que esta es la que hace de nosotros personas completas. Los aprecio por haber dejado huella en mi, su colaboración a lo largo de nuestra formación es digna de mención. Gracias: Adrian, Angélica, César, César David, Franco, Julián, Luis Ángel, Julio César, Luis Enrique, Marcos, Uriel, etc.

Roberto Fernando Huerta Trejo:
Gracias por ser mi amigo, por permitir que nos conozcamos cada vez mejor y por tu colaboración en este trabajo para finalizar exitosamente un ciclo de nuestras vidas.

Ingeniero Roberto Macías Pérez:
Gracias por su amplia experiencia para la dirección de este proyecto, por la comprensión, paciencia y confianza que nos proporcionó a lo largo del desarrollo del mismo. Le agradezco los conocimientos y puntos de vista que me transmitió dentro y fuera del aula, para ser un buen ser humano así como un buen ingeniero.

Facultad de Ingeniería:
Gracias por darme una formación integral en el ámbito académico y humano para desempeñarme orgullosa y responsablemente en el ámbito de la ingeniería.

Universidad Nacional Autónoma de México:
Gracias a mi "Alma Mater" por darme la invaluable oportunidad de aprender de su extensa cultura y diversidad y así formarme como mexicano profesionista con el espíritu y los ideales universitarios para mantener en alto y con orgullo los nombres de México y de la UNAM.

Daniel Calderón Reyes

AGRADECIMIENTOS

Por fortuna no estamos solos mientras recorremos el camino, Dios nos pone personas que nos hacen encontrarle sentido a la vida; honestamente he sido muy afortunado en este aspecto así que no me queda más que agradecer a Dios por ponerlos aquí y a ustedes por permanecer cerca de mí.

Gracias a mi familia por estar siempre y de tan diversas formas conmigo; a mis papás, Susana y Leobardo, quienes además de engendrarme me han entregado buena parte de su vida haciendo que yo pueda tener la mía, gracias por enseñarme que con esfuerzo y trabajo honesto es como se puede salir adelante.

Gracias a mis hermanos, Ame, Arturo y Blanca, por el gran ejemplo que me han dado, me han mostrado los resultados de luchar por ser grandes profesionistas, pero sobretodo me han mostrado lo que es ser grandes seres humanos pese a la adversidad. Los cuatro somos diferentes pero pese a ello estamos siempre tan unidos como si fuésemos uno solo, gracias hermanitos.

Gracias a mi Ewok por esa inocencia, alegría e inteligencia que sólo un pequeño como tu puede tener, gracias por dejarme ser tu tío Beto.

Gracias Arturo Robles por el apoyo que me has brindado en este tiempo de tratarnos, gracias por cuidar a mi hermanita y a mi sobrino.

Gracias Vero por estar ahí, por todas las veces que me has escuchado, apoyado, alegrado, enseñado e inspirado; gracias por todo niña.

Gracias a mis amigos que de una u otra forma han estado conmigo: Mauricio, Máximo, Daniel, Yolanda, Rosa, Rebeca, Emilio, Severo, Campa, Miguel, Hugo, Adán, Huante y Manning.

Gracias Daniel porque sin tu empeño y dedicación esta tesis no podría ser realidad, trabajar juntos ha sido una gran experiencia y gracias a esto nos hemos podido conocer mejor.

Gracias ingeniero Roberto Macías Pérez por toda la paciencia y las enseñanzas que nos ha dado al dirigir la presente tesis, tanto por el aspecto profesional como por el ámbito personal. Gracias por su confianza.

Gracias al Plan de Becas de la Dirección de Sistemas de DGSCA, a la licenciada Laura Liyen Galicia y a la bióloga Susana Torres por permitirme ser parte de esa aventura multicultural en la que conocí a personas extraordinarias. Gracias por creer en mí.

Gracias a la Subdirección de Cómputo de DGIRE, a la matemática Olivia Suárez y la ingeniero Juan Manuel Carriles por la oportunidad que me han dado para desarrollarme profesionalmente, gracias por confiar en mí.

Agradezco también a mi querida Facultad de Ingeniería y a mi muy amada Universidad, mi *Alma Mater*. En sus aulas, jardines, áreas deportivas y culturales tuve el honor y la dicha de crecer integralmente como ser humano. Muchas gracias por estar ahí brindando la oportunidad de crecer a quien así desee hacerlo dentro de sus campus llenos de excelentes profesores, investigadores, trabajadores y estudiantes. Así mismo agradezco a mi país, México, por ser este lugar tan bello lleno de los más increíbles contrastes en donde tengo el orgullo de haber nacido. Siempre me emocionaré y llenaré de dicha y orgullo al escuchar un *Goya* o el Himno Nacional Mexicano.

En la vida hay muchas etapas que nos permiten conocer cómo somos y que nos dejan huella volviéndose parte de nosotros, pero también son etapas que vamos forjando con nuestras acciones y anhelos. Esta tesis, junto con el consecuente examen profesional, marca el final de una etapa muy importante, pero también es punto de inicio de caminos que aún nos faltan por recorrer. Jamás podré retribuir o agradecer lo suficiente pero trabajaré siempre por ello.

Roberto Fernando Huerta Trejo

INTRODUCCIÓN 1

CAPÍTULO 1. PROCESAMIENTO DIGITAL DE AUDIO 3

1.1. Señales 3
1.1.1. Señales analógicas y digitales 3
1.1.2. Señales de audio 4

1.2. Adquisición de las señales (conversión analógica-digital) 4
1.2.1. Transducción 5
1.2.2. Acondicionamiento de señales 6
1.2.3 Muestreo de señales 6
1.2.3.1 Muestreo natural 6
1.2.3.2 Muestreo ideal uniforme 8
1.2.3.3 Muestreo de señales pasabanda 9
1.2.4 Cuantificación 10
1.2.4.1 Cuantificación uniforme 10
1.2.4.2 Cuantificación no uniforme 13

1.3. Correlación y autocorrelación 15

CAPÍTULO 2. FILTRADO ADAPTABLE 18

2.1. Filtros electrónicos y su clasificación 18

2.2. Filtros Adaptables 20
2.2.1 Características de los filtros adaptables 21
2.2.2 Estructuras lineales de filtros adaptables 22
2.2.3. Algoritmos para filtrado adaptable lineal 25
2.2.4. Filtros Adaptables no lineales 27
2.2.5. Aplicaciones de los filtros adaptables 29

2.3. Cancelador adaptable de ruido 31

CAPÍTULO 3. FUNDAMENTOS DE LÓGICA DIFUSA 35

3.1. Información Lingüística 35

3.2. Conceptos básicos de Lógica Difusa 36
3.2.1. El universo del discurso 37
3.2.2. Conjunto difuso y función de pertenencia. 37
3.2.3. Soporte, núcleo y singleton. 39
3.2.4. Intersección, unión y complemento. 39
3.2.5. Relación difusa. 40
3.2.6. Implicaciones difusas. 41
3.2.7. Modo generalizado Ponens y Modo generalizado Tollens 41

3.3. Sistemas difusos 42
3.3.1. Sistema difuso de Takagi y Sugeno 42
3.3.2. Máquina de inferencia difusa con difusor y desdifusor 43

3.3.2.1 Máquina de inferencia difusa 44
3.3.2.2. Difusor 44
3.3.2.3. Desdifusor 45
3.3.2.4. Base de reglas difusa 45

3.3. Consideraciones sobre los sistemas difusos 46

3.5. Sistemas lógicos difusos como aproximadores universales 46

CAPÍTULO 4. IMPLEMENTACIÓN 47

4.1. Descripción del cancelador adaptable de ruido con lógica difusa 48

4.2. Transducción y acondicionamiento de las señales de voz 49
4.2.1. Micrófonos electret 49
4.2.2. Amplificación 51
4.2.3. Conversión Analógico-Digital 53
4.2.4. Comunicación USB 55
4.2.5. Filtro para eliminar componente de directa 56
4.3.1. Correlación cruzada de las señales 63

4.4. Definición de la máquina de inferencia difusa con difusor y desdifusor 65
4.4.1. Máquina de inferencia difusa 65
4.4.2. Difusor 65
4.4.3. Desdifusor 68
4.4.4. Base de Reglas difusas 70
4.4.5. Máquina de inferencia difusa en MATLAB 72

CAPÍTULO 5. PRUEBAS 77

5.1. Prueba 1 77

5.2. Prueba 2 77

5.3. Prueba 3 81

5.4. Prueba 4 84

CONCLUSIONES 88

BIBLIOGRAFÍA 90

APÉNDICE A. UNIVERSAL SERIAL BUS 92

Introducción

Un problema muy común en el mundo de la electrónica es el problema de filtrado con el propósito de eliminar ruido presente en las diversas señales que se emplean en la vida diaria. Particularmente las señales de audio y de voz son señales muy susceptibles a ser contaminadas por ruido de diversas fuentes, entre otras, por otras señales de audio o de voz presentes en el medio ambiente.

El presente proyecto de tesis aborda esta problemática planteando una solución al filtrado de señales, con la estimación y cancelación de ruido presente en el mismo rango de frecuencias que la señal de interés, por medio de estructuras conocidas en el procesamiento digital de señales como filtros adaptables con la variante de que la recursividad de dicha estructura es empleada como punto de partida de un sistema de control difuso, que a partir de una estimación del error y de una derivada del error (PD) modifica los parámetros del filtro adaptable para así eliminar el ruido.

Existen diversos algoritmos recursivos que se integran a los filtros adaptables para reducir el error, sin embargo, actualmente se ha venido aplicando la lógica difusa a la solución de diversos problemas con aplicaciones electrónicas, ya que presenta ventajas sobre soluciones convencionales, destacando que son sistemas no lineales. Por lo anterior se decidió hacer uso de ambas teorías en la solución práctica de un problema real de ingeniería.

El objetivo es por lo tanto, a partir de micrófonos electret, capturar una señal de voz contaminada por ruido presente en el entorno y una estimación del mismo ruido, transferir esta información a la computadora y procesar los datos por medio de un filtro adaptable con lógica difusa.

Para lo anterior es importante considerar la teoría relacionada al procesamiento digital se señales, al filtrado adaptable y a la lógica difusa, y en la práctica es necesario entender la aplicación de dichas teorías con microprocesadores y las etapas de acoplamiento para obtener un sistema completo.

Considerando los requerimientos del proyecto, se propone una solución que implica 3 etapas, la primera, analógica, que permita adquirir las señales de voz y acondicionarlas para la conversión analógico digital; la segunda etapa, realizada por un microcontrolador PIC, que convierta las señales analógicas en digitales y las transfiera a través del puerto USB a una computadora, por último, la tercera etapa, a cargo de una interfaz programada en Visual Basic, que almacene la información de las muestras que envía el PIC y, posteriormente, procese los datos recibidos para obtener la respuesta del filtro implementado con Visual Basic en la computadora.

La estructura de éste trabajo escrito está organizada en cinco capítulos, dónde se proporcionan las bases teóricas necesarias que se deben comprender para la implementación de la solución planteada, el diseño de cada etapa del filtro, las distintas pruebas realizadas con sus respectivos resultados y las conclusiones generales del proyecto.

El capítulo 1 se enfoca en abarcar los conceptos tomados del procesamiento digital de señales y en particular de señales de audio y voz, así como de herramientas matemáticas para su análisis.

El capítulo 2 presenta la teoría de filtrado adaptable, algunas de sus aplicaciones y las consideraciones que se deben realizar para su implementación, como es la capacidad de procesamiento. Se presentan a grandes rasgos algunos de los abundantes algoritmos recursivos empleados frecuentemente en su implementación.

El capítulo 3 abarca los fundamentos de la lógica difusa como parte de las aplicaciones utilizadas en la solución de problemas de ingeniería, que proporciona las razones de su aplicación en el filtrado adaptable en conjunto de las herramientas del procesamiento digital de señales.

El capítulo 4 describe el procedimiento del desarrollo de las etapas que involucra el proyecto, analizando desde la adquisición de datos, el muestreo, la comunicación y el procesamiento con lógica difusa.

El capítulo 5 muestra pruebas que se realizaron con sus respectivas respuestas bajo diversas circunstancias muy particulares.

Capítulo 1
Procesamiento Digital de Audio

Ya que el procesamiento digital de audio es un área de la ingeniería encargada de realizar el tratamiento de señales que se encuentran dentro del rango audible, empleando técnicas digitales, nos proporciona las herramientas fundamentales para analizar distintas etapas del procesamiento en el proyecto de tesis, tanto para la adquisición de las señales, como para la obtención de la señal de salida.

En este capítulo se describe la teoría básica que se debe considerar para llevar a cabo ciertas etapas del proyecto, en la adquisición (transducción, muestreo, cuantización y codificación) y en el procesamiento (correlación), los cuáles se emplearán en las estructuras que veremos en los próximos capítulos.

Ejemplos de las posibles aplicaciones de esta área son: aplicaciones musicales (ecualizadores digitales, efectos de sonido, espacialización del sonido, códecs de audio, étc.), aplicaciones médicas (audífonos para personas con problemas auditivos), así como sistemas de control (reconocedores de voz, étc.), entre otros,

1.1. Señales

Dentro del ámbito electrónico, denominamos señal a toda magnitud eléctrica cuyas variaciones llevan información de interés sobre un proceso o magnitud física.

1.1.1. Señales analógicas y digitales

Las señales cuya amplitud varía de forma continua con el tiempo se denominan analógicas. Aquellas que pueden tomar sólo una serie de valores predefinidos, se denominan señales de amplitud discreta continuas en el tiempo. Las señales que pueden tomar un valor cualquiera pero sólo en instantes de tiempo concretos, se denominan de amplitud continua y discretas en el tiempo.

Las señales que sólo pueden tomar valores de amplitud discretos en instantes concretos, se denominan digitales; su amplitud viene dada por un código que se representa físicamente mediante señales con sólo dos niveles de tensión (1 y 0). La representación del código es lo que se denomina

comúnmente una señal digital. Por extensión, suelen designarse como digitales todas las señales de amplitud discreta, aunque sean continuas en el tiempo.

1.1.2. Señales de audio

Una señal de audio es la representación eléctrica de una señal sonora (ondas de presión de aire), normalmente comprendida dentro del espectro audible.

El espectro audible lo conforma toda la gama de frecuencias que pueden ser percibidas por el oído humano que, para un oído sano y joven, son las frecuencias comprendidas entre los 20 Hz y los 20 kHz. Sin embargo, este margen varía en cada persona y se altera con la edad (presbiacusia).

El siguiente diagrama describe las frecuencias del espectro audible:

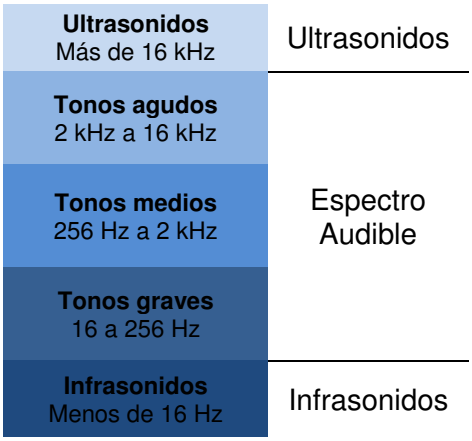


Figura 1.1 Espectro auditivo

Es importante señalar que, particularmente, las señales de voz humana (señal vocal) no suelen tener información relevante más allá de los 10 kHz; asimismo, como en telefonía fija, se pueden considerar solamente los primeros 4 kHz. De hecho, basta considerar únicamente 2kHz para que la voz sea comprensible aunque no se pueda reconocer al hablante.

1.2. Adquisición de las señales (conversión analógica-digital)

En un sistema de procesamiento de audio hay que adquirir la información (en forma de señales analógicas o digitales), procesarla y presentarla. A veces, además, hay que registrarla. Como primera etapa de este proceso se requiere un elemento que actúe como interfaz entre el medio de la señal original de interés y el sistema de procesamiento, a esto comúnmente se le conoce como transductor o sensor de entrada y funciona, frecuentemente, en conjunto con otros elementos que adecúan la señal para que ésta pueda ser suministrada como entrada al siguiente elemento del sistema, este acondicionamiento puede hacerse mediante filtros y amplificadores.

En el caso de que la información se presente como una señal analógica, como en el caso del presente proyecto, se requiere realizar una conversión a su equivalente digital puesto que el procesamiento será de este tipo. La conversión analógico-digital (A/D) se encarga de convertir un tipo de señales en otro, y debe hacerlo conservando la información. Es decir, idealmente debiera ser

posible, a partir de la salida digital, reconstruir fielmente la señal analógica de entrada, para ello se requiere considerar tres procesos distintos: el muestreo, la cuantificación y la codificación.

El muestreo obtiene, a partir de la señal continua de entrada, una señal discreta en el tiempo y continua en el dominio de las amplitudes, es decir, una serie temporal. La cuantificación es una operación no lineal que convierte una señal de amplitud continua en otra de amplitud discreta, es decir, con un número finito de valores o niveles de amplitud. La codificación es el proceso de asignar una palabra (o combinación) de un código digital a cada uno de una serie de niveles de tensión discretos.

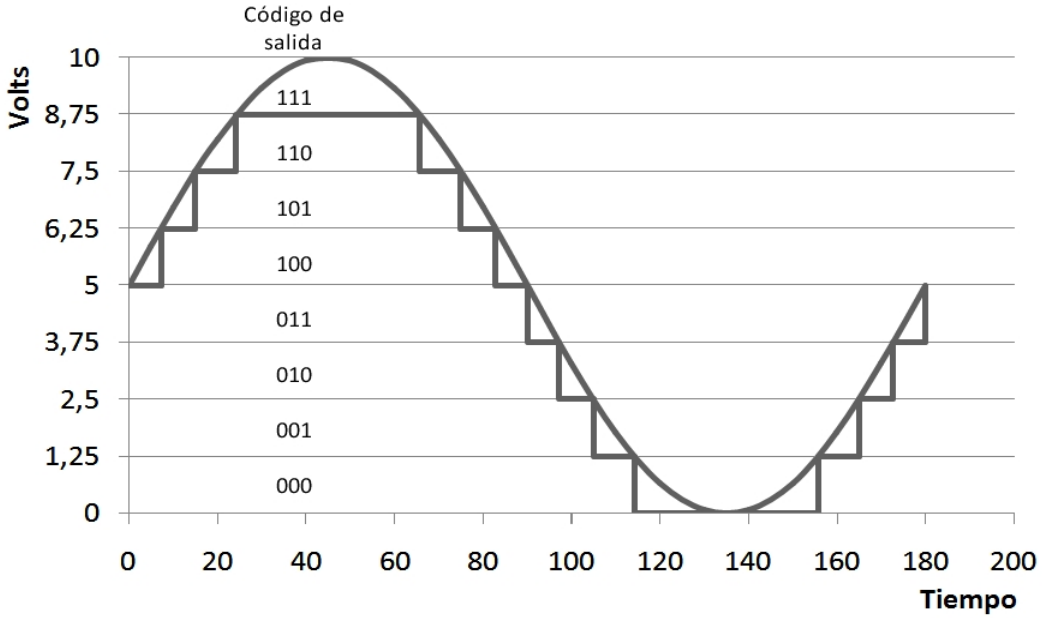


Figura 1.2 Señal senoidal y su versión digital con una resolución de 3 bits.

En el caso de la figura 1.2 hay 8 estados de salida, representados por los códigos desde 000 hasta 111. Si el margen *M* de tensiones de entrada del CAD es de 0 a 10 V, cada código representa un intervalo de amplitudes de 10 V/8 =1,25 V.

1.2.1. Transducción

Un transductor convierte energía de una forma física en otra forma distinta; en nuestro caso convierte energía mecánica en energía eléctrica.

El primer elemento del sistema para procesar audio a partir de una señal sonora analógica es el que detecta la magnitud de interés. Dicha magnitud es mecánica por lo que se requiere un elemento que nos permita tomar su valor y lo traduzca a una magnitud eléctrica, para tal fin se utiliza un transductor de entrada (un micrófono o un conjunto de ellos).

Los parámetros básicos de un sensor son: margen o campo de medida, sensibilidad, resolución y exactitud. La sensibilidad es la relación entre la magnitud de salida y la de entrada, en ausencia de errores. Si la sensibilidad es constante en todo el margen de medida, se dice que el sensor es lineal. La resolución es el menor cambio en la magnitud de entrada que se puede detectar a la salida y está limitada en último término por las fluctuaciones aleatorias presentes en todo elemento (mecánico o eléctrico) que disipe energía, como es el caso de los sensores.

La exactitud describe el grado de coincidencia entre el valor real de la entrada y su valor deducido a partir de la salida actual y de la sensibilidad.

La salida de los sensores es normalmente, aunque no exclusivamente, en forma de señal continua en cuanto a amplitud y en el tiempo (analógica). En algún punto del sistema, dicha señal tiene que ser convertida en digital, para realizar alguna operación digital sobre la señal, esto será abordado más adelante.

1.2.2. Acondicionamiento de señales

Los circuitos de acondicionamiento de la señal de entrada se encargan de la amplificación, filtrado y adaptación de la señal del sensor al Convertidor Analógico Digital (CAD) u otro tipo de etapa posterior.

Dado que manejan sólo señales eléctricas sin convertirlas en otras formas físicas, los acondicionadores de señal constituyen lo que se denomina una interfaz. Otras interfaces del sistema son las necesarias para enviar la información a través de los canales o líneas de comunicación, en nuestro caso a través del puerto USB de la computadora. Su función y estructura dependen tanto del medio de comunicación como del código empleado para el mensaje.

1.2.3 Muestreo de señales

El muestreo es un proceso mediante el que se transforma una señal continua en el tiempo y de banda limitada, en una sucesión (continua) de valores de amplitud en instantes discretos, que constituyen la denominada señal muestreada o serie temporal. Esta transformación se puede hacer de distintas formas, pero lo más frecuente es que la señal muestreada esté formada por los valores de la señal original en instantes de tiempo equiespaciados, en este caso se denomina muestreo uniforme.

El muestreo no es un proceso exclusivo de la conversión A/D sino que también está presente, por ejemplo, en sistemas de multiplexado por división del tiempo (TDM, *time-division multiplexing*) y en la comunicación por impulsos modulados en amplitud (PAM, *pulse-amplitude modulation*).

1.2.3.1 Muestreo natural

La figura 1.3 es un modelo gráfico del muestreo uniforme, cuando se considera que la señal muestreadora, *m(t),* es un tren de pulsos de amplitud unidad y duración finita τ, separados un tiempo *T,* que se denomina periodo de muestreo [1].

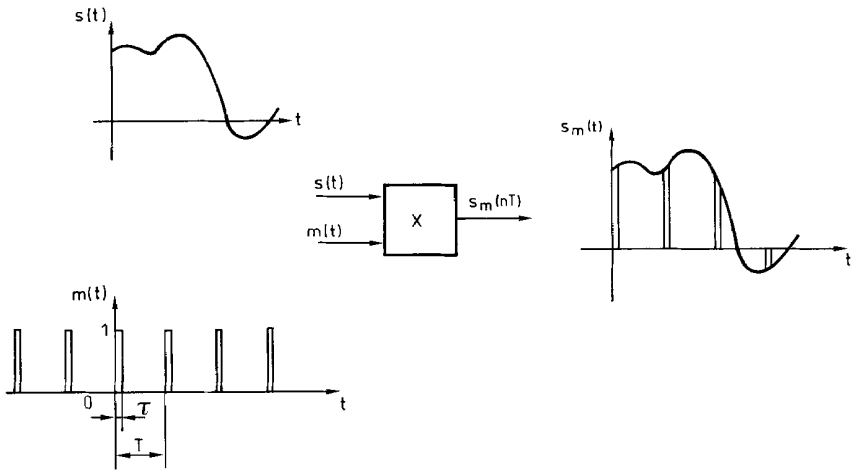


Figura 1.3 Modelo gráfico del proceso de muestreo

La señal a muestrear *s(t)* es de banda limitada, con máxima componente frecuencial f_M y mínima 0 (o $-f_M$ en su descripción mediante análisis de Fourier). La señal muestreada, $s_m(nT)$ (*n* entero), es un tren de pulsos modulados en amplitud, es decir, es discreta en el tiempo pero continua en amplitud, y puede interpretarse como el producto de la señal de entrada por la muestreadora,

$$s_m(nT) = s(t)\, m(t) \tag{1.1}$$

El teorema del muestreo de Nyquist-Shannon establece la relación que debe haber entre f_M y T siendo dicha relación la siguiente:

$$T < 1/2f_M \tag{1.2}$$

De lo anterior se desprende que la frecuencia de muestreo mínima que tenemos que utilizar debe ser mayor que 2 veces la frecuencia más alta contenida en la señal analógica (f_M), a este criterio también se le conoce como criterio de Nyquist.

Cuanto menor sea T con respecto a $1/2f_M$ más separadas quedan las bandas de $s_m(f)$ y, por lo tanto, más relajadas pueden ser las características de atenuación del filtro de reconstrucción de la señal. Si, por el contrario, $T > 1/2\, f_M$ entonces las bandas se solapan, fenómeno que se denomina *aliasing* en inglés (las muestras tomadas representan una señal distinta de la original) y que supone un mezclado heterodino que da origen a una frecuencia inexistente, en principio, en la señal de entrada. O bien, si resulta que en la señal de entrada ya había componentes de frecuencia igual a la que aparece por *aliasing*, entonces resulta que las componentes ficticias enmascaran a las de interés.

En la figura 1.4 se presentan dos situaciones distintas al muestrear una senoide de periodo T_s primero con un periodo $T < T_s/5$ y luego con un periodo $T > T_s/2$. Puede verse cómo, en este segundo caso, la señal reconstruida a partir de las muestras tomadas no coincide con la original.

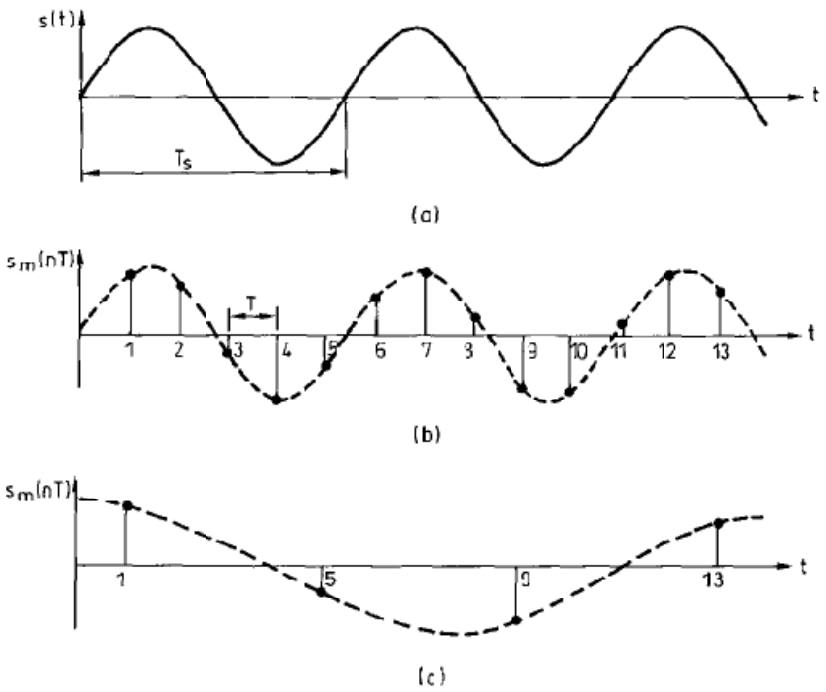


Figura 1.4 Muestreo de una señal senoidal s(t) de periodo *Ts* (a), con un tren de impulsos de periodo *T < Ts/5* (b), Y con un tren de impulsos de periodo *T> Ts/2* (c). Obsérvese cómo, en este segundo caso, las muestras tomadas sugieren que la señal de partida era de frecuencia distinta a la real

Para evitar el mezclado heterodino hay que poner un filtro paso bajas *antialiasing* antes del muestreador. El orden de este filtro se elige de forma que las componentes no deseadas presentes en la señal de entrada, y que estén por encima de la mitad de la frecuencia de muestreo, queden reducidas a una amplitud menor que la correspondiente, por ejemplo, a la resolución del CAD o al valor máximo del error de cuantificación. Cuanto mayor sea la frecuencia de muestreo con respecto a la frecuencia de las señales no deseadas, menor puede ser el orden del filtro de paso bajo.

1.2.3.2 Muestreo ideal uniforme

Algunos CAD exigen que la señal aplicada a su entrada permanezca constante durante el tiempo de conversión, de modo que no es posible que su entrada sea transparente a la señal aplicada, como sucede en el muestreo natural. Para que la entrada del CAD permanezca constante, se le antepone un circuito de muestreo y retención (S&H) que toma una muestra en un tiempo muy breve y la retiene durante el tiempo necesario para efectuar su conversión. En una primera aproximación, se supone entonces que el muestreo es instantáneo (ideal).

El teorema del muestreo, o de Shannon [2], afirma que si s*(t)* es una señal cuya transformada de Fourier s(f) es tal que s(f) ≡ 0 para toda frecuencia | *f* | >= *fM,* entonces:

$$s(t) = \sum_{n=-\infty}^{n=\infty} s(nT)\frac{sen\ 2\pi\ f_M(t-nT)}{2\pi\ f_M(t-nT)} \qquad (1.3a)$$

donde *T = 1/2 f_M. f_M* es la denominada frecuencia de Nyquist y *1/T* es la frecuencia de muestreo de Nyquist (Nyquist rate).

Es decir, *s(t)* se puede reconstruir a partir de *s(nT),* que es la secuencia de sus muestras instantáneas, empleando la fórmula de interpolación (1.3a), que se denomina serie cardinal.

En la práctica se muestrea a una frecuencia de 7 a 10 veces mayor que la mínima y se reconstruye la señal con un número finito de muestras. Una expresión equivalente a (1.3a) pero más fácil de calcular es:

$$s(t) = \frac{1}{\pi} sen\,(2\pi f_M t) \sum_{n=-\infty}^{n=\infty} \frac{(-1)^n s(nT)}{2 f_M t - n} \qquad (1.3b)$$

El teorema del muestreo no dice que la única forma de recuperar la señal *s(t)* sea tal como indica (1.3a). Sólo afirma que *s(t)* se puede recuperar de dicha forma si se toman las muestras a frecuencia mayor que $2f_M$. Así, por ejemplo, también es posible reconstruir *s(t)* tomando muestras con menor frecuencia, siempre y cuando se conozca además el valor de las derivadas de *s(t)* en el instante de muestreo. Si se conocen los valores de las *k* primeras derivadas, es posible entonces tomar muestras separadas $T = (k + 1)/2 f_M$ y aún se puede reconstruir la señal [3].

1.2.3.3 Muestreo de señales pasabanda

Cuando se trata de una señal en forma de portadora modulada en amplitud por la magnitud de interés, de frecuencia mucho menor que la de la portadora, el objetivo es en estos casos reconstruir la información, no la portadora.

De acuerdo a O.D. Grace y S.P. Pitt [6] *s(t)* se puede reconstruir a partir de muestras de sus componentes en fase y en cuadratura, *p(t)* y *q(t),* muestreadas a una frecuencia determinada por su ancho de banda $(f_2 - f_1)$ no por la máxima frecuencia presente.

En muchos casos lo que interesa es obtener precisamente las componentes en fase y en cuadratura. Para cualquier señal limitada a la banda de frecuencias $f_c - B/2 \leq f \leq f_c + B/2$, es posible obtener las componentes en cuadratura *p(t)* y *q(t)* a partir de muestras tomadas de *s(t),* sin procesado previo (es decir, sin 'hacer la transformada de Hilbert), mediante las fórmulas de interpolación siguientes:

$$p(t) = T B \sum_{n=-\infty}^{\infty} (-1)^{nk} s(nT)\, sinc\, B(t - nT)$$

$$q(t) = T B \sum_{n=-\infty}^{\infty} (-1)^{nk} s(nT + T_c/4 + mT)\, sinc\, B(t - nT - T_c/4 - mT)$$

donde $T = k/2f_c = kT_c/2 \leq 1/B,$ siendo *k* cualquier entero, $B \leq f_c$ y $m = 0, \pm 1, \pm 2$, etcétera

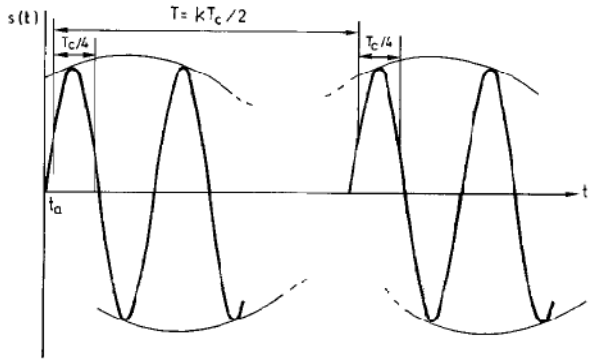


Figura 1.5 Muestreo síncrono de una señal para obtener sus dos componentes, en fase y cuadratura.

Se trata, pues, de tomar las muestras en sincronismo con la portadora *(T* debe ser un múltiplo de *1/2f_c.).* Cada par de muestras debe estar separado por un cuarto del periodo de la portadora, $T_c/4$, (y por varios ciclos de muestreo si se desea, haciendo $m \neq 0$). En la figura 1.5 se ilustran estas relaciones. El muestreo a una frecuencia submúltiplo de la portadora produce solapamientos de bandas, y la banda que cae en el origen de frecuencias es precisamente la que se recupera. Para que la señal quede efectivamente limitada a la banda de frecuencias $f_c - B/2 \leq f \leq f_c + B/2$, conviene disponer un filtro pasabanda antes del muestreador. No basta el filtro de paso bajo habitual (*antialiasing*) antes del muestreador.

Esta forma de muestreo, síncrona, se denomina muestreo en cuadratura porque permite obtener las dos componentes de la señal.

Es muy importante observar que este tipo de submuestreo permite utilizar un CAD que tenga un tiempo de conversión grande, mayor por ejemplo que el periodo de la portadora, pero inferior al periodo de la información (señal moduladora). Ahora bien, la muestra hay que tomarla de una señal que cambia rápidamente y, por lo tanto, el amplificador S&H debe ser rápido.

1.2.4 Cuantificación

La cuantificación de una señal consiste en representarla mediante una serie finita de niveles de amplitud o estados de salida. A diferencia del muestreo, la cuantificación es un proceso no lineal. En la adquisición de señales, la cuantificación la realiza el CAD. En estos casos la señal de partida es a veces, pero no siempre, una señal muestreada, es decir, discreta en el tiempo.

1.2.4.1 Cuantificación uniforme

En la figura 1.6a se representa gráficamente un proceso de cuantificación uniforme. En este ejemplo, al conjunto de valores de la entrada entre 0 y $q/2$ ($0 \leq x < q/2$) se les ha asignado el nivel o estado 1; a los valores entre $q/2$ y $3q/2$ ($q/2 \leq x < 3q/2$), el estado 2; a los valores entre $3q/2$ y $5q/2$ ($3q/2 \leq x < 5q/2$) el estado 3; y, en general, a los valores entre $(k - 3/2)q$ y $(k - 1/2)q$ se les ha asignado el estado k. El último estado no incluye el valor de entrada $2nq$, sino que corresponde al intervalo $(2^n - 3/2)\, q \leq x < (2^n - 1/2)q$.

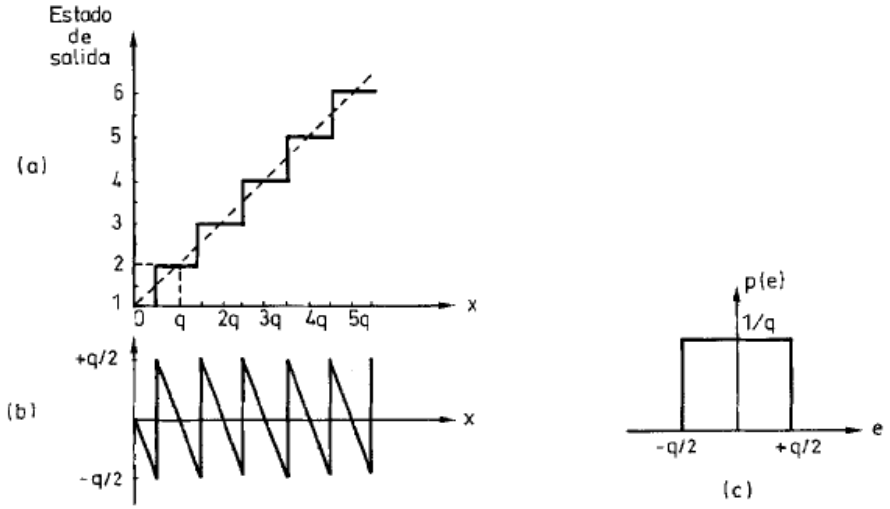


Figura 1.6 (a) Cuantificación lineal. (b) Error de cuantificación y (c) su función de densidad de probabilidad.

El valor *q* se denomina intervalo de cuantificación, y coincide con la diferencia entre el mayor y el menor valor de la entrada a los que se les asigna el mismo estado de salida. A este tipo de cuantificadores, en los que el estado de salida viene determinado exclusivamente por un valor de la entrada, pero no por los valores pasados ni por los futuros, se les denomina de memoria cero, y son los más frecuentes.

En los cuantificadores secuenciales, por ejemplo, el estado de salida depende de la entrada presente y de las anteriores, por lo que son un tipo de codificadores con memoria.

El número de estados de salida expresado en bits *(n)* determina la resolución del cuantificador; $2^n = N$ es el número de estados.

La diferencia entre el mayor y el menor valor aceptados a la entrada se denomina margen de entrada, *M.* En la cuantificación uniforme *q* es constante en todo el margen *M,* y se cumple que:

$$q = \frac{M}{2^n} \tag{1.4}$$

En este caso hay 2^n - 1 puntos de decisión en el margen *M* para asignar un estado a la señal de entrada. (Cuando a partir de dos valores de entrada extremos se asigna el mismo estado a todas las entradas, entonces hay 2^n + 1 puntos de decisión).

En el cuantificador de la figura 1.6a los puntos de decisión están en el centro de cada intervalo.

La cuantificación introduce inevitablemente un error, por cuanto si se intenta reconstruir la entrada a partir de la salida del cuantificador, no se obtiene el continuo de valores del margen *M* o, en definitiva la relación $y = x$, sino la curva en escalera indicada en la figura 1.6a. El máximo error cometido es $\pm q/2$, y su evolución en función del valor de la entrada tiene forma de diente de sierra (figura 1.6b*).*

Si todos los valores del margen *M* tienen igual probabilidad, el error de cuantificación se puede interpretar como si la entrada tuviera superpuesto un ruido aleatorio con función de densidad de probabilidad rectangular (figura 1.5c), y que se denomina ruido de cuantificación. Su valor medio para un cuantificador con una característica de transferencia como la de la figura 1.5a (y suponiendo que todos los valores de entrada tengan la misma probabilidad) es:

$$\bar{e} = E\{e\} = \int_{-\infty}^{\infty} e\, p(e)\, de = \int_{-\infty}^{\infty} e\, \frac{1}{q}\, de = \frac{1}{q} \int_{-q/2}^{q/2} -x\, dx = 0$$

donde E{ }es el operador esperanza matemática. La variancia será:

$$var\{e\} = E\{(e-\bar{e})^2\} = E\{e^2\} - E^2\{\bar{e}\} =$$
$$= \int_{-\infty}^{\infty} e^2 \frac{1}{q} de = \frac{1}{q} \int_{-q/2}^{q/2} x^2 dx = \frac{q^2}{12} \qquad (1.5)$$

El valor eficaz del ruido es la raíz cuadrada de la suma de la media al cuadrado y la variancia, y será $q/2\sqrt{3}$. Resulta, pues, que para reducir el valor eficaz de este ruido hay que reducir proporcionalmente el intervalo de cuantificación. Esta reducción viene limitada por factores tecnológicos y, en último término, por el ruido eléctrico asociado a los componentes electrónicos.

La acción del cuantificador se puede describir en términos de relación señal-ruido (S/N), definida como el cociente entre la amplitud de la señal deseada y la amplitud del ruido presente. Suele expresarse en decibeles. Cuando se trata de ruido aleatorio, ambas amplitudes se dan en valor eficaz. Así, para el caso de una señal senoidal con amplitud entre 0 y $(2^n - 1/2)q$ (es decir, unipolar), su valor cuadrático medio es:

$$\frac{1}{2\pi} \int_0^{2\pi} (2^n - 1/2)^2\, q^2\, sen^2 \alpha\; d\alpha = \frac{(2^n - 1/2)^2\, q^2}{2}$$

La relación S/N será:

$$S/N = \frac{(2^n - 1/2)^2\, q^2/2}{q^2/12} = 6(2^n - 1/2)^2 \approx 6 \times 2^{2n} \text{ cuando n>4.}$$
$$\text{S/N [dB]} \approx 10 \log 6 + 20n \log 2 = 7.78 + 6.02n \qquad (1.6)$$

Este valor es para señales senoidales con amplitud de pico igual al margen de entrada del cuantificador. Para señales de menor amplitud, *S/N* será más pequeño. Por esto es importante amplificar la señal para que su margen de variación coincida con *M*.

Cuando la señal de entrada es aleatoria, su amplitud puede tomar en principio cualquier valor entre $+\infty$ y $-\infty$, y viene caracterizada por su función de densidad de probabilidad, $p(x)$ [4]. La bondad del cuantificador se describe entonces mediante la denominada distorsión cuadrática media, *D*

$$D = \int_{-\infty}^{\infty} [C(x) - x]^2\, p(x) dx \qquad (1.7a)$$

donde $C(x)$ es la relación entrada-salida del cuantificador. Si el número de estados de salida *N* es elevado, la expresión anterior se puede sustituir por:

$$D = \sum_{i=1}^{N} \int_{x_{i-1}}^{x_i} (y_i - x)^2\, p(x)\, dx \qquad (1.7b)$$

donde $y_i = C(x)$ para toda x_i que caiga dentro de la región R_i entre x_{i-1} y x_i. Al ser *N* grande, cada intervalo de valores asignados a un mismo estado es muy pequeño (salvo los dos extremos, denominadas zonas de sobrecarga), de forma que $p(x)$ se puede considerar constante en cada región R_i. Si tomamos $p(x) \approx p(y_i)$ en cada región R_i, excepto en las dos extremas donde suponemos $p(x) \approx 0$, y llamando $q_i = x_i - x_{i-1}$, si $y_i \approx (x_i + x_{i-1})/2$, queda:

$$D = \sum_{i=2}^{N-1} p(y_i) \int_{x_{i-1}}^{x_i} (y_i - x)^2 \, dx = \sum_{i=2}^{N-1} p(y_i) \left[\frac{-(y_i - x_i)^3 + (y_i - x_{i-1})^3}{3} \right]$$

$$D = \frac{1}{12} \sum_{i=2}^{N-1} p(y_i) \, q_i^3 \tag{1.8}$$

Para cuantificación uniforme, q_i es la misma para todas las regiones, $q_i = q$, de modo que tendremos lo siguiente:

$$D = \frac{q^2}{12} \sum_{i=2}^{N-1} p(y_i) \, q \tag{1.9}$$

Dado que:

$$\sum p(y_i) \, q \approx \int p(u) \, du = 1$$

queda finalmente:

$$D \approx \frac{q^2}{12} \tag{1.10}$$

que coincide con (1.5), obtenida sin considerar el ruido de sobrecarga (en las zonas extremas del margen de entrada).

La relación *S/N* para el cuantificador cuando su entrada es aleatoria será, en general:

$$S \Big/ N \, [dB] = 10 \, log \frac{var[x]}{D} \tag{1.11}$$

1.2.4.2 Cuantificación no uniforme

La cuantificación uniforme no es siempre la opción más eficiente. Basta observar, por ejemplo, que mientras el error absoluto de cuantificación es constante de unos a otros intervalos de cuantificación, el error relativo será en cambio mucho mayor para las amplitudes de entrada pequeñas. En consecuencia, *S/N* es menor para las pequeñas amplitudes, que en muchos casos son precisamente las que se dan más a menudo. Si se desea mantener *S/N* constante en un amplio margen de amplitudes (cuantificación robusta), entonces se puede variar, por ejemplo, el intervalo de cuantificación proporcionalmente a la amplitud de entrada.

El interés de *S/N* constante en un amplio margen se presenta, por ejemplo, en la transmisión de voz por Modulación de Pulsos Codificados (*Pulse Code Modulation* o PCM en inglés). Una cuantificación uniforme aportaría una mayor degradación de la señal de voz de quienes hablan más bajo. Como resulta que el oído es capaz de distinguir señales inmersas en un nivel de ruido elevado siempre y cuando las señales sean también de amplitud elevada, no importa que el ruido de cuantificación sea mayor para señales grandes. Pero, además, la cuantificación no uniforme permite reducir el número de bits utilizados.

La cuantificación no uniforme se puede describir con el modelo de la figura 1.7.

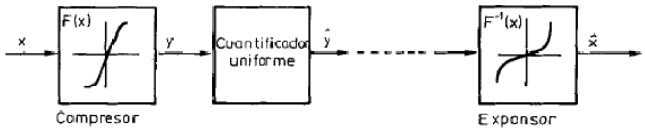


Figura 1.7 Modelo de la cuantificación no uniforme que la describe mediante una compresión, una cuantificación uniforme y una expansión [4].

El cuantificador se descompone en dos bloques: un compresor *F(x)* no lineal y un cuantificador lineal (uniforme, relación ideal *y = x).* La función *F(x)* es monótona creciente con simetría impar, toma valores entre *-V* y *+V,* y cumple *F(V) = V y F(0)* = 0. Por ser *F(x)* monótona, se puede invertir, de manera que si se aplica $F^{-1}(x)$ a la salida del compresor se recupera la entrada *x* y, por lo tanto, la compresión no acarrea ninguna pérdida de información. $F^{-1}(x)$ es el denominado expansor.

El cuantificador uniforme se elige con *N* - 2 (≈ *N)* intervalos, dejando fuera las zonas de sobrecarga, de manera que *q* =2 *V/N.* En la figura 1.7 se muestra la relación entre los niveles de decisión del cuantificador y los intervalos de valores de la entrada. El conjunto de las dos operaciones compresión-expansión se denomina en inglés *companding* (*COMPressing - expANDING*); una posible traducción equivalente puede ser compansión, aunque se presta a equívocos.

Una forma de aplicar el método de la figura 1.7 es utilizar como compresor un amplificador logarítmico, como muestra la figura 1.8, y luego un cuantificador lineal [5]. Al aplicar una entrada v_a a un amplificador logarítmico, la salida es

$$v_o = k \log \frac{v_a}{v_r} \tag{1.12}$$

donde v_r es una tensión de referencia. Si el cuantificador lineal tiene *n* bits y su margen de entrada es *M,* deberá cumplirse:

$$v_{omax} - v_{omin} = M = q\,2^n \tag{1.13}$$

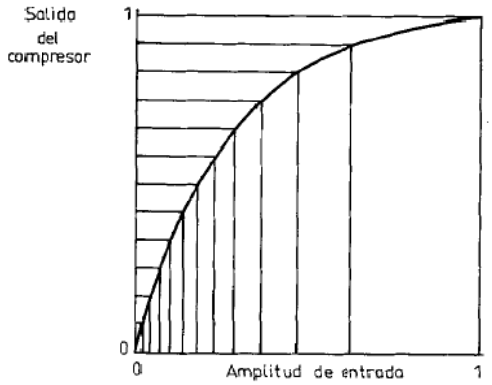


Figura 1.8 Correspondencia entre los niveles de decisión y la salida del compresor (limitado a amplitudes positivas).

El intervalo de cuantificación ΔV equivalente a la entrada será la diferencia entre las tensiones que a la salida del amplificador produzcan tensiones que difieran en *q.* Es decir,

$$v_o = k \log \frac{v_a}{v_r} \tag{1.14a}$$

$$v_o + q = k \log \frac{v_a + \Delta V}{v_r} \tag{1.14b}$$

Restando (1.14a) de (1.14b) obtenemos:

$$q = k \log \frac{v_a + \Delta V}{v_a} = \frac{k}{2.3} ln\left(1 + \frac{\Delta V}{v_a}\right) \tag{1.15}$$

Dado que cuando $x \ll 1$, $\ln(1 + x) \approx x$, si $\Delta V \ll v_a$ tendremos:

$$\Delta V = \frac{2.3\, v_a}{k} q \tag{1.16}$$

Es decir, la amplitud del intervalo de cuantificación equivalente a la entrada es proporcional a la amplitud de la entrada: será menor para entradas pequeñas que para entradas grandes.

Ahora bien, los amplificadores logarítmicos sólo aceptan entradas unipolares (el logaritmo de una cantidad negativa no está definido) y requieren muchos ajustes. Es preferible por ello emplear un cuantificador no lineal junto con un codificador apropiado.

La mejora obtenida con respecto al caso de cuantificación uniforme en todo el margen se evalúa con el denominado factor de ventaja de la compansión (companding advantage):

$$c_A = \frac{q}{q_0} = g(0) \tag{1.17}$$

que es un criterio de comparación de distintos compresores.

Existe un tipo particular de cuantificadores no uniformes denominados cuantificadores robustos, que ofrecen *S/N* constante en todo el margen de entrada, con independencia de la función de densidad de probabilidad de la señal pero requiere un compresor logarítmico que no es realizable; sin embargo se utilizan las leyes mu (en Norteamérica y Japón) o la ley A (en Europa).

1.3. Correlación y autocorrelación

El concepto de correlación es especialmente importante en el procesamiento digital de señales, ya que permite, junto con el concepto de densidad espectral, unificar el estudio frecuencial de las señales deterministas y las señales aleatorias. Además, está íntimamente relacionado con el concepto de distancia entre señales (o distintos segmentos de una misma señal).

El análisis de correlación es el conjunto de técnicas empleado para medir la intensidad de la asociación entre dos variables, esto es, nos permite cuantificar el grado de similitud entre varias señales o procesos. Algunas de sus aplicaciones se encuentran en la biología para encontrar la similitud entre las características propias de diversas regiones permitiendo predecir dónde podría encontrarse alguna especie animal o vegetal; dentro de los sistemas de radar y sonar se utiliza para la localización de la posición y velocidad de los objetos de interés.

Definición formal.

Sean x[n] e y[n] dos secuencias con energía finita:

$$E_x = \sum_{n=-\infty}^{\infty} |x[n]|^2 < \infty \qquad\qquad E_y = \sum_{n=-\infty}^{\infty} |y[n]|^2 < \infty$$

Una definición de distancia entre y[n] y una versión desplazada de la otra señal x[n+m] es la energía de su diferencia:

$$D[m] = \sum_{n=-\infty}^{\infty} |x[n+m] - y[n]|^2 = \sum_{n=-\infty}^{\infty} |x[n] - y[n-m]|^2 = E_x + E_y - (r_{xy[m]} + r_{yx[-m]})$$

donde

$$r_{xy[m]} = \sum_{n=-\infty}^{\infty} x[n+m]y^*[n] = \sum_{n=-\infty}^{\infty} x[n]y^*[n-m] = x[m] * y^*[-m] \quad (1.18)$$

se denomina *correlación cruzada* de las dos secuencias. Fijada la energía de las mismas, su parecido será mayor cuanto mayor sea la suma $r_{xy}[m] + r_{yx}[-m]$. En la definición (1.18) se comprueba sin dificultad que

$$r_{xy}[m] = r_{yx}^*[-m] \quad (1.19)$$

Así, para secuencias reales, cuya correlación cruzada también es real, se puede escribir:

$$D[m] = E_x + E_y - 2\, r_{xy}[m] \quad (1.20)$$

lo que indica una menor distancia entre secuencias cuanto mayor sea su correlación cruzada.

De la definición 1.18 se puede inferir que cuando las dos secuencias sean parecidas la suma de los productos tenderá a incrementar este valor. Valores grandes y positivos indicarán que ambas señales son parecidas y crecen a la vez, mientras que valores negativos indican que el crecimiento de una variable está asociado al decrecimiento de la otra. Además, valores cercanos a cero indican que las señales no tienen parecido ya que las contribuciones negativas y positivas tenderán a cancelarse.

Existe un inconveniente con la definición matemática de la correlación arriba expuesta ya que dadas dos señales de igual forma pero de magnitudes diferentes se obtendrán distintos valores de correlación dependiendo de las magnitudes de dichas señales. Para resolver este problema puede usarse la definición del *coeficiente de correlación normalizado*, que consiste en dividir el valor obtenido de la ecuación de correlación entre la energía de las señales involucradas:

$$\rho_{xy}(m) = \frac{\sum_{n=0}^{N-1} x(n)y(n-m)}{[\sum_{n=0}^{N-1} x^2(n) \sum_{n=0}^{N-1} y^2(n)]^{\frac{1}{2}}}$$

Con lo anterior el coeficiente de correlación solo toma valores dentro del intervalo [-1,1].

Para aclarar el concepto de correlación se presenta el siguiente ejemplo tomado de José B. Mariño Acebal [8]

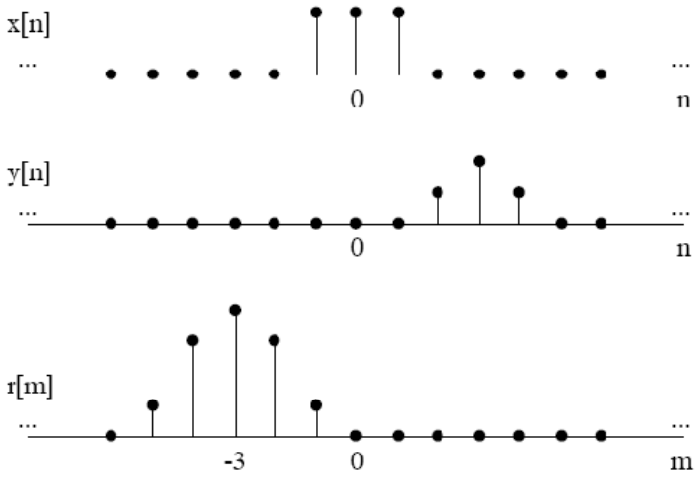


Figura 1.9. Secuencias y correlación cruzada.

Considérense las secuencias x[n] e y[n] de la figura anterior, cuya correlación cruzada se proporciona en la misma figura. El valor máximo de ésta ocurre para la muestra con el ordinal m = - 3, lo que sugiere que, si x[n] se retrasa 3 muestras, se obtendrá la máxima coincidencia entre ambas secuencias. Este resultado es acorde con la intuición, como se comprueba examinando las señales x[n] e y[n].

Existen situaciones en las que el interés es comparar una secuencia consigo misma, en este caso la correlación se denomina *autocorrelación* y se utiliza la siguiente notación:

$$r_{xx}[m] = x[m] * x^{*}[-m] = r_x[m] = r[m]$$

Cuando la secuencia es real, se puede hacer uso del siguiente argumento cualitativo: de acuerdo con la ecuación 1.18 la autocorrelación será máxima cuando la distancia entre la secuencia y una versión desplazada de la misma sea mínima; es claro que tal mínimo se produce cuando el desplazamiento es nulo (m = 0).

En la siguiente tabla se presentan algunas de las propiedades de la autocorrelación:

Energía	$r[0] = E_x$
Simetría	$r[m] = r^*[-m]$
x[n] real	r[m] real r[m] = r[-m] (par)
$x[n]e^{j\omega_0 n}$	$r[m]\ e^{j\omega_0 m}$
Máximo	$\lvert r[m]\rvert \leq r[0]$

Tabla 1. Propiedades de la autocorrelación

Capítulo 2
Filtrado Adaptable

La estructura fundamental del presente proyecto está basada en la teoría del filtrado adaptable, por lo que es necesario comprender inicialmente el concepto de filtro electrónico y los distintos tipos que existen. El presente capítulo describe en general las características de los filtros y se profundiza particularmente en los filtros adaptables y sus aplicaciones.

2.1. Filtros electrónicos y su clasificación

El término filtro se usa comúnmente para describir un dispositivo físico, ya sea puramente de hardware o con software, que procesa un conjunto de datos (con ruido) para extraer información de interés, discriminando una determinada frecuencia o gama de frecuencias de una señal, modificándola tanto en amplitud como en fase.

A partir de la función de transferencia de un filtro, se puede obtener la respuesta en frecuencia del mismo, lo que ofrece ciertas características deseables del filtro, como son, banda de paso, banda de transición, banda de corte o rechazo y desfasamiento.

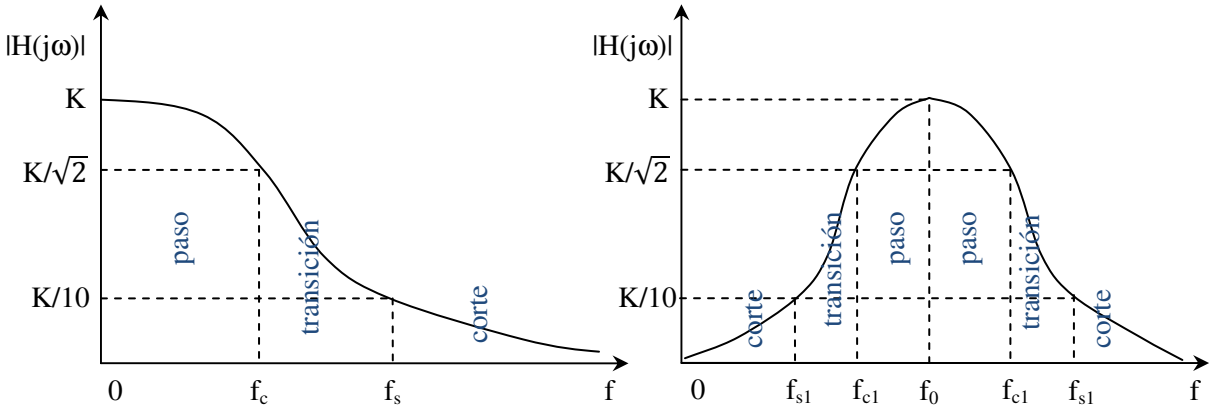


Figura 2.1. Bandas de paso, transición y corte de filtros

Si se fija una característica que debe tener el filtro, es posible determinar la función de transferencia del filtro, esto es, se pueden determinar los polos y ceros de los polinomios del numerador y del denominador de la función de transferencia, así se definen algunos filtros comúnmente usados, como:

- Filtro Butterworth. Este filtro tiene una respuesta aproximadamente lineal en la banda de paso y con una banda de transición bastante extensa, ya que decae suavemente en esta banda.
- Filtro Chevyshev. Este filtro tiene un corte abrupto, pero en la banda de paso tiene una respuesta ondulatoria.
- Filtro elíptico. Este filtro cuenta con una transición más abrupta que los anteriores pero presenta una respuesta ondulatoria tanto en la banda de paso como en la banda de rechazo.
- Filtro de Bessel. Este filtro presenta una respuesta lineal en las variaciones de la fase.

De acuerdo a los componentes físicos que emplee el filtro, a la naturaleza de las señales que procesa, respuesta en frecuencia, método de diseño, etc., se puede clasificar a los filtros.

Con base a la respuesta en frecuencia:

- Filtro paso bajas: Es aquel que permite el paso de frecuencias bajas, incluyendo componente de directa, hasta una frecuencia de corte y rechaza las altas frecuencias por arriba de su frecuencia de corte.
- Filtro paso altas: Es el que permite el paso de frecuencias mayores a una frecuencia de corte determinada, sin que exista un límite superior especificado.
- Filtro paso banda: Son aquellos que permiten el paso de componentes frecuenciales contenidos en un determinado rango de frecuencias, comprendido entre una frecuencia de corte superior y otra inferior.
- Filtro supresor de banda: Es el que impide el paso de componentes frecuenciales contenidos en un determinado rango de frecuencias, comprendido entre una frecuencia de corte superior y otra inferior.
- Filtro variable: Es aquel que puede cambiar su(s) frecuencia(s) de corte.

Filtros activos y pasivos

- Filtro pasivo: Es el que está constituido únicamente por componentes pasivos como capacitores, bobinas y resistores.
- Filtro activo: Es el que está constituido por componentes activos y pasivos, por lo que puede presentar ganancia.

Atendiendo a la naturaleza de las señales procesadas por el filtro, pueden ser filtros analógicos o digitales:

- Filtro analógico: Diseñado para el tratamiento de señales analógicas.
- Filtro digital: Diseñado para el tratamiento de señales digitales.

También es posible clasificar a los filtros como lineales o no lineales. Se dice que un filtro es lineal cuando la respuesta del dispositivo es una combinación lineal de la entrada del filtro y de sus derivadas, en otro caso, se dice que el filtro no es lineal.

2.2. Filtros Adaptables

Los filtros digitales adaptables son aquellos cuyos parámetros varían dinámicamente de acuerdo a un algoritmo que involucra la entrada del filtro, la salida actual y la salida deseada. En general, un filtro digital adaptable consta de dos partes: un filtro digital que procesa la señal de entrada y un algoritmo recursivo encargado de ajustar los coeficientes de dicho filtro mediante el cálculo del error y el uso de herramientas estadísticas.

Esto implica que este tipo de filtros son variantes con el tiempo, a diferencia de la mayoría de los filtros, en los cuales los parámetros de sus funciones de transferencia permanecen constantes.

Ya que el problema del filtrado lineal se puede resolver también con el uso de herramientas estadísticas, como promedios y correlaciones, se plantea la necesidad de diseñar un filtro que considere la señal de ruido aditivo como una entrada de forma tal, que se minimice su efecto de acuerdo a un criterio estadístico. Un criterio muy útil para conseguir este propósito es minimizar el valor del error cuadrático medio del error, el cual se define como la diferencia de la señal deseada menos la respuesta actual del filtro. Para señales estacionarias[1], la solución que cumple con los requisitos anteriores es conocida como filtro Wiener, la cual es considerada la solución óptima.

El filtro Wiener tiene la desventaja de ser inadecuado bajo circunstancias donde el problema involucra señales no estacionarias con ruido intrínseco. Para esta situación se tiene una solución más adecuada, pero más complicada, el filtro Kalman.

El diseño de un filtro Wiener requiere información a priori, de algunos parámetros estadísticos de la entrada, por lo que, el filtro es óptimo sólo cuando la información de la señal de entrada se ajusta por completo a la información a priori con la que se diseñe el filtro. Sin embargo, existe una mejor aproximación para dichas situaciones, es el procedimiento de "aproximación y conexión", este procedimiento consta de dos fases, primero el filtro estima los parámetros estadísticos de las señales relevantes y luego "conecta" los resultados obtenidos en una fórmula no recursiva para calcular los parámetros del filtro. Para operación en tiempo real, éste procedimiento presenta muchas desventajas, ya que requiere hardware costoso y elaborado.

El procedimiento anterior es retomado por la teoría de filtrado y se aprecia en la estructura general de un filtro adaptable, como se muestra en la figura 2.2, que está conformado por la entrada general del sistema x(n) (entrada), la salida general del sistema y(n) (salida) y la salida deseada d(n) (entrada), a partir de las cuales, el filtro realiza internamente dos procesos básicos fácilmente identificables:

1. Un proceso de filtrado, diseñado para producir una salida en respuesta a una secuencia de datos de entrada.

[1] Las señales estacionarias son aquellas cuyos parámetros estadísticos permanecen constantes con respecto al tiempo. Las señales estacionarias se dividen en señales deterministas y aleatorias.

2. Un proceso adaptable, con el propósito de proporcionar un mecanismo para el control adaptable de un conjunto de parámetros usados en el proceso de filtrado.

El proceso adaptable, que actualiza parámetros, suele ser un algoritmo recursivo, el cual parte de un conjunto de condiciones iniciales predeterminadas, que representan algunos aspectos conocidos del entorno, minimizando el error e(n) definido como e(n)=d(n)-y(n).

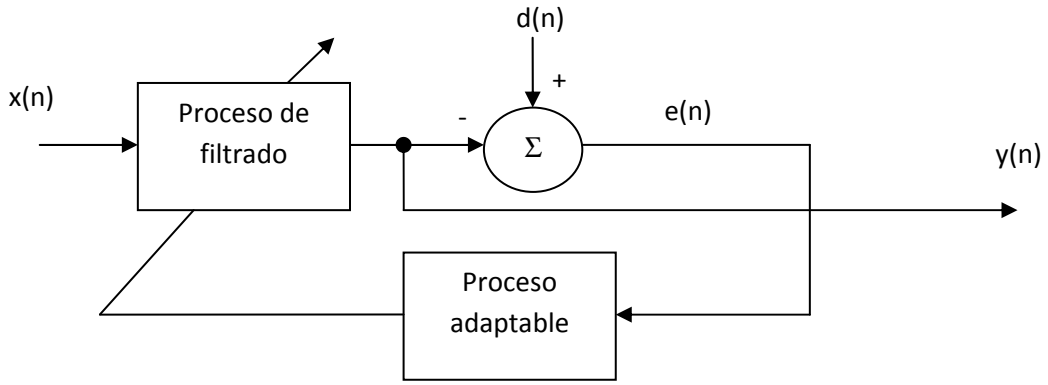


Figura 2.2. Filtro adaptable básico

2.2.1 Características de los filtros adaptables

Como consecuencia de utilizar un algoritmo recursivo, los parámetros del filtro son actualizados de una iteración a otra, pero estos parámetros dependen de la entrada. Los filtros adaptables son comúnmente clasificados como lineales y no lineales.

Se dice que un filtro adaptable es lineal si la estimación de la respuesta es calculada de forma adaptable (a la salida del filtro) como una combinación lineal del conjunto disponible de observaciones aplicadas a la entrada del filtro, de otra forma, se dice que el filtro adaptable no es lineal.

Se ha desarrollado una amplia gama de algoritmos recursivos para la operación de filtros adaptables lineales. Para escoger un algoritmo de entre otros, es necesario prestar atención a uno o más de los siguientes criterios:

- *Convergencia.* Es definida como el número de iteraciones requeridas por el algoritmo, en respuesta a entradas estacionarias, para converger "lo suficientemente cerca" a la solución óptima de Wiener, en términos del error cuadrático medio. Una convergencia más rápida permite al algoritmo adaptarse rápidamente al entorno estacionario de las características estadísticas desconocidas.
- *Desajuste.* Para un algoritmo de interés, este parámetro provee una medida cuantitativa de la cantidad por la cual el valor final del error cuadrático medio, promediado sobre un conjunto de filtros adaptables, se desvía del error cuadrático medio producido por un filtro Wiener.

- *Seguimiento.* Cuando un algoritmo de un filtro adaptable opera en un entorno no estacionario, es necesario que el algoritmo siga las variaciones estadísticas del entorno. El desempeño del seguimiento es influenciado por dos características contradictorias: (1) convergencia, y (2) fluctuación del estado estable debido a ruido del algoritmo.

- *Robustez.* Para un filtro adaptable que sea robusto, pequeñas variaciones (por ejemplo pequeñas alteraciones de energía) sólo puede reflejarse en pequeñas estimaciones de errores. Las alteraciones pueden deberse a una gran variedad de factores, internos o externos al filtro.

- *Requerimientos computacionales.* En este rubro se debe considerar (a) el número de operaciones (por ejemplo multiplicaciones, divisiones, sumas y restas) requeridas para realizar una iteración completa del algoritmo, (b) la cantidad de localidades de memoria requeridas para almacenar los datos y el programa, y (c) la inversión requerida para programar el algoritmo en una computadora.

- *Estructura.* Se refiere a la estructura del flujo de la información en el algoritmo, determinando la manera en la cual se implementa en hardware. Por ejemplo, un algoritmo, cuya estructura implica una gran modularidad, paralelismo, o concurrencia, ajustado para su implementación con el uso de una gran escala de integración (VLSI).

- *Propiedades numéricas.* Cuándo un algoritmo es implementado numéricamente, el error de cuantización se puede propagar (los errores de cuantización se deben a la conversión analógico digital de los datos de entrada y la representación digital de los cálculos internos). Normalmente, ésta es la última fuente de los errores que representa un problema serio de diseño. En particular, hay dos aspectos básicos a ser considerados: estabilidad numérica (es una característica inherente de un algoritmo de filtrado adaptable) y exactitud numérica (es determinado por el número de bits usados en la representación numérica de muestras de los datos y coeficientes del filtro). Un algoritmo de filtrado adaptable se dice robusto numéricamente cuando es insensible a las variaciones de la longitud de palabra usada en la implementación digital.

Estos factores también influyen en el diseño de filtros adaptables no lineales.

2.2.2 Estructuras lineales de filtros adaptables

Como ya se mencionó, la operación de algoritmos adaptables involucra dos procesos básicos: (1) un proceso de filtrado, y (2) un proceso adaptable. Estos dos procesos trabajan interactivamente el uno con el otro. Naturalmente, la elección de una estructura para el proceso de filtrado tiene un efecto contundente en la operación del algoritmo como un todo.

Existen tres tipos de estructuras de filtrado que se distinguen a sí mismas en el contexto de un filtro adaptable con respuesta impulso finita (FIR). Las tres estructuras son las siguientes:

1. Filtro transversal

   Esta estructura de filtrado, descrita por primera vez por Kalman como un dispositivo de tiempo continuo, también es conocido como filtro lineal con retrasos y derivaciones, consta de tres elementos básicos:

   a. Elemento unitario de retardo
   b. Multiplicador
   c. Sumador.

   El número de retardos usados en el filtro determina la duración finita de la respuesta impulso. El número de elementos de retardo M-1, es también el orden del filtro.

En la figura 2.3, los elementos de retardo son identificados cada uno por el operador retardo unitario z^{-1}, en particular, cuando el operador se aplica a la entrada u(n) el resultado es u(n-1). El papel de cada multiplicador en el filtro es multiplicar la derivación de la señal de entrada (a la que esté conectada) por un coeficiente del filtro conocido como peso de la derivación, donde w^*_k es el peso de la derivación k=0,1,…,M-1.

El asterisco denota conjugación compleja. El papel combinado de los sumadores, es sumar las salidas individuales de los multiplicadores y producir una salida final del filtro. Para el filtro transversal descrito en la figura 2.3, la salida del filtro está dada por:

$$y(n) = \sum_{k=0}^{M-1} w_k^* u(n-k)$$

Esta ecuación es llamada una suma finita de convolución ya que involucra la duración finita de la respuesta impulso del filtro con la entrada del filtro para producir una salida.

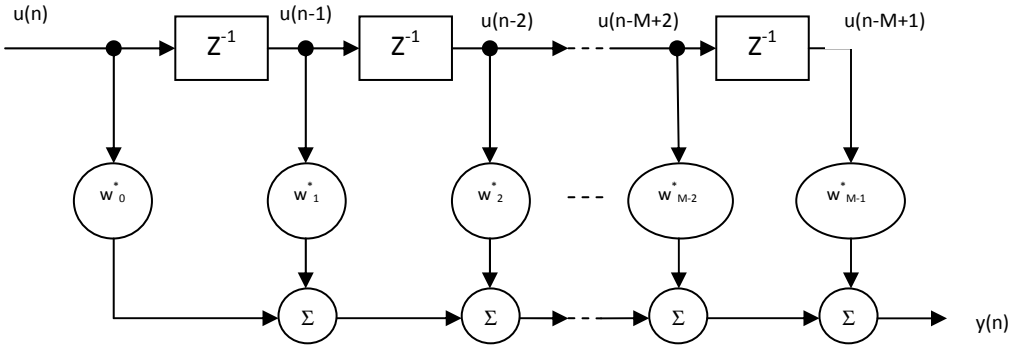


Figura 2.3. Filtro Transversal

2. Predictor lattice (retículo)

Un predictor Lattice es una estructura modular que consiste de un número de etapas individuales, las cuales tienen la apariencia lattice. En la figura 2.4 se observa un predictor lattice consistente de M-1 etapas, el número M-1 también es el orden del predictor, la etapa número m del predictor laticce de la figura está descrita por un par de relaciones entrada salida:

$$f_m(n) = f_{m-1}(n) + k_m^* b_{m-1}(n-1)$$
$$b_m(n) = b_{m-1}(n-1) + k_m f_{m-1}(n)$$

donde m=1,2,…,M-1. La variable $f_m(n)$ es la m-predicción de error hacia adelante, y $b_m(n)$ es la m-predicción de error hacia atrás. El coeficiente km es llamado el m-coeficiente de reflexión. La predicción del error hacia adelante está definida como la diferencia entre la entrada u(n) y su valor predicho por un paso; la última está basada en el conjunto de las m entradas pasadas u(n-1), …,u(n-m). Correspondientemente, la predicción de error hacia atrás está definida como la diferencia entre la entrada u(n-m) y sus "predicciones hacia atrás" basadas en el conjunto de m entradas "futuras" u(n),…,u(n-m+1). Considerando las condiciones en la entrada de la etapa 1 de la figura 2.4 como:

$$f_0(n) = b_0(n) = u(n)$$

donde u(n) es la entrada del predictor lattice en el tiempo n, entonces, comenzando con las condiciones iniciales y dado un conjunto de coeficientes de reflexión $k_1,k_2,\ldots,k_{M-1}$, se puede determinar el par de salidas finales $f_{M-1}(n)$ y $b_{M-1}(n)$ moviéndose a través del predictor lattice, etapa por etapa.

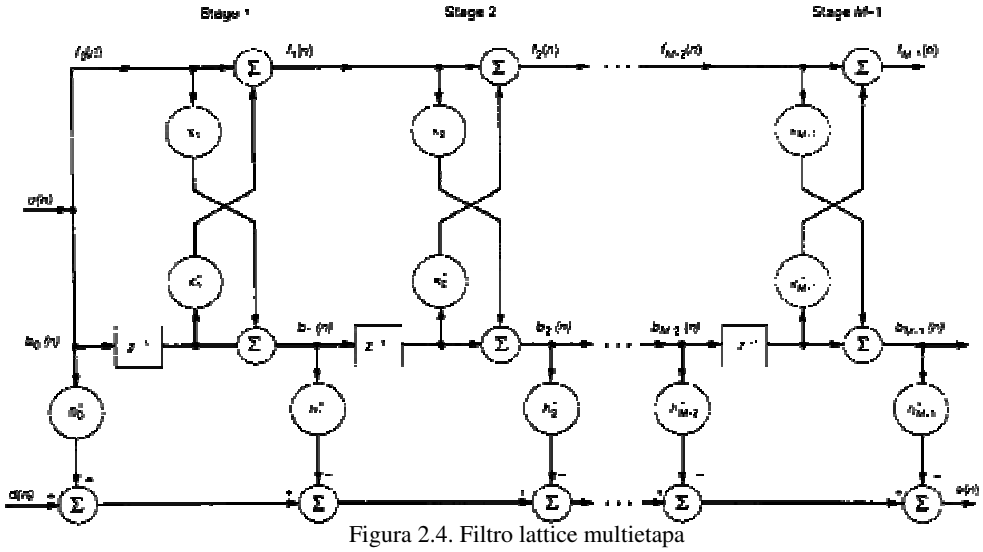


Figura 2.4. Filtro lattice multietapa

3. Arreglo Sistólico

Un arreglo sistólico (propuesto por Kung y Leiserson en 1978) representa una red de procesamiento en paralelo idealmente ajustada para el mapeo de cálculos de algebra lineal, tal como matrices, multiplicaciones, triangulaciones y sustitución hacia atrás. Se pueden distinguir dos tipos de elementos básicos de procesamiento en un arreglo sistólico:

1. Celdas de frontera
2. Celdas internas.

Sus funciones son representadas en la figura 2.5(a y b). En cada caso, el parámetro r representa el valor almacenado dentro de la celda.

La función de la celda de frontera es producir una salida igual a la entrada u dividida por un número r almacenado en la celda.

La función de la celda interna es doble:

(a) multiplicar la entrada z (que viene de la parte superior) por un número r almacenado en la celda, substraer el producto rz de la segunda entrada (que viene de la izquierda), y así producir la diferencia u-rz como una salida del lado derecho de la celda, y

(b) transmitir la primera entrada z hacia abajo sin alteraciones.

Considerando el ejemplo del arreglo triangular de 3 por 3 mostrado en la figura 2.5c, el arreglo sistólico involucra una combinación de celdas de frontera y celdas internas. En este caso, el arreglo triangular calcula un vector de salida relacionado al vector de entrada u de la siguiente forma:

$$Y=R^{-T}u$$

Donde R^{-T} es la matriz inversa de la matriz transpuesta R^T. Los elementos de R^T son los contenidos de las celdas del arreglo triangular. Los ceros se agregan a las entradas del arreglo de la figura 2.5c y son destinados a proveer los retardos necesarios para la continuidad del cálculo descrito en la ecuación anterior.

Una arquitectura de un arreglo sistólico, como la anteriormente descrita, ofrece características deseables de modularidad, interconexión local y proceso interconectado y paralelo sincronizado; la sincronización se logra por medio de un reloj global.

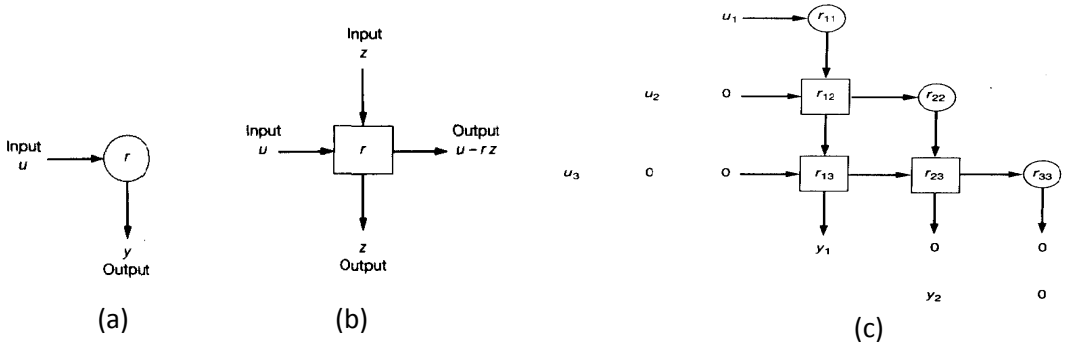


Figura 2.5. (a) celda frontera, (b) celda intermedia, (c) arreglo sistólico triangular

El filtro transversal, el proceso conjunto del estimador basado en el predictor lattice y, finalmente, el arreglo sistólico están caracterizados por la respuesta impulso de duración finita, en otras palabras, son ejemplos de filtros FIR.

Por otra parte, la estructura de filtro IIR (respuesta impulso de duración infinita) se distingue por la inclusión de líneas de realimentación, es más, es ésta la que produce una respuesta impulso infinita, pero la realimentación involucra un nuevo problema: el de estabilidad. En particular, es posible para un filtro IIR ser inestable. En contraste, un filtro FIR que es inherentemente estable, por lo que son comúnmente la base estructural para el diseño de filtros adaptables lineales.

2.2.3. Algoritmos para filtrado adaptable lineal

No existe una única solución para el problema del filtrado adaptable, más bien se cuenta con un "conjunto de herramientas" presentes en una variedad de algoritmos recursivos, donde cada algoritmo ofrece características deseables únicas.

Básicamente hay dos aproximaciones para generar algoritmos recursivos para la operación de filtros adaptables lineales:

- Aproximación del gradiente estocástico (Algorítmo LMS)

Aquí se puede usar el filtro transversal como la base estructural. Para el caso de entradas estacionarias, la función de costo (también conocida como índice de rendimiento) se define como el error cuadrático medio. **Esta función de costo es precisamente una función de segundo orden de los pesos del filtro transversal**. La dependencia del error cuadrático medio de los pesos

desconocidos se representa como una superficie con la forma de un paraboloide multidimensional con un único y definido valor mínimo, que es la superficie de desempeño del error.

Para desarrollar un algoritmo recursivo que actualice los pesos del filtro adaptable transversal, se procede en dos etapas.

Primero se modifica la ecuación de Wiener-Hopf (por ejemplo la ecuación matricial que define la solución óptima de Wiener) a través del uso del método de obtención de mínimos, una técnica bien conocida en la teoría de optimización. Ésta modificación requiere el uso de un vector gradiente, cuyo valor depende de dos parámetros: la matriz de correlación de las entradas derivadas en el filtro transversal y el vector de correlación cruzada entre la respuesta deseada y las mismas entradas derivadas.

A continuación, se usan los valores instantáneos de forma tal que se genere una estimación del vector gradiente, haciéndolo asumir en general un carácter estocástico. El algoritmo resultante es ampliamente conocido como el algoritmo del error cuadrático medio mínimo (LMS), cuya esencia se describe con las siguientes palabras (para el caso de la operación del filtrado transversal con datos de variable real):

$$\begin{pmatrix} Valor\_actualizado \\ del\_vector\_de\_pesos \\ de\_derivaciones \end{pmatrix} = \begin{pmatrix} Valor\_anterior \\ del\_vector\_de\_pesos \\ de\_derivaciones \end{pmatrix} + \begin{pmatrix} parámetro\_de \\ aprendizaje \end{pmatrix} \begin{pmatrix} Vector\_de \\ entrada\_de \\ derivación \end{pmatrix} \begin{pmatrix} señal \\ de \\ error \end{pmatrix}$$

Donde la señal de error está definida como la diferencia entre cierta respuesta deseada y la respuesta actual del filtro transversal.

El algoritmo LMS es simple y proporciona resultados satisfactorios bajo las condiciones descritas. Sus limitaciones son: convergencia relativamente lenta y baja sensibilidad a las variaciones de las entradas. No obstante, el algoritmo LMS es muy popular y ampliamente usado en una gran variedad de aplicaciones.

La aproximación del gradiente estocástico puede ser implementada con la estructura lattice. Y el algoritmo de filtrado adaptable que resulta es llamado algoritmo Lattice de Gradiente Adaptable (GAL). A su forma propia, los algoritmos LMS y GAL son sólo dos miembros de la familia de gradientes estocásticos de filtros adaptables lineales, además se debe decir que el algoritmo LMS es por mucho el miembro más popular de esta familia.

- Estimación de mínimos cuadrados recursivos RLS

La segunda aproximación para el desarrollo de algoritmos de filtrado adaptable lineal está basada en el método de mínimos cuadrados. De acuerdo a éste método se minimiza **la función de costo que se define como la suma del cuadrado de los errores pesados**, donde el error o residuo se define asimismo como la diferencia entre alguna respuesta deseada y la salida actual del filtro.

El método de mínimos cuadrados puede formularse ya sea con bloques de estimación o por estimación recursiva.

En un bloque de estimación, la entrada actual es organizada en forma de bloques de igual longitud (duración) y el filtrado de la entrada se procesa con base bloque a bloque.

En la estimación recursiva, las estimaciones de interés (por ejemplo los pesos de un filtro transversal) son actualizadas con una base muestra a muestra. El estimador recursivo requiere menos almacenamiento que un estimador de bloque, por lo que se usa con más frecuencia en la práctica.

La estimación por mínimos cuadrados recursivos (RLS) puede ser descrita como un caso especial del filtrado de Kalman.

Una característica distintiva del filtro Kalman es la noción de estado, el cual provee una medición de todas las entradas aplicadas al filtro hasta el instante de tiempo específico. Así, en el corazón del algoritmo del filtrado de Kalman se tiene una recursión que puede ser descrita de la siguiente forma:

$$\begin{pmatrix} Valor\_actualizado \\ del\_estado \end{pmatrix} = \begin{pmatrix} Valor\_anterior \\ del\_estado \end{pmatrix} + \begin{pmatrix} ganancia\_de \\ Kalman \end{pmatrix} \begin{pmatrix} vector\_de \\ innovación \end{pmatrix}$$

Donde el vector de innovación representa nueva información puesta dentro del proceso de filtrado al momento del cálculo. Para el estado presente, es suficiente decir que hay una correspondencia uno a uno entre las variables de Kalman y las variables del RLS. Además se puede clasificar la familia de mínimos cuadrados recursivos de los algoritmos de filtrado adaptable lineal, dentro de tres categorías distintas, dependiendo de la aproximación tomada:

- *Algoritmos RLS estándar,* el cual usa filtros. Las limitaciones incluyen las carencias de robustez numérica que ha impulsado el desarrollo de otras dos categorías de algoritmos RLS.
- *Algoritmos raíz-cuadrado RLS,* que están basados en la descomposición QR de la nueva matriz de datos. Los filtros adaptables lineales resultantes quedan referidos a filtros adaptables de raíz-cuadrática.
- *Algoritmos RLS rápidos.* El algoritmo RLS estándar y raíz-cuadrado RLS implican complejidad computacional que incrementa con el cuadrado de M, donde M es el número de pesos adaptables en el algoritmo (por ejemplo los grados de libertad). Tales algoritmos son referenciados frecuentemente como algoritmos O (M^2), donde O(.) denota "orden de". En contraste, el algoritmo LMS es un O(M), donde la complejidad computacional se incrementa linealmente con respecto a M. Cuando M es mayor, la complejidad computacional de los algoritmos O(M^2) puede resultar objetable desde el punto de vista de la implementación con hardware. Con el objetivo de reducir la complejidad computacional del algoritmo RLS equiparable a la del LMS, en el caso de procesamiento temporal, dada la virtud de redundancia, se combinan las características deseables de la estimación recursiva de mínimos cuadrados con una complejidad computacional de O(M).

2.2.4. Filtros Adaptables no lineales

La teoría de filtros lineales óptimos se basa en el criterio del error cuadrático medio. El filtro Wiener que resulta de la minimización de tal criterio, y que representa la meta del filtrado adaptable lineal para un entorno estacionario, sólo puede ser relacionado a estadística de segundo orden de la variable de entrada y no mayores, esto limita la habilidad de un filtro adaptable lineal para extraer información de una entrada que no sea Gaussiana.

Los procesos no Gaussianos son bastante comunes en muchas aplicaciones de procesamiento de señales llevadas a la práctica. El uso de un filtro Wiener o un filtro adaptable lineal para extraer señales de interés en presencia de tales procesos no gaussianos convergerá a soluciones que no sean del todo óptimas. Con el uso de un filtro adaptable no lineal (incluyendo alguna no linealidad en la estructura del filtro adaptable) se deben superar estas limitaciones, porque así se considera un orden estadístico mayor. Como resultado se esperaría beneficiar en dos aspectos: mejorando la eficiencia de aprendizaje y una ampliación de las áreas de aplicación.

Fundamentalmente, hay dos tipos de filtros adaptables no lineales, como se describe a continuación:

Filtros adaptables no lineales basados en Volterra
En este tipo de filtros adaptables no lineales, la alinealidad está localizada en la parte final del filtro. Se basa en el uso de series de Volterra que proveen un método atractivo para describir la relación entrada salida de un dispositivo no lineal con memoria. Esta forma especial de series (estudiadas por primera vez por Vito Volterra alrededor de 1880) como una generalización de las series de Taylor de una función. Sin embargo fué Norbert Wiener (1958) el primero en usar las series de Volterra para modelar la relación entrada salida de un sistema no lineal.

Filtros adaptables difusos
El uso de la lógica difusa por medio de sistemas de inferencia difusa, encuentra un gran nivel de aplicación en sistemas de control con realimentación, proporcionando ventajas muy importantes, como es la elaboración de sistemas de control no lineales sin la necesidad de una complejidad muy elaborada de cálculos. Por la versatilidad de la lógica difusa, podemos analizar diversos procesos en lo cuáles podemos aplicar los sistemas de inferencia difusos para su solución, con cierta facilidad, por lo que en el presente trabajo de tesis se aplica en la implementación de filtros adaptables.

En la práctica, la información viene usualmente de dos fuentes: sensores que proveen datos numéricos asociados a un problema y descripciones lingüísticas (frecuentemente en forma de reglas IF-THEN difusas) provenientes de expertos humanos sobre el problema.

Los filtros existentes sólo pueden procesar datos numéricos, mientras que los sistemas expertos sólo pueden hacer uso de información lingüística. De ahí, sus aplicaciones exitosas están limitadas a problemas donde, ya sean datos numéricos o lingüísticos, no juegan un papel crítico.

Un filtro adaptable difuso está construido por un conjunto de reglas difusas IF-THEN modificables. Estas reglas difusas son definidas ya sea por expertos humanos o por conjuntos de pares entrada-salida a través de procedimientos de adaptación. Los algoritmos adaptables actualizan los parámetros de las funciones de pertenencia, las cuáles caracterizan los conceptos difusos en las reglas IF_THEN minimizando algunas funciones de criterio

Redes Neuronales
Una red neuronal artificial, o red neuronal como se le conoce comúnmente, consiste en la interconexión de un gran número de unidades procesadoras no lineales llamadas neuronas; esto es, la no linealidad está distribuida a través de la red.

Es de particular interés la clase de redes neuronales que aprenden del entorno de forma supervisada. En otras palabras, al igual que en la forma tradicional de filtros adaptables lineales, se cuenta con una respuesta deseada que provee una señal objetivo, a la cual, la red neuronal intenta aproximarse durante su proceso de aprendizaje. La aproximación es lograda ajustando el conjunto de parámetros,

llamados pesos sinápticos, de forma sistemática. En efecto, los pesos sinápticos proveen un mecanismo de almacenamiento para la información contenida por los datos de entrada.

Dentro del contexto de aplicaciones del procesamiento adaptable de señales, las redes neuronales ofrecen las siguientes ventajas:

- No linealidad, lo cual hace posible medir el comportamiento no lineal del fenómeno físico responsable de la generación del dato de entrada.
- La habilidad de aproximar cualquier mapeo prescrito de entrada-salida
- Hipótesis estadística débil sobre el entorno en el que la red está embebida
- Capacidad de aprendizaje, la cual es completada teniendo una sesión de entrenamiento con ejemplos entrada salida que son representativos del entorno.
- Generalización, la cual hace referencia a la habilidad de la red neuronal para proveer un desempeño satisfactorio en respuesta a la entrada de prueba, nunca vista antes por la red neuronal.
- Tolerancia a errores, lo que significa que la red continúa funcionando para proveer un desempeño satisfactorio a pesar de la falla de algunas neuronas en la red.
- Implementabilidad VLSI, que explota la construcción masiva en paralelo dentro del diseño de una red neuronal.

Esta lista de atributos es importante para el amplio espectro de interés en el uso de redes neuronales para resolver tareas de procesamiento de señales que son muy difíciles para los filtros adaptables convencionales (lineales).

2.2.5. Aplicaciones de los filtros adaptables

En numerosas aplicaciones de procesamiento digital y de comunicaciones, se enfrenta la necesidad de suprimir el ruido y la distorsión de las señales. Este fenómeno se relaciona con los procesos físicos de variación con el tiempo, el cual es desconocido en muchas ocasiones. Una de estas situaciones es durante la transmisión de una señal (mensaje) de un punto a otro. El medio de transmisión (cables, fibras, microondas, etc.), conocido como canal, agrega ruido y distorsión en función de las variaciones de sus propiedades. Estas variaciones pueden suceder lenta o rápidamente. Debido a que la mayoría de las variaciones en el tiempo son desconocidas, existe el uso de filtros adaptables que disminuye y a veces elimina completamente la distorsión de las señales.

La capacidad de que un filtro adaptable opere satisfactoriamente en un entorno desconocido y siga las variaciones en el tiempo para una entrada hace al filtro adaptable un dispositivo poderoso para el procesamiento digital y aplicaciones de control. Asimismo, los filtros adaptables han sido aplicados exitosamente en una amplia gama de campos como comunicaciones, radar, sonar, sismología e ingeniería biomédica. Si bien todas estas aplicaciones son en realidad distintas en cuanto a naturaleza, tienen en común una característica: un vector de entrada y una respuesta deseada son empleadas para calcular un error estimado, el cual es empleado para controlar los valores de un conjunto de coeficientes ajustables, parámetros de rotación o pesos sinápticos, dependiendo de la estructura de filtro empleada. Además, la diferencia esencial entre las distintas aplicaciones de los filtros adaptables radica en la forma en que la respuesta deseada es extraída. Dentro de este contexto se pueden distinguir cuatro configuraciones básicas de filtros adaptables para diferentes campos de aplicación:

1. Identificación. La noción de un modelo matemático es fundamental para las ciencias y la ingeniería. En este tipo de aplicaciones que tratan con identificadores, un filtro adaptable es usado para proveer un modelo lineal que represente el mejor ajuste (en algún sentido) para una planta desconocida. La salida de la planta provee la respuesta deseada para el filtro adaptable como se aprecia en la figura 2.6. Si la planta es de naturaleza dinámica, el modelo será variante en el tiempo.

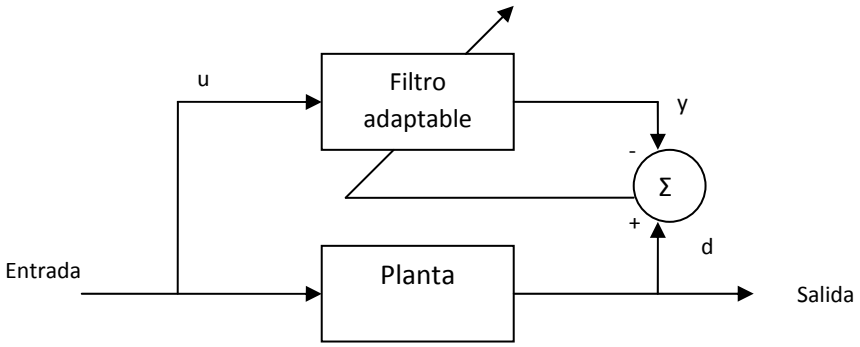


Figura 2.6. Identificador

2. Modelo inverso. En este segundo tipo de aplicación, la función de un filtro adaptable es proveer un modelo inverso que represente el mejor ajuste para una planta desconocida con ruido. Idealmente, en el caso de un sistema lineal, el modelo inverso tiene una función de transferencia recíproca a la función de transferencia de la planta, de modo que la combinación de los dos sistemas constituye un medio de transmisión ideal. Una versión atrasada de la entrada de la planta constituye la respuesta deseada para el filtro adaptable. En algunas aplicaciones, la entrada de la planta es usada sin retraso como respuesta deseada. Véase figura 2.7.

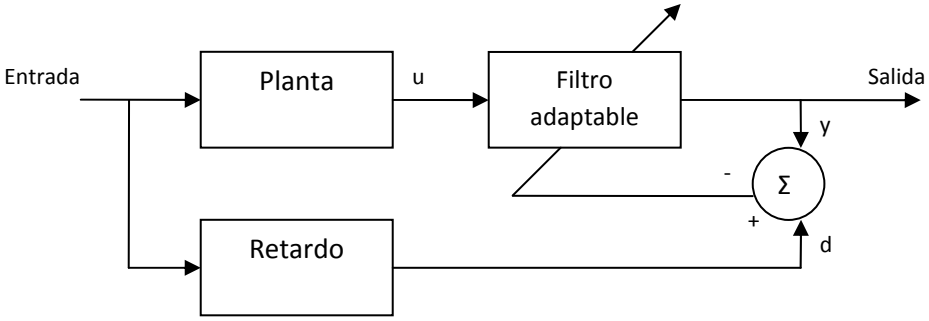


Figura 2.7. Modelo inverso

3. Predicción. Aquí la función del filtro adaptable es proveer la mejor predicción del valor actual de una señal aleatoria. El valor presente de la señal es usado como respuesta deseada por el filtro adaptable. Valores pasados de la fuente de la señal son aplicados como entrada para el filtro adaptable. Dependiendo de la aplicación de interés, el error a la salida del filtro adaptable o estimación (predicción) puede servir como la salida del sistema. En el primer caso, el sistema opera como un predictor, y en el último caso, el sistema opera como un filtro predictor de error. Véase figura 2.8.

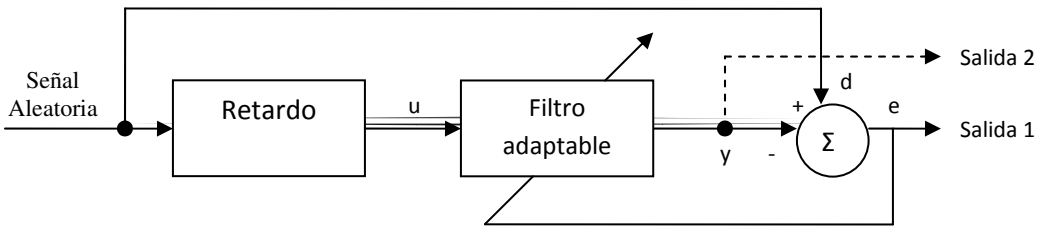


Figura 2.8. Predicción

4. Cancelación de interferencia. En este último tipo de aplicaciones, el filtro adaptable es usado para cancelar la interferencia desconocida contenida en una señal primaria, optimizando en cierto modo la cancelación de la interferencia. La señal primaria sirve como la respuesta deseada del filtro adaptable. Una señal de referencia (auxiliar) es empleada como entrada del filtro adaptable. La señal de referencia es derivada de un sensor o conjunto de sensores localizados con relación al sensor que registra la señal primaria, de modo tal que los componentes de la señal de referencia sean débiles o prácticamente indetectables con respecto a la información de la señal primaria. Véase figura 2.9.

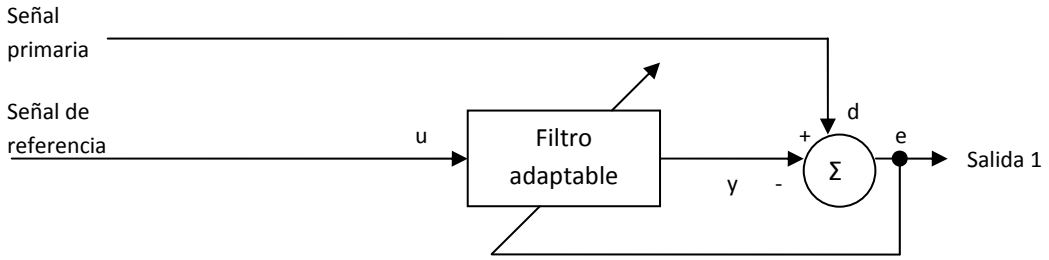


Figura 2.9. Cancelación de interferencia

Clase de filtro adaptable	Aplicación
Identificación	Identificación del sistema Modelado de capas de la tierra
Modelado inverso	Deconvolución predictiva Ecualización ciega
Predicción	Codificación lineal predictiva Modulación adaptable diferencial en código de pulso Análisis autorregresivo espectral Detección de señales
Cancelador de interferencia	Cancelador adaptable de ruido Cancelación de eco Contorneado adaptable

Tabla 2.1. Aplicaciones de filtros adaptables

2.3. Cancelador adaptable de ruido

Como el nombre lo dice, el cancelador adaptable de ruido substrae ruido de una señal recibida, por medio de una operación controlada de una forma adaptable, con el propósito de mejorar la razón señal a ruido. Cuando el filtrado con substracción es controlado por un proceso adaptable, es posible alcanzar un buen desempeño del sistema comparando con el filtrado directo de la señal recibida.

Básicamente, un cancelador adaptable de ruido es un sistema adaptable realimentado en lazo cerrado, de dos entradas, como se ilustra en la figura 2.10. Las dos figuras del sistema son originadas por un par de sensores: un sensor primario y un sensor de referencia (auxiliar) .

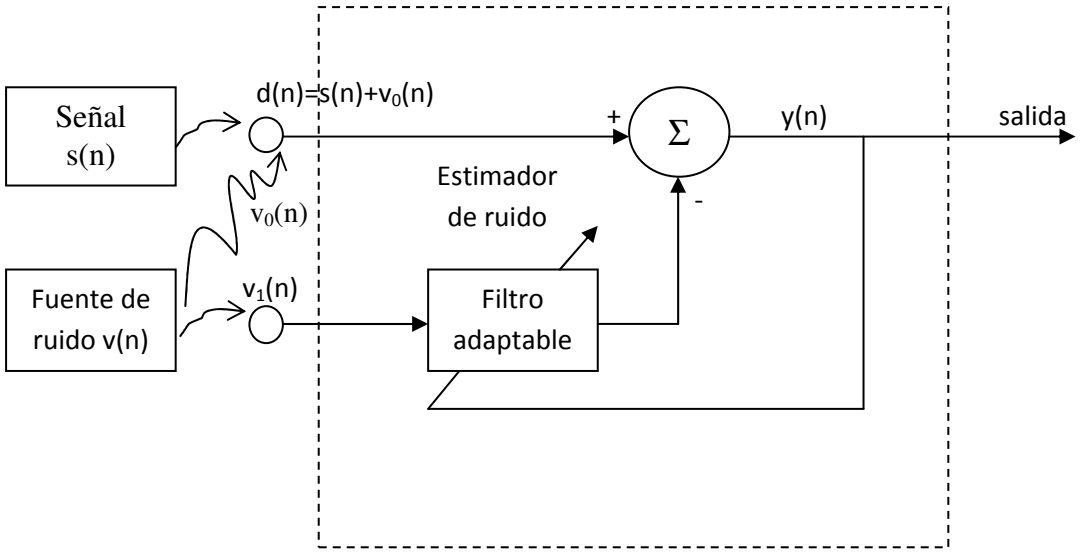


Figura 2.10. Cancelador adaptable de ruido

1. El sensor primario recibe una señal de información s(n) que llega corrupta por ruido aditivo $v_0(n)$ como se muestra:

   $$d(n) = s(n) + v_0(n) \quad (2.1)$$

   La señal s(n) y el ruido $v_0(n)$ no están correlacionados entre si,

   $$E[s(n)v_0(n)] = 0, \; para\_toda\_k$$

   Donde s(n) y $v_0(n)$ son consideradas de variable real.

2. El sensor de referencia recibe un ruido $v_1(n)$, que no está correlacionado con la señal s(n), pero sí está correlacionado con el ruido $v_0(n)$, en la salida del sensor primario de una forma desconocida, esto es

   $$E[s(n)v_1(n)] = 0, \; para\_toda\_k$$

   Y

   $$E[v_0(n)v_1(n)] = p(k)$$

Donde, como antes, las señales son de variable real y p(k) es una correlación cruzada en k.

La señal de referencia $v_1(n)$ es procesada por un filtro adaptable para producir una señal de salida:

$$y(n) = \sum_{k=0}^{M-1} \hat{w}_k(n) v_1(n-k) \quad (2.2)$$

Donde $\hat{w}_k(n)$ son los pesos ajustables (reales) del filtro adaptable. La salida del filtro y(n) es substraída de la señal primaria d(n), sirviendo como respuesta "deseada" para el filtro adaptable. La señal de error está definida por:

$$e(n)=d(n)-y(n) \tag{2.3}$$

y sustituyendo (2.1) en (2.3):

$$e(n)=s(n)+ v_0(n)-y(n) \tag{2.4}$$

La señal de error es, en cambio, usada para ajustar los pesos del filtro adaptable, y cerrando el lazo de control alrededor de las operaciones de filtrado y substracción. Nótese que la señal capturada s(n) es en sí parte de la señal de error e(n), como se indica en la ecuación (2.4).

La señal de error e(n) constituye la salida final del sistema. De la ecuación (2.4) se ve que la componente de ruido en el sistema de salida es v0(n)-y(n). Ahora, el filtro adaptable intenta minimizar el valor del error cuadrático medio e(n) (potencia promedio). La información que se tiene de la señal s(n) es esencialmente inalterada por el cancelador adaptable de ruido.

Minimizar el valor cuadrático medio de la señal de error e(n) es equivalente a minimizar el valor cuadrático medio del ruido de salida v0(n)-y(n). Con la señal s(n) permaneciendo prácticamente constante, la minimización del valor cuadrático medio del ruido es en sí misma la maximización de la razón señal a ruido de la salida del sistema.

Las operaciones de procesamiento de señales descritas tienen dos grandes limitantes:

1. La operación del filtrado adaptable es perfecta en caso que

   $y(n)= v_0(n)$

   En este caso, la salida del sistema queda libre de ruido y la cancelación de ruido es perfecta. Asimismo, la razón señal a ruido a la salida es infinitamente grande.

2. La señal de referencia $v_1(n)$ no tiene correlación alguna con ninguna de las dos señales, la señal de interés ni con las componentes de ruido de la señal primaria d(n), esto es

   $$E[d(n)v_1(n-k)]=0, para\_toda\_k$$

   En este caso, el filtro adaptable "se apaga", resultando en un valor de cero en la salida y(n), ya que el cancelador adaptable de ruido no tiene efecto en la señal primaria d(n) y la razón señal a ruido de la salida no se altera.

El uso efectivo del cancelamiento adaptable de ruido requiere, por las razones anteriores, que se coloque el sensor de referencia en el campo de ruido del sensor primario con dos objetivos específicos en mente:

Primero. La componente de la señal de información capturada de la salida del sensor primario es indetectable en la salida del sensor de referencia.

Segundo. La salida del sensor de referencia esté altamente correlacionada con la componente de ruido de la salida del sensor primario. Además, la adaptación de los coeficientes ajustables del filtro debe ser óptima.

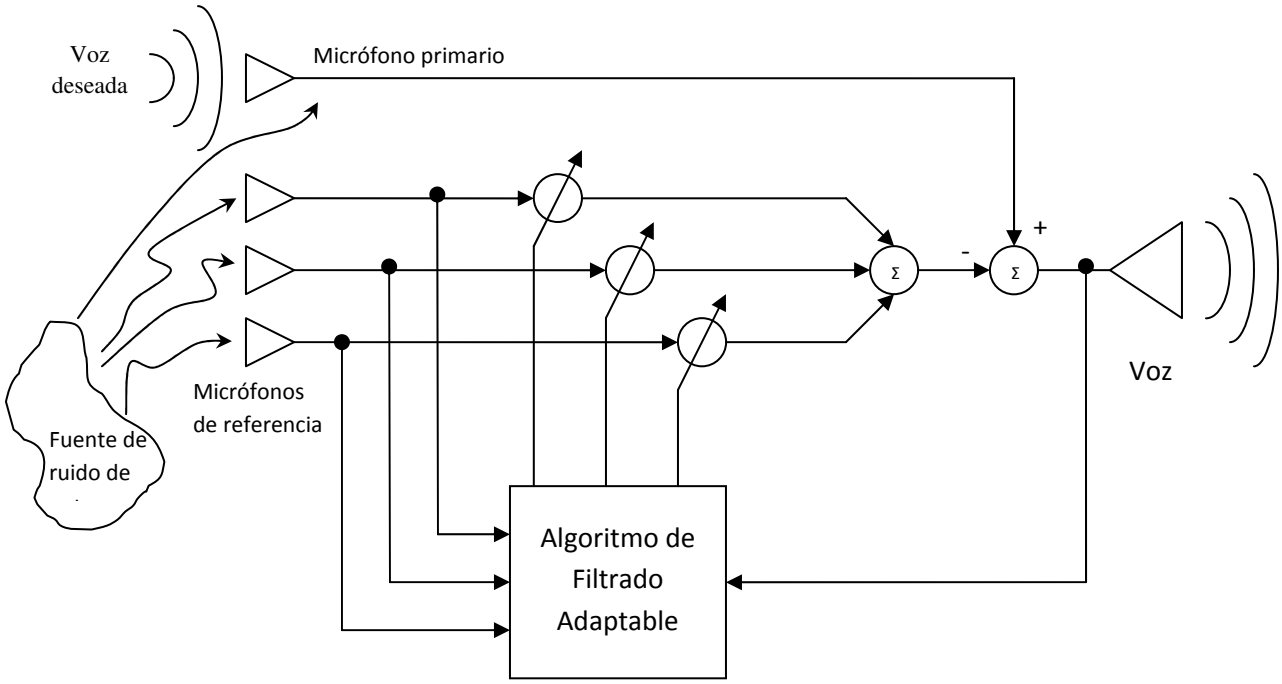


Figura 2.11. Diagrama de un cancelador adaptable de ruido para voz

Considerando la situación planteada en la figura 2.11, se requiere escuchar una voz deseada proveniente de un orador en presencia de un entorno ruidoso, lo que se puede solucionar empleando un filtro adaptable para la cancelación de ruido. Como se indica en la figura 2.11, una suma pesada de micrófonos auxiliares y un algoritmo de filtro adaptable (usado para ajustar los pesos de forma tal que minimice la potencia promedio de salida). El fenómeno se refiere a la habilidad de una persona normal para concentrarse en una conversación que tiene lugar en una locación lejana o en un lugar muy concurrido. Esta habilidad disminuye debido a la sensibilidad extrema a la presencia del ruido de fondo. La sensibilidad es atribuida a dos factores:

(a) la falta de una dirección, y

(b) la capacidad limitada del canal del oído causado por la reducción tanto en rango dinámico como en respuesta en frecuencia.

Chazan (1988) [14] describe una técnica de cancelación adaptable de ruido con el objetivo de superar dicho problema. La técnica involucra el uso de un arreglo de micrófonos para aprovechar la diferencia en características espaciales entre la señal deseada y el ruido del cuarto concurrido. La solución aproximada está basada en el hecho de que la salida de cada micrófono puede ser vista como la suma de las señales producidas por los hablantes individuales unidos en una conversación en el cuarto. Cada contribución de la señal en la salida de un micrófono en especial es esencialmente el resultado del diálogo de cada hablante habiendo pasado a través del filtro del cuarto. En otras palabras, cada hablante (incluyendo el hablante de interés) produce una señal a la salida del micrófono que es la suma de la transmisión directa de su diálogo y sus reflexiones en los muros del cuarto. El requisito es reconstruir la señal del hablante deseado, incluyendo las reverberaciones del cuarto, mientras se cancela la fuente de ruido. En general, la transformación sufrida por la señal del diálogo del hablante deseado no es conocida. Además, Las características del ruido de fondo son variables. Así se tiene un problema de procesamiento de señales para el cual el cancelamiento adaptable de ruido ofrece una solución factible.

Capítulo 3
Fundamentos de Lógica Difusa

En este capítulo se encuentran los conceptos fundamentales de la teoría de la lógica difusa junto con algunos ejemplos que permiten un mejor entendimiento de los mismos. Se parte desde la incorporación de la información lingüística a las aplicaciones de ingeniería, los conjuntos difusos, su representación, operaciones y leyes, las reglas de inferencia difusa, la máquina de inferencia difusa, el sistema difuso de Takagi y Sugeno, hasta llegar al teorema de aproximación universal.

3.1. Información Lingüística

Para analizar sistemas en ingeniería, se cuenta con dos fuentes de información importantes: la información física proveniente directamente de las variables del sistema y la información lingüística, que es proporcionada por humanos expertos en forma de instrucciones y descripciones lingüísticas sobre el sistema.

Mientras que la información de los sensores es numérica (representada por medio de números), la información de los expertos humanos es información lingüística (representada por palabras).

Comúnmente, las soluciones ingenieriles sólo hacen uso de información numérica y difícilmente incorporan información lingüística, sin embargo, mucho conocimiento humano es representado en términos lingüísticos, por lo que es muy importante incorporar esta información en sistemas ingenieriles de una forma eficiente y sistemática.

La información lingüística es representada comúnmente en términos difusos debido a que nuestra comunicación y conocimiento sobre muchos problemas es esencialmente difuso. Asimismo, hay muchos sistemas demasiado complejos para ser descritos en términos bien definidos. Debido a esto, se pueden usar variables lingüísticas junto con información numérica para describir y manipular sistemas en ingeniería.

Un aspecto importante es, que si bien la información lingüística no es precisa, provee información importante acerca del sistema y, a veces, puede ser la única fuente de información, sin embargo, se debe de usar esta información difusa de una forma científica.

Existen dos interpretaciones para las variables lingüísticas:

Formal. Una variable lingüística está caracterizada por una quíntupla (x, T(x), U, G, M) en donde x es el nombre de la variable; T(x) es el conjunto de término de x, esto es, el conjunto de nombres de valores lingüísticos de x, con cada valor siendo un conjunto difuso definido en el universo del discurso U; G es una regla sintáctica para la generación de nombres de valores de x; y M es una regla semántica para la asociación de cada valor con su significado.

Por ejemplo, si se dice: "El movimiento es muy rápido", entonces la variable movimiento debería ser entendida como una variable lingüística; pero eso no quiere decir que la variable movimiento no pueda tomar valores reales.

Informal. Si una variable puede tomar palabras en los lenguajes naturales (como pequeño, rápido, etc.) como sus valores, esta variable es definida como una variable lingüística.

Por ejemplo, la variable rapidez puede tomar los valores "lenta", "moderada" y "veloz", como se muestra en la figura 3.1. También puede tomar cualquier valor numérico real en el intervalo [0,Rapidezmax] como sus valores. La variable lingüística es un concepto importante que proporciona un camino formal para cuantificar descripciones lingüísticas acerca de las variables.

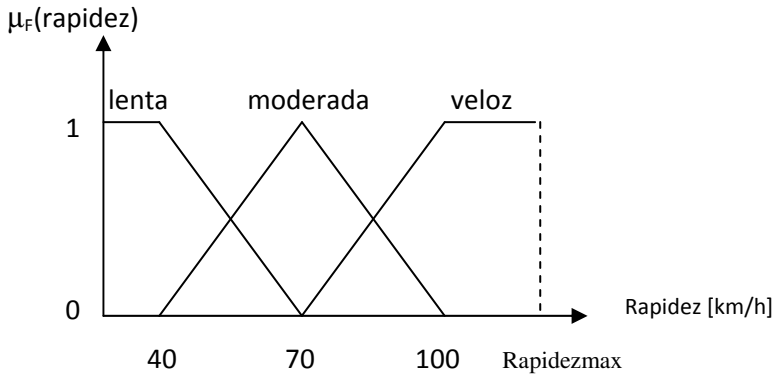


Figura 3.1. Valores para la variable lingüística Rapidez

3.2. Conceptos básicos de Lógica Difusa

El concepto de conjuntos difusos fue introducido por primera vez por L. A. Zadeh en 1965 para procesar información y datos afectados de incertidumbre no probabilística. Estos conjuntos son la base de la lógica difusa y fueron diseñados para representar matemáticamente la incertidumbre y la ambigüedad y así contar con herramientas formalizadas para trabajar con la imprecisión intrínseca de muchos problemas.

En palabras de Zadeh (1992), las características más notables de la Lógica difusa son:

- En Lógica Difusa todo es cuestión de grado

- El Razonamiento Exacto es un caso limite del Razonamiento Aproximado.

- En Lógica Difusa, el conocimiento se interpreta como una colección de restricciones difusas sobre un conjunto de variables.

• En Lógica Difusa, la inferencia puede verse como la propagación de un conjunto de restricciones difusas.

• Un Sistema Difuso es el resultado de la difusión de un sistema convencional.

• Los Sistemas Difusos operan con conjuntos difusos en lugar de con números.

• En esencia la representación de la información en Sistemas Difusos imita el mecanismo de Razonamiento Aproximado que realiza la mente humana.

El término difuso (*fuzzy*) se emplea para expresar la vaguedad, ambigüedad e imprecisión de algún concepto.

3.2.1. El universo del discurso es una colección de objetos de los que se hablará en un discurso específico. Por ejemplo, si se ha de tratar de contribuyentes entonces el universo consistirá de las personas físicas o morales que pagan o han de pagar impuestos.

3.2.2. Conjunto difuso y función de pertenencia. Sea U una colección de objetos, por ejemplo, $U=R^n$, U será llamado el universo del discurso. Un conjunto difuso F en U está caracterizado por una función de pertenencia μ_F :U->[0,1], con $\mu_F(u)$ representando el grado de pertenencia de $u \in U$ en el conjunto difuso F. Un conjunto difuso puede ser visto como una generalización del concepto de los conjuntos ordinarios cuya función de pertenencia sólo toma los valores {0,1}.

La figura 3.2 representa un conjunto difuso F, donde la parte más oscura indica una pertenencia de uno al conjunto difuso, y conforme se aleja de este punto, el grado de pertenencia al conjunto decrece, hasta llegar al grado de cero.

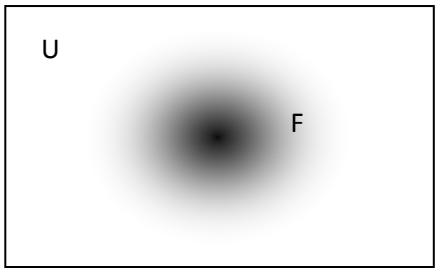
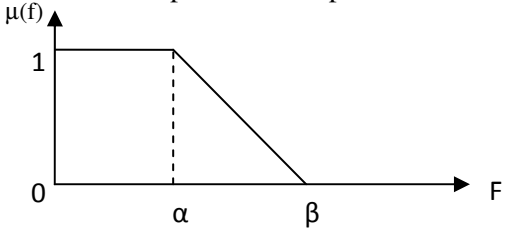


Figura 3.2. Representación gráfica de un conjunto difuso F en un universo U

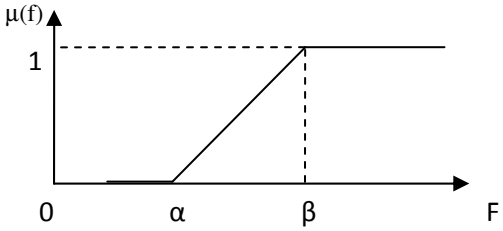
Las funciones de pertenencia más usadas para aplicaciones de ingeniería son las siguientes:

Función de pertenencia tipo "Z"



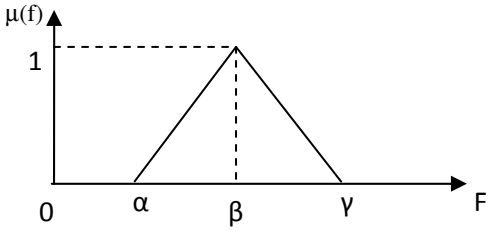
$$\mu(f)=\begin{cases} 1, & f<\alpha \\ \frac{f-\beta}{\alpha-\beta}, & \alpha \le f \le \beta \\ 0, & f>\beta \end{cases}$$

Función de pertenencia tipo "S"



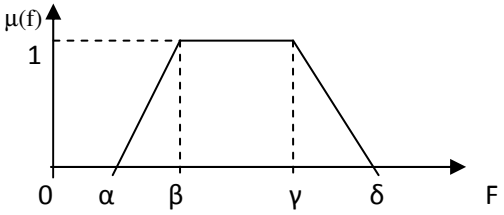
$$\mu(f)=\begin{cases}0, f<\alpha\\ \frac{f-\alpha}{\beta-\alpha}, \alpha\le f\le\beta\\ 1, f>\beta\end{cases}$$

Función de pertenencia triangular o tipo "Λ" (lambda)



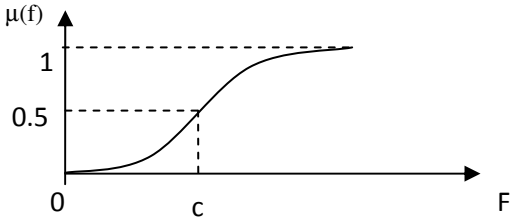
$$\mu(f)=\begin{cases}0, f<\alpha\\ \frac{f-\alpha}{\beta-\alpha}, \alpha\le f\le\beta\\ \frac{f-\gamma}{\beta-\gamma}, \beta<f\le\gamma\\ 0, f>\gamma\end{cases}$$

Función de pertenencia trapezoidal o tipo "Π" (pi)



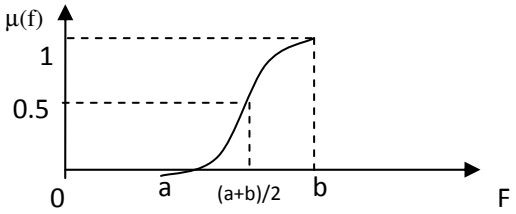
$$\mu(f)=\begin{cases}0, f<\alpha\\ \frac{f-\alpha}{\beta-\alpha}, \alpha\le f\le\beta\\ 1, \beta<f<\gamma\\ \frac{f-\delta}{\beta-\delta}, \gamma\le f\le\delta\\ 0, f>\delta\end{cases}$$

Función de pertenencia tipo sigmoide



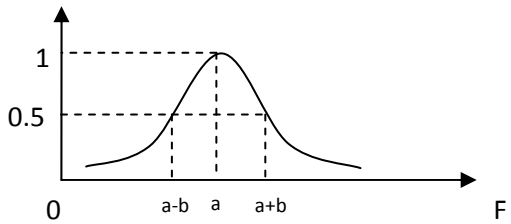
$$\mu(f)=\frac{1}{1+e^{-(f-c)}}$$

Función de pertenencia tipo "S" suave



$$\mu(f)=\begin{cases}0, f<a\\ 2\left(\frac{f-a}{b-a}\right)^2, a\le f\le\frac{a+b}{2}\\ 1-2\left(\frac{f-b}{b-a}\right)^2, \frac{a+b}{2}<f\le b\\ 1, f>b\end{cases}$$

Función de pertenencia tipo campana



$$\mu(f)=\frac{1}{1+\left(\frac{f-a}{b}\right)^2}$$

3.2.3. Soporte, núcleo y singleton. El soporte de un conjunto difuso F es el conjunto definido de todos los puntos $u \in U$ para el cual $\mu_F(u)>0$.

El núcleo de un conjunto difuso F son los puntos $u \in U$ para los que $\mu_F(u)$ alcanza su valor máximo. Esto se ilustra en la figura 3.3.

Si el soporte de un conjunto difuso F es un único punto de U al cual $\mu_{Fs}(u)=1$, F es llamado un singleton, el cual se ilustra en la figura 3.4.

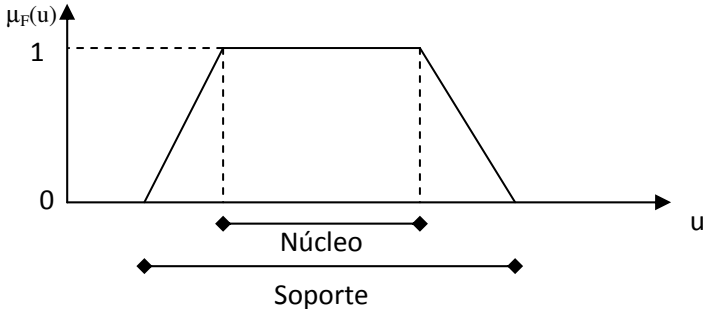


Figura 3.3. Núcleo y Soporte de una función de pertenencia de un conjunto difuso F

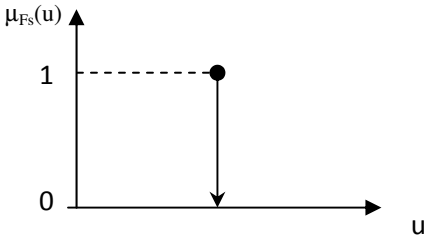


Figura 3.4. Función de pertenencia de un singleton

3.2.4. Intersección, unión y complemento. Sean A y B dos conjuntos difusos en U. La intersección, $A \cap B$, de A y B es un conjunto difuso en U con función de pertenencia definida para toda $u \in U$ por $\mu_{A \cap B}(u)=\min\{\mu_A(u) , \mu_B(u) \}$; dónde min denota mínimo, un ejemplo de intersección se proporciona en la figura 3.5.

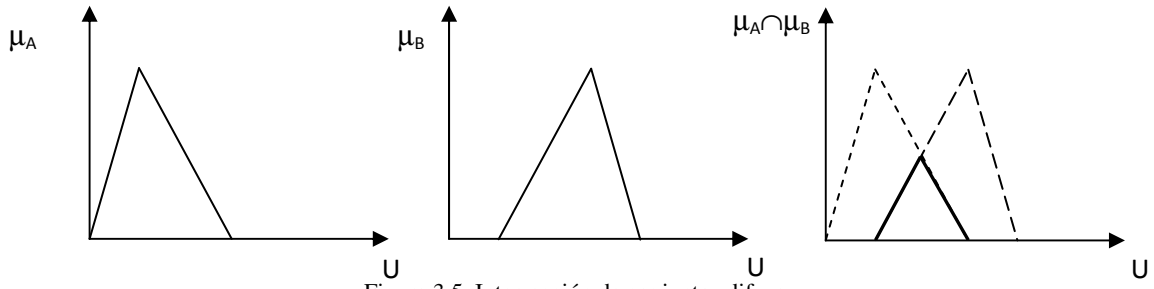


Figura 3.5. Intersección de conjuntos difusos

La unión $A \cup B$, de A y B es un conjunto difuso en U con la función de pertenencia definida para toda $u \in U$ como $\mu_{A \cup B}(u)=\max\{\mu_A(u) , \mu_B(u) \}$; donde max denota máximo, como se ilustra en el ejemplo de la figura 3.6.

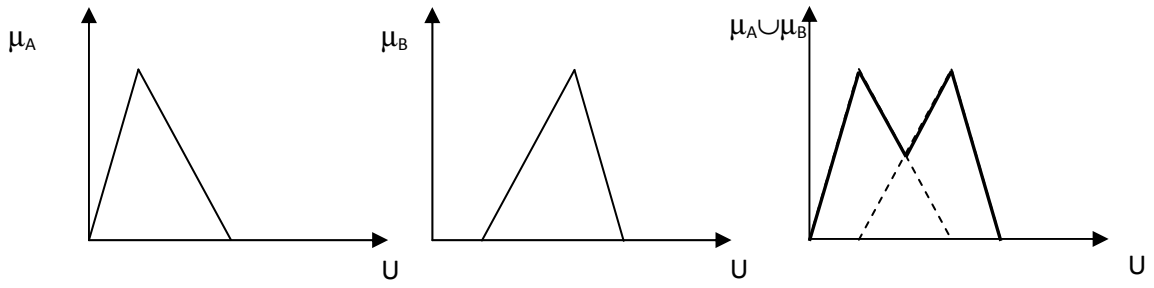


Figura 3.6. Unión de conjuntos difusos

Usualmente, los operadores de intersección y unión son denotados por $\wedge$ y $\vee$, respectivamente. El complemento $\tilde{A}$ de A es un conjunto difuso en U con la función de pertenencia definida para toda $u \in U$ como $\mu_{\tilde{A}}(u)=1-\mu_A(u)$, como se observa en el ejemplo de la figura 3.7.

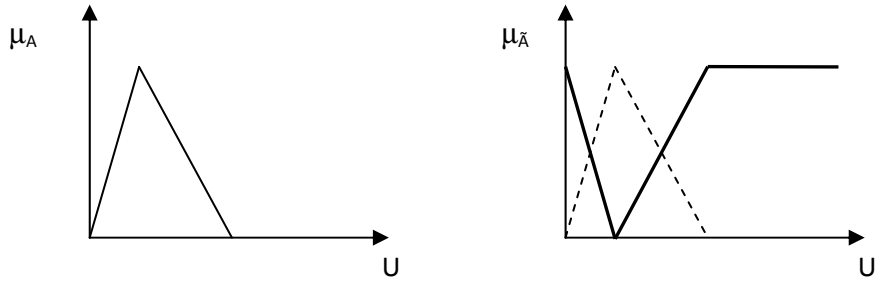


Figura 3.7. Complemento de conjuntos difusos

Las operaciones con los conjuntos difusos cumplen con las siguientes leyes:

- Conmutatividad: $A \cup B=B \cup A$; $A \cap B=B \cap A$
- Asociatividad: $A \cup (B \cup C)=(A \cup B) \cup C$; $A \cap (B \cap C)=(A \cap B) \cap C$
- Distributividad: $A \cup (B \cap C)=(A \cup B) \cap (A \cup C)$; $A \cap (B \cup C)=(A \cap B) \cup (A \cap C)$
- Idempotencia: $A \cup A=A$; $A \cap A=A$
- Identidad: $A \cup \varnothing=A$; $A \cup U=U$; $A \cap U=A$; $A \cap \varnothing=\varnothing$
- Transitividad: Si $A \subset B \subset C \rightarrow A \subset C$
- Involución: $\overline{\overline{A}}=A$
- Leyes de De Morgan: $\overline{A \cup B}=\overline{A} \cap \overline{B}$; $\overline{A \cap B}=\overline{A} \cup \overline{B}$

Donde A, B y C son conjuntos difusos definidos en el universo U.

3.2.5. Relación difusa. Sean U y V dos universos del discurso. Una relación difusa R es un conjunto difuso en el espacio producto UxV; esto es, R tiene la función de pertenencia $\mu_R(u)=(u,v)$, donde $u \in U$ y $v \in V$.

Una forma de representar una relación difusa se presenta a continuación (figura 3.8):

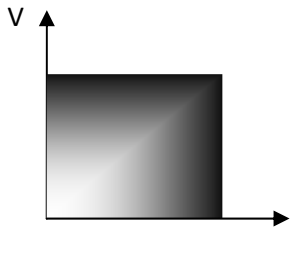


Figura 3.8. Representación gráfica de una relación difusa UxV

3.2.6. Implicaciones difusas.

Sean A y B conjuntos difusos definidos en U y V, respectivamente. Una implicación difusa, denotada por A->B, es un tipo especial de relación difusa en UxV.

Una implicación difusa A->B puede ser comprendida como una regla difusa *IF THEN* (SI ENTONCES): SI x es A, ENTONCES y es B, donde x∈ U y y∈ V y son variables lingüísticas.

Cada regla se puede entender a su vez como un conjunto difuso definido con su propia función de pertenencia que mide el grado de verdad de la relación de implicación entre A y B. Ejemplos posibles de las funciones de pertenencia asociadas a la implicación difusa son:

$$\mu_{A\to B}(x,y) = 1 - \mu_{A\cap B}(x,v) = 1 - \min[\mu_A(x), 1-\mu_B(y)]$$
$$\mu_{A\to B}(x,y) = \max[1-\mu_A(x), \mu_B(y)]$$
$$\mu_{A\to B}(x,y) = 1 - \mu_A(x)(1-\mu_B(y))$$

Las funciones de pertenencia de la implicación difusa corresponden a interpretaciones de la regla difusa SI-ENTONCES basadas en criterios intuitivos o generalizaciones de la lógica clásica.

Una expresión que permite calcular fácilmente la implicación difusa con varios antecedentes y un consecuente es la siguiente. Sean A y B dos conjuntos difusos de entrada definidos en U y C un conjunto difuso de salida definido en V.

Si A y B entonces C: $\mu_C(y) = \mu_{A\cap B}(x)\mu_C(y)$

Y para agregación de varias reglas de este tipo se usa la unión difusa.

3.2.7. Modo generalizado Ponens y Modo generalizado Tollens

En lógica difusa y razonamiento aproximado, hay dos reglas de inferencia difusa muy importantes, nombradas, modo generalizado Ponens (GMP) y modo generalizado Tollens (GMT).

Modo generalizado Ponens (GMP). Está definido como indica el siguiente procedimiento de inferencia:

Precedente 1: x es A'

Precedente 2: SI x es A, ENTONCES y es B

Consecuente: y es B'

Donde A', A, B y B' son conjuntos difusos, y *x* y *y* son variables lingüísticas.

Modo generalizado Tollens (GMT). Está definido como indica el siguiente procedimiento de inferencia:

Precedente 1: y es B'

Precedente 2: SI x es A, ENTONCES y es B

Consecuente: x es A'

Donde A', A, B' y B son conjuntos difusos, *y* x *y* y son variables lingüísticas.

3.3. Sistemas difusos

Un sistema adaptable difuso se define como un sistema lógico difuso equipado con un algoritmo de entrenamiento, donde el sistema lógico difuso está construido con un conjunto de reglas difusas IF-THEN empleando principios de lógica difusa, y el algoritmo de entrenamiento ajusta los parámetros (y las estructuras) del sistema lógico difuso basado en información numérica.

Existen dos estrategias para combinar información numérica con lingüística usando sistemas difusos:

3.3.1. Sistema difuso de Takagi y Sugeno

Takagi y Sugeno proponen el uso de las siguientes reglas difusas SI-ENTONCES

$$F^{(l)} : Si\_x_1\_es\_F_1^l\_y \ldots y\_x_n\_es\_F_n^l, entonces\_y^l = c_0^l + c_1^l x_1 + \ldots + c_n^l x_n$$

Donde F_j^l son conjuntos difusos, c_j son parámetros reales, y^l es la salida del sistema debida a la regla $L^{(l)}$, y l=1,2,…,M. esto es, se consideran reglas, cuyas partes SI es difusa pero cuya parte ENTONCES es definida, la salida es una combinación lineal de las variables de entrada. Para un vector de entradas reales $\underline{x}=(x_1,\ldots,x_n)^T$, la salida y ($\underline{x}$)del sistema difuso de Takagi y Sugeno es un promedio pesado de las y^l's:

$$y(\underline{x}) = \frac{\sum_{l=1}^{M} w^l y^l}{\sum_{l=1}^{M} w^l}$$

Donde el peso w^l implica el valor sobre verdadero de la regla previa $L^{(l)}$ para la entrada y se calcula como sigue

$$w^l(\underline{x}) = \prod_{j=1}^{n} \mu_{F_j^l}(x_j)$$

La configuración del sistema difuso de Takagi y Sugeno se muestra a continuación.

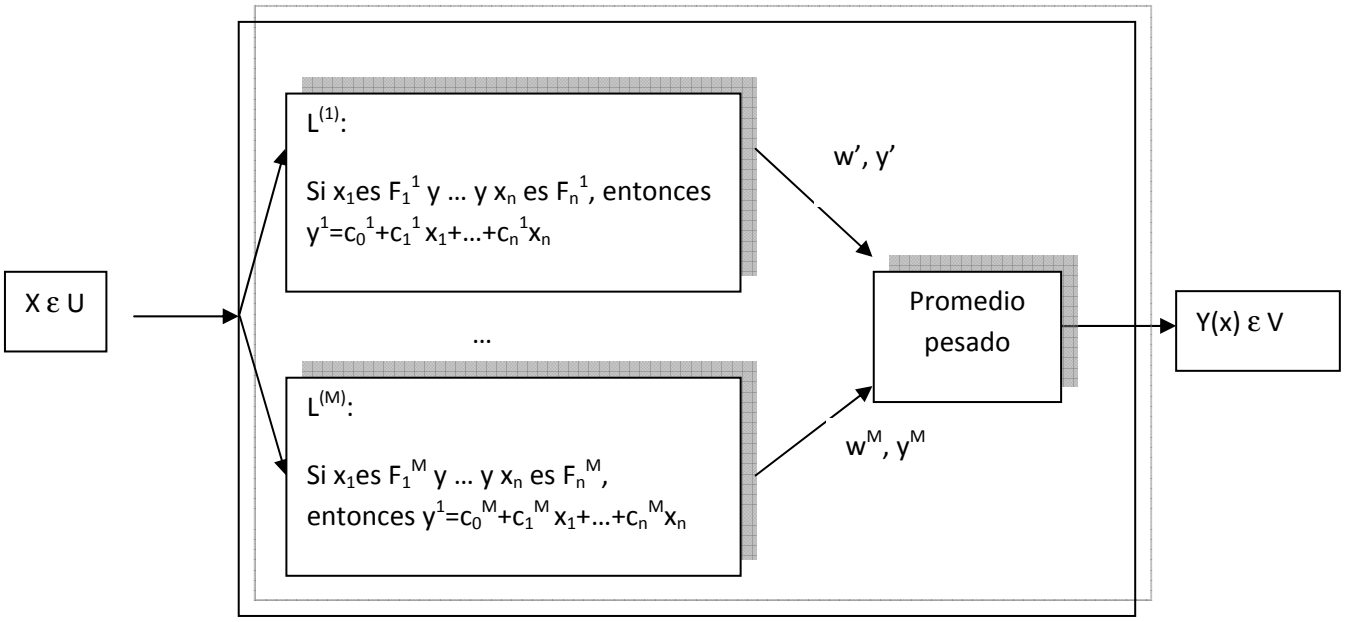


Figura 3.9. Configuración del sistema difuso de Takagi y Sugeno

La ventaja de este tipo de sistema lógico difuso es que provee una ecuación del sistema compacta, y asimismo, los métodos de estimación de parámetros y determinación del orden pueden ser desarrollados para estimar los parámetros c_j^l y el orden M. Un punto débil de estos sistemas lógicos difusos es que la parte ENTONCES de la regla no es difusa; por lo que, no provee una referencia natural para incorporar reglas difusas de humanos expertos. Además, no hay mucha libertad para aplicar diferentes principios en la lógica difusa.

3.3.2. *Máquina de inferencia difusa con difusor y desdifusor*

Una Máquina de inferencia difusa, es un sistema lógico difuso, que combina información numérica y lingüística por medio de reglas SI-ENTONCES completamente difusas.

La configuración básica de un sistema basado completamente en lógica difusa, dónde la base de reglas difusas consiste de un conjunto de reglas difusas SI-ENTONCES, y la máquina de inferencia difusa usa esta base para determinar un mapeo de conjuntos difusos en la entrada del universo del discurso $U \subset R^n$ en conjuntos difusos a la salida en el universo del discurso $V \subset R$ basado en principios de lógica difusa.

Las reglas difusas SI-ENTONCES son de la siguiente forma:

$R^{(l)}$: SI x_1 es F_1^l y … y x_n es F_n^l, ENTONCES y es G^l

Donde F_j^l y G^l son conjuntos difusos, $\underline{x}=(x_1,\ldots,x_n)^T \in U$ y $y \in V$ son variables lingüísticas de entrada y salida, respectivamente, y l=1,2,…,M.

Cada regla difusa define un conjunto difuso $F_1^l x \ldots x F_n^l \rightarrow G^l$ en el producto de espacio UxV.

El sistema completamente difuso constituye la parte esencial de los sistemas lógicos difusos. Una desventaja importante de estos sistemas es que son sistemas que manejan información lingüística, por lo que tanto sus entradas y como sus salidas son conjuntos difusos; mientras que casi todos los sistemas en ingeniería emplean valores numéricos tanto en las entradas como en las salidas, por lo que se complementa, agregando un bloque de difusión y uno de desdifusión que le permiten interactuar con información numérica. El difusor mapea puntos definidos en U en conjuntos difusos de U, y el desdifusor mapea conjuntos difusos de V en puntos definidos de V.

La máquina de inferencia difusa con difusor y desdifusor cuenta con características que le proporcionan ventajas frente a otros sistemas.

- Se ajusta a sistemas de ingeniería porque sus entradas y salidas son variables reales valuadas.
- Provee un marco natural para incorporar reglas difusas SI-ENTONCES de humanos expertos.
- Hay libertad para elegir un difusor, máquina de inferencia difusa, y desdifusor, de modo que podemos obtener el sistema lógico difuso que mejor se ajuste a un problema en particular.
- Permite el desarrollo de distintos algoritmos de entrenamiento para estos sistemas para incorporar un marco efectivo para integrar información numérica y lingüística.

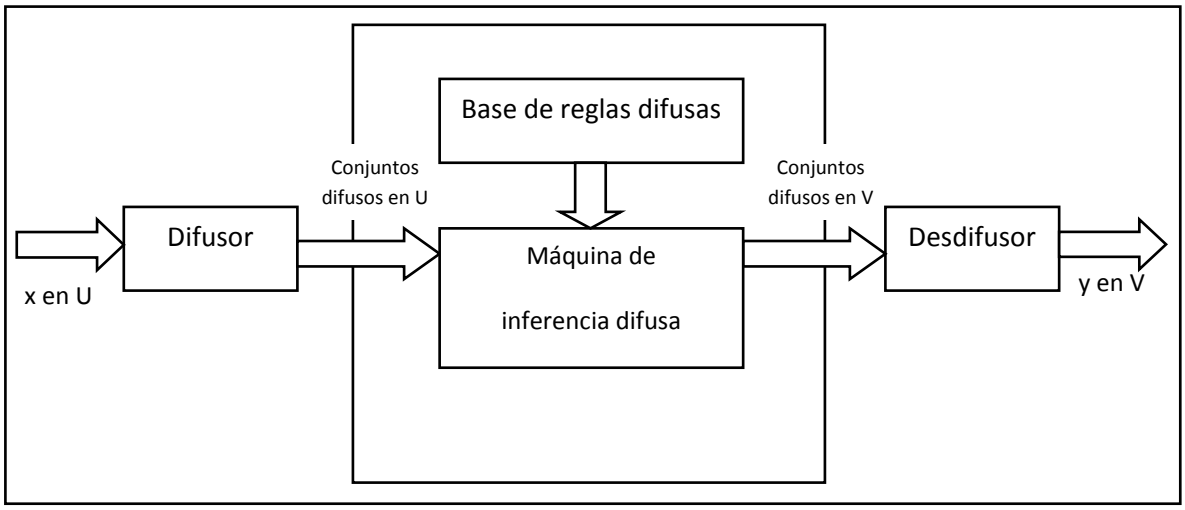


Figura 3.10. Máquina de inferencia difusa con difusor y desdifusor

3.3.2.1 Máquina de inferencia difusa

Una máquina de inferencia que utiliza los principios de la lógica difusa para combinar las reglas difusas SI-ENTONCES de la base de reglas difusas dentro de un mapeo de conjuntos difusos en $U= U_1 x \ldots x U_n$ a conjuntos difusos en V se denomina máquina de inferencia difusa.

3.3.2.2. Difusor

El difusor mapea un punto definido $\underline{x}=(x_1,\ldots,x_n)^T \in U$ en un conjunto difuso A' en U. Existen varias posibles elecciones para este mapeo, que dependen de las formas de las funciones de pertenencia de los conjuntos difusos para las variables de entrada.

En general podemos clasificar al difusor en difusor con singletons, con funciones triangulares o con funciones de campana, de acuerdo al tipo de funciones que se elijan. Se piensa que un difusor que no usa singleton puede ser útil si las entradas están contaminadas con ruido.

3.3.2.3. Desdifusor

El desdifusor mapea de conjuntos difusos de V en puntos definidos $y^* \in V$. Hay posibles elecciones para este mapeo, de entre los cuales, los más representativos son:

- Centroide

$$y^* = \frac{\int \mu(y)\, y dy}{\int \mu(y) dy}$$

- Centro de gravedad (COG)

$$y^* = \frac{\sum \mu(y_i)\, y_i}{\sum \mu(y_i)}$$

Definido para funciones singleton en los conjuntos difusos de salida en los puntos y_i.

3.3.2.4. Base de reglas difusa

Una base de reglas difusas consiste en una colección de reglas difusas SI-ENTONCES de la forma siguiente:

$R^{(l)}$: SI x_1 es F^l_1 y … y x_n es F^l_n ENTONCES y es G^l

Dónde F^l_j y G^l son conjuntos difusos en $U_j \subset R$ y $V \subset R$, respectivamente, y $\underline{x}=(x_1,\ldots,x_n)^T \in U_1 x \ldots x U_n$ y $y \in V$ y siendo variables lingüísticas.

Sea M el número de reglas difusas SI-ENTONCES expresadas de la forma anterior en la base de reglas difusas; esto es, j=1,2,…,M. La x y y son la entrada y la salida del sistema lógico difuso, respectivamente.

Para no perder generalidad se consideran sistemas lógicos difusos de salidas y entradas múltiples o únicas, ya que un sistema de múltiples salidas siempre puede ser descompuesto en un grupo de sistemas de salida única.

La base de reglas difusas es el corazón del sistema lógico difuso, en el entendido de que los otros componentes son usados para interpretar las reglas y hacerlas útiles para problemas específicos. La práctica ha mostrado que las reglas difusas SI-ENTONCES expresadas correctamente, proveen una solución muy conveniente para los humanos expertos para expresar su conocimiento dominante.

3.3. Consideraciones sobre los sistemas difusos

Debido a que hay distintas formas de interpretar las reglas difusas SI-ENTONCES que conllevan a distintos mapeos de la máquina de inferencia difusa, también hay distintos tipos de difusores y desdifusores. Muchas combinaciones de estas máquinas de inferencia difusas, difusores, y desdifusores puede constituir sistemas lógicos difusos útiles.

Por lo general se tienen los siguientes criterios para elegir las reglas de inferencia difusa, el difusor, el desdifusor y las formas del funcional para las funciones de pertenencia de los conjuntos difusos en las reglas difusas SI-ENTONCES.

Ajuste empírico. Ya que los sistemas lógicos difusos se usan para incorporar información lingüística, es importante que las elecciones generen modelos apropiados del comportamiento de sistemas reales.

Eficiencia computacional. Para muchos problemas o para una potencia limitada de cómputo, se puede seleccionar sistemas lógicos difusos más simples.

Para los difusores propuestos, el difusor con singleton proporciona sistemas lógicos difusos más simples, mientras que los demás difusores pueden trabajar mejor en ambientes ruidosos.

Para los desdifusores, tanto el método de centroides (contínuo) como el de centro de gravedad (discreto) proporcionan soluciones satisfactorias para las cuales difieren la complejidad de cálculo.

Para la elección de funciones de pertenencia de F_j^l y G^l, curvas suaves, triangulares y trapezoidales son los tipos más usados.

3.5. Sistemas lógicos difusos como aproximadores universales

Existe un teorema que muestra que los sistemas lógicos difusos son capaces de aproximar uniformemente cualquier función no lineal sobre U a cualquier grado de exactitud, si U es compacto.

Teorema de aproximación universal. Para cualquier función real y continua g sobre un conjunto compacto $U \subset R^n$ y una $\varepsilon > 0$ arbitraria, existe un sistema lógico difuso f (con desdifusor de centroide, funciones de pertenencia con curvas suaves y difusor con singleton) tal que:

$$\sup_{\underline{x} \in U} | f(\underline{x}) - g(\underline{x}) | < \varepsilon$$

Con respecto al teorema de aproximación universal, destaca que:

- Este teorema es una justificación para la aplicación de sistemas lógicos difusos a casi cualquier problema de modelado no lineal. Además provee una explicación para el éxito práctico de los sistemas lógicos difusos en aplicaciones de ingeniería.
- Este teorema muestra que existe un sistema lógico difuso que puede aproximar uniformemente cualquier función dada con una exactitud arbitraria.

Capítulo 4
Implementación

En este capítulo se describe detalladamente cada aspecto de la implementación del cancelador adaptable de ruido con lógica difusa, se parte de la adquisición de las señales de audio, explicando el diseño que se siguió para la elección de los componentes, así como la descripción de la tarjeta interfaz entre la señal adquirida y la computadora dónde, por último, el programa desarrollado en Visual Basic que procesa digitalmente los datos recibidos y obtiene la respuesta final del sistema.

4.1. Descripción del cancelador adaptable de ruido con lógica difusa

La figura 4.1 ilustra los diversos bloques que integran el proyecto de tesis, con una breve indicación de las tareas que desarrolla cada bloque en la parte inferior y en la parte superior la herramienta empleada para su elaboración.

El primer bloque lo conforma un procesamiento analógico por medio de dispositivos electrónicos pasivos y activos para transducir y acondicionar las señales de audio que se procesarán.

El segundo bloque está integrado por la interfaz realizada con un microcontrolador PIC, que se encarga de convertir la señal analógica en digital y comunicarlas a la PC por medio de conexión USB.

El bloque final compuesto por un programa en lenguaje Visual Basic, se encarga de realizar el almacenamiento, procesamiento y obtención de la señal de salida de las señales de audio adquiridas.

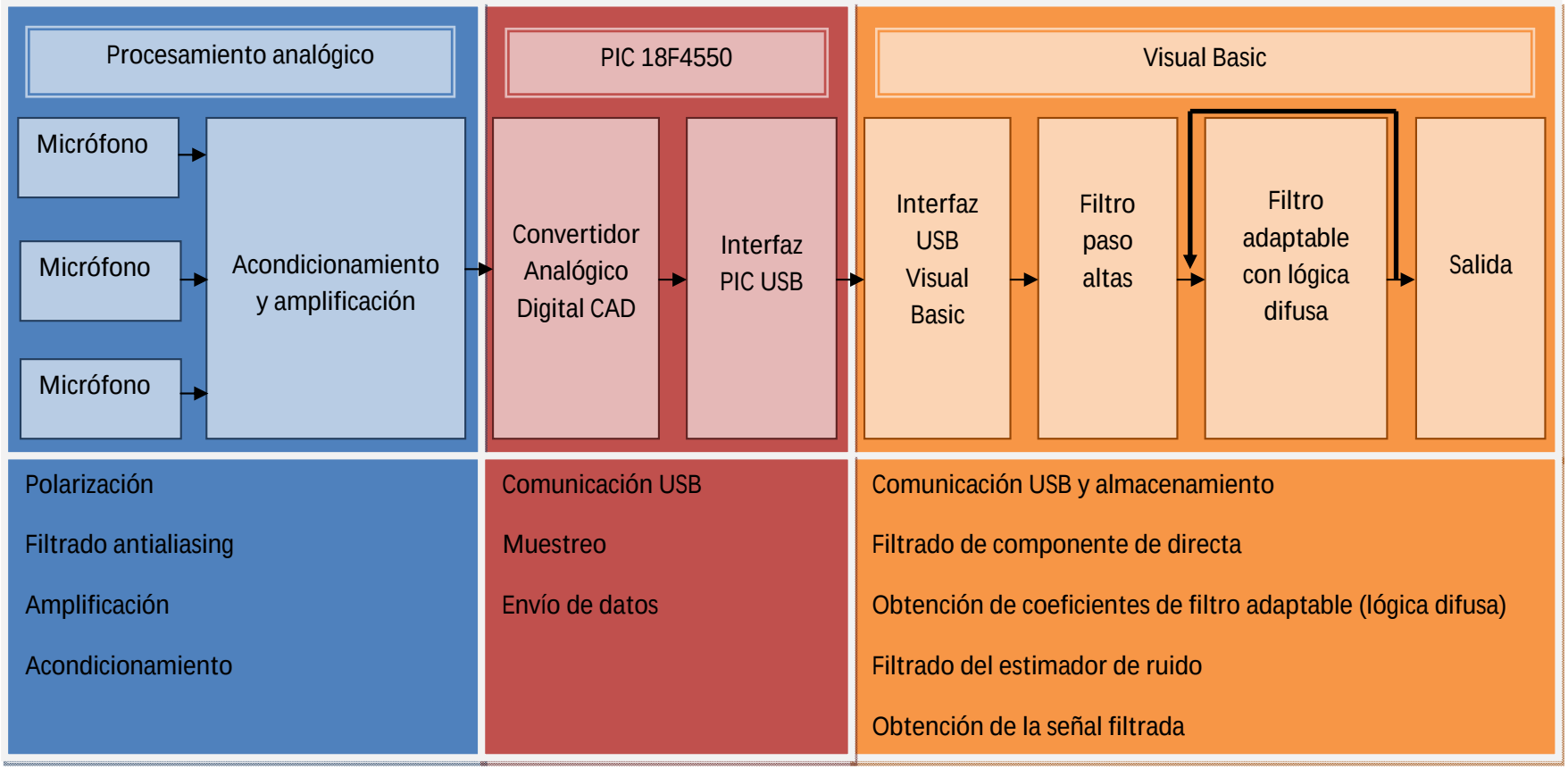


Figura 4.1. Bloques del filtro adaptable con lógica difusa

4.2. Transducción y acondicionamiento de las señales de voz

4.2.1. Micrófonos electret

En la entrada al sistema se tienen tres micrófonos electret, dos actúan como micrófonos de referencia captando el ruido del entorno y uno como micrófono primario abocado a obtener la señal de interés junto con el ruido del entorno en el que ésta se encuentra. Se eligió este tipo de micrófono por su bajo costo, su versatilidad y su reducido tamaño, además su respuesta en frecuencia (50 a 15.000 Hz) está dentro del rango de frecuencias en las que trabajará el sistema.

Los micrófonos electret son un tipo particular de micrófonos capacitivos denominados así debido a que su sección de captación de audio tiene la estructura de un capacitor formado por un diafragma y una placa posterior como se observa en la figura 4.2.

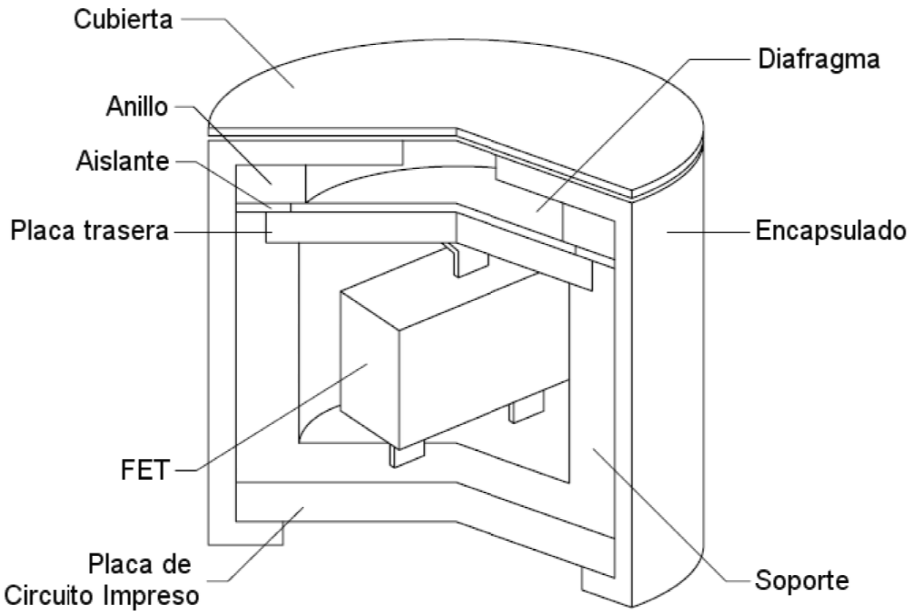


Figura 4.2. Diagrama estructural de un micrófono electret.

Las ondas sonoras provocan el movimiento oscilatorio del diafragma ocasionando con ello una variación en la capacitancia existente entre el diafragma y la placa posterior, originando así una corriente que aplica al gate del FET incorporado en el encapsulado. La caída de tensión producida, es proporcional a la presión incidente debida al sonido. La base del funcionamiento de este micrófono es la variación de la energía eléctrica almacenada entre las placas, por lo que cuanto mayor sea esta energía, o lo que es igual, la tensión continua de polarización que la produce, mayor será la salida de tensión (sensibilidad) del micrófono.

Para que exista un efecto capacitivo entre el diafragma y la placa posterior se necesita que haya una polarización presente, por lo general, una tensión de decenas o cientos de volts, sin embargo, la

carga eléctrica puede mantenerse en una película de polímero por el efecto Electret[1], eliminando así la polarización de corriente directa de alto voltaje.

Debido a que el micrófono consiste en un capacitor de muy baja capacitancia presenta una impedancia de salida elevada, especialmente en baja frecuencia. Esto da lugar a fuertes pérdidas de tensión en la transmisión a través del cable de conexión, en cuanto este cable supere longitudes de aproximadamente un metro. Para evitar estas pérdidas se inserta, inmediatamente detrás de la cápsula del capacitor, un pre amplificador acoplador de impedancias, que presenta al micrófono una muy alta impedancia y a la línea de transmisión una baja impedancia, del orden de 200 ohms. Este previo presenta una alta impedancia al micrófono, gracias a utilizar como elemento activo un transistor de efecto de campo (FET) o un tubo de vacío; en ambos casos será necesario alimentar a dicho elemento activo con una tensión continua, que se suele extraer de la polarización del micrófono. De aquí la necesidad de mantener el signo de la polarización que indique el fabricante.

Dicho transistor de efecto de campo (FET) se incluye dentro del encapsulado del micrófono, acoplando su gate al diafragma del electret, el drain se toma como terminal de salida y el source se conecta a tierra como se muestra en la figura 4.3.

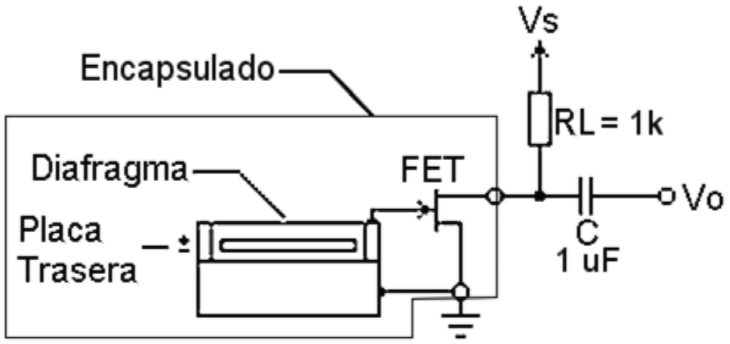


Figura 4.3. Diagrama de conexión del micrófono electret.

Para usar un micrófono electret de 2 terminales, como se hizo en este caso, la salida de señal de audio es conectada a la terminal de suministro (vía una resistencia de corriente restrictiva, el valor típico 1 o 2.2 [kΩ], en este caso utilizamos una de 1[kΩ]). La salida de la señal de audio por lo tanto tiene una componente de corriente continua que es derivada mediante el capacitor de salida, un valor típico puede estar entre 1 y 10 [uF], siendo de 1[uF] el que se utilizó en este trabajo.

Como inconvenientes de este tipo de micrófonos se encuentran:

- Necesidad de suministrar la polarización al FET incluido en el encapsulado.
- Poco robustos debido a que su diafragma es muy frágil.
- Sensibles a la humedad, que puede destruirlos por descarga entre las placas. Poco fiables en exteriores.

[1] Efecto que consiste en lograr polarizar en forma permanente, ciertos materiales calentados y luego enfriados en presencia de un intenso campo eléctrico. Algunas sustancias, como la cerámica titanato de Bario, cera natural u otras orgánicas, pueden usarse para tal fin. El campo eléctrico de un electret, se compara al campo magnético de un imán permanente. [17]

Las ventajas de su uso son:

- Fundamentalmente su respuesta plana hasta las altas frecuencias, lo que les confiere una gran fidelidad junto a un buen comportamiento en agudos.
- Salida de tensión (sensibilidad) elevada. Brindan muy buenas relaciones señal/ruido, aunque también pueden saturar tanto el previo como la entrada a la etapa posterior, si la fuente emite presiones elevadas.
- Por la pequeña masa del diafragma son poco sensibles a las vibraciones y al ruido de manipulación.
- Sus características son estables en el tiempo.
- Tamaño pequeño.

Es necesario que los micrófonos de referencia se sitúen alejados del micrófono primario, pues el interés de éstos recae en buscar obtener la totalidad de ruido ambiental y, preferentemente, no captar la señal de interés, ya que con el sistema propuesto se intenta restar el ruido ambiental a la señal obtenida mediante el micrófono primario. Para mejorar esto, además de la separación de los micrófonos, puede utilizarse micrófonos omnidireccionales situados en puntos alejados como micrófonos de referencia mientras que para el micrófono primario se elegiría un micrófono unidireccional dirigido hacia la fuente de la señal auditiva de interés.

Tras la salida de la señal de cada micrófono electret se requiere acondicionar la señal para las etapas posteriores, es por ello que se acopla una etapa de amplificación descrita a continuación.

4.2.2. Amplificación

La figura 4.4 ilustra la polarización del micrófono electret descrita anteriormente, la amplificación de la señal a modo de acondicionamiento, y el filtrado antialiasing previo a la conversión analógico – digital.

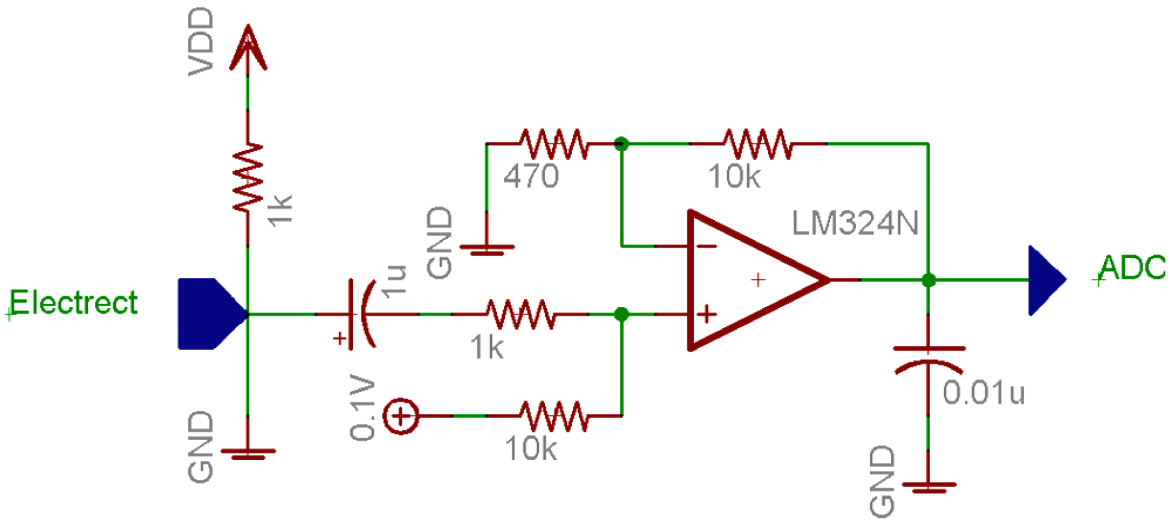


Figura 4.4. Diagrama de acondicionamiento de la señal.

La amplificación de la señal de entrada es realizada por un amplificador operacional (circuito LM324) en configuración sumador no inversor. Al mismo tiempo cumple con la función de agregar una componente de directa a la señal que proviene del micrófono, ya que es necesario que la señal a

muestrear se encuentre dentro del rango de 0 a 5 [V] (debido a los niveles mínimo y máximo del convertidor analógico digital del microcontrolador).

Las características intrínsecas del amplificador operacional, nos proporcionan el perfecto acoplamiento entre la señal proveniente del micrófono electret y el convertidor analógico digital.

El análisis del amplificador empleado representado en la figura 4.5 es:
Por linealidad (superposición)
Vout=Vout1+Vout2
Sea V1=0 y V2≠0
Entonces Vp=V2 ya que I2=0 por la gran impedancia de entrada del Amplificador operacional
De igual forma Iout=I3

$$Vn = (R4)(I3)$$
$$Vout2 = A(Vp - Vn)$$
$$Vout2 = A(V2 - R4i3)$$
$$I3 = Iout = \frac{Vout2}{R3 + R4}$$
$$Vout2 = A\left(V2 - \left(\frac{R4}{R3 + R4}\right)Vout2\right)$$

Despejando Vout

$$Vout2 = \frac{R3 + R4}{R3 + R4 + AR4}V2$$

Considerando que la ganancia en malla abierta del amplificador operacional → ∞

$$Vout2 = \frac{R3 + R4}{R4}V2 = \left(\frac{R3}{R4} + 1\right)V2$$

De forma similar, considerando que V2=0 y V1≠0

$$Vout1 = \left(\frac{R3}{R4} + 1\right)V1$$

Entonces

$$Vout = \left(\frac{R3}{R4} + 1\right)(V1 + V2)$$

Siendo los valores de R3=10kΩ y R4=470Ω y V2=0.1V (tensión fija)
Entonces

$$Vout = 21.2766(V1 + 0.1)$$
$$Vout = 21.2766V1 + 2.12766$$

Esta expresión nos indica que el voltaje que proviene del electret (V1) se amplifica en una razón de 21.2766 y se le agrega una componente de 2.12766 V fijos, es decir, aproximadamente 2.5 V para centrar la señal en el rango de 0 a 5 V y obtener el mejor desempeño del convertidor analógico digital.

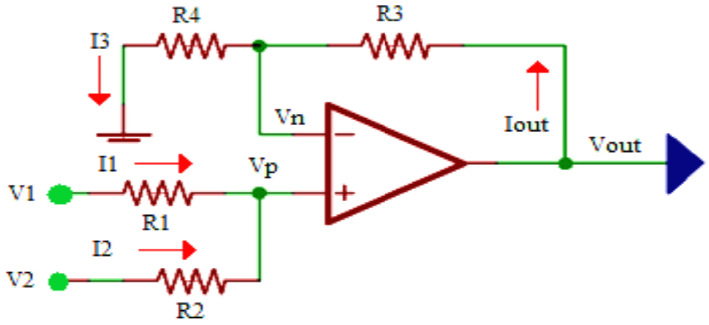


Figura 4.5. Análisis del amplificador sumador no inversor.

La configuración aquí presentada se implementa en los 3 canales de entrada de la interfaz para así poder capturar 3 señales distintas de audio provenientes de 3 micrófonos electrets colocados en una configuración como la maneja el cancelador adaptable de ruido.

Al final de la etapa de amplificación, el circuito acondicionador contiene un capacitor en paralelo a tierra de 0.01uF, el cual sirve para eliminar ruido de alta frecuencia.

4.2.3. Conversión Analógico-Digital

El primer paso de la conversión analógico digital es el muestreo de las señales, para lo que se usa un sistema semejante a los sistemas de multiplexión en el tiempo y de esta forma intercalar el muestreo de las tres señales en el tiempo.

Debido a que la máxima frecuencia de comunicación USB de las muestras es de 64 kbps, la frecuencia máxima de muestreo individual para cada señal es de 21 kbps y de esta forma se limita la conversión analógico digital a muestras de 8 bits, bajo estas dos condiciones se logra una frecuencia de muestreo de 21 kHz para cada señal, lo que implica que la máxima frecuencia que se puede registrar es de 10.5 kHz, que cubre perfectamente el rango de frecuencias de las voz humana.

Para no disminuir la frecuencia de muestreo, se transmiten bloques de 63 muestras (de las tres señales) cada milisegundo vía puerto USB, lo que requiere un sistema de muestreo que se esquematiza en la figura 4.6.

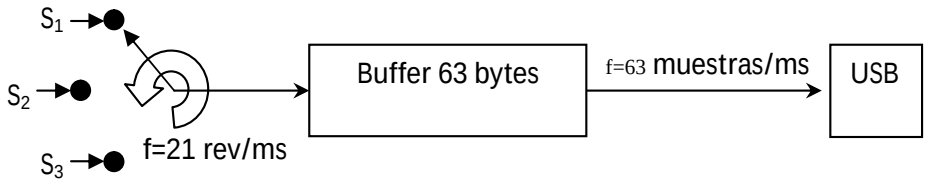


Figura 4.6. Esquema de muestreo con multiplexión en tiempo.

El siguiente paso de la conversión analógico digital es la cuantificación, que lo realiza el ADC interno del PIC, que es un convertidor de 10 bits, de los cuáles, solamente se usan los 8 bits más significativos, ya que usar más bits por muestra, reduciría la velocidad de comunicación y por lo tanto reduciría la frecuencia de muestreo de las señales.

La figura 4.7 ilustra la función del cuantificador del PIC18F4550 dónde se aprecia que se trata de un cuantificador uniforme, que aunque no es ideal para señales de audio, tiene un desempeño efectivo para ser complementado con la comunicación USB del microcontrolador.

Esta etapa es desempeñada completamente por el PIC18F4550, cuyo programa está desarrollado en lenguaje C, con ayuda de programas de ejemplo que incluye compilador CCS, que sirvieron de ayuda para desarrollar la interfaz USB HID que se describe a grandes rasgos más adelante.

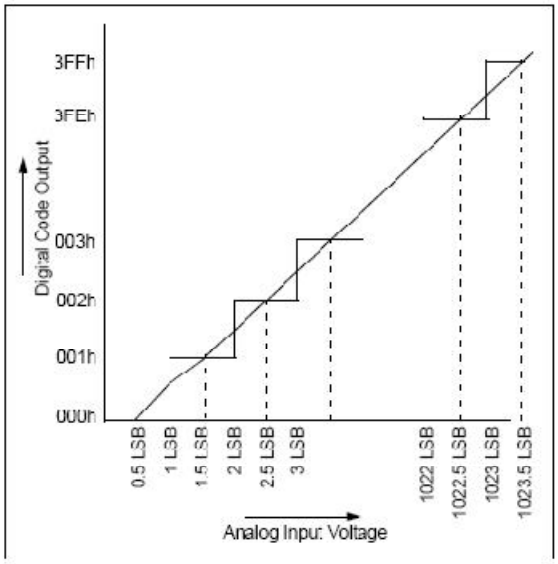


Figura 4.7. Función de transferencia Analógico Digital del PIC 18F4550.

El programa del PIC referente al muestreo se presenta a continuación:

```
void usb_debug_task(void) {
  static int8 last_connected;
  static int8 last_enumerated;
  int8 new_connected;
  int8 new_enumerated;

  new_connected=usb_attached();
  new_enumerated=usb_enumerated();

  if (new_enumerated)
    LED_ON(MILED);
  else
    LED_OFF(MILED);

  if (new_connected && !last_connected)
    printf("\r\n\nUSB connected, waiting for enumaration...");
  if (!new_connected && last_connected)
    printf("\r\n\nUSB disconnected, waiting for connection...");
  if (new_enumerated && !last_enumerated)
    printf("\r\n\nUSB enumerated by PC/HOST");
  if (!new_enumerated && last_enumerated)
    printf("\r\n\nUSB unenumerated by PC/HOST, waiting for enumeration...");

  last_connected=new_connected;
  last_enumerated=new_enumerated;
}
void main() {
  int8 out_data[70];
  int8 in_data[2];
  int8 send_timer=0;
  int8 cuenta=0;
  LED_OFF(MILED);
  LED_OFF(LED2);
  LED_OFF(LED3);
 #ifdef __PCH__
  printf("\r\nPCH: v");
  printf(__PCH__);
 #else
  printf("\r\n\PCM: v");
  printf(__PCM__);
 #endif
  usb_init_cs();
 #if !(__USB_PIC_PERIF__)
  printf("\r\nUSBN: 0x%X", usbn_get_version());
 #endif
  printf("\r\n");
  setup_adc_ports(AN0_TO_AN3);
setup_adc(ADC_CLOCK_INTERNAL);

  while (TRUE) {
    usb_task();
    usb_debug_task();
    if (usb_enumerated()) {
        for (cuenta=0;cuenta<=62;cuenta=cuenta+3)
        {
          set_adc_channel(1);
          out_data[cuenta]=read_adc();
          set_adc_channel(0);
          out_data[cuenta+1]=read_adc();
          set_adc_channel(2);
          out_data[cuenta+2]=read_adc();
        }
              usb_put_packet(1,out_data,63,USB_DTS_TOGGLE);

              if (usb_kbhit(1)) {
                        usb_get_packet(1, in_data, 2);
               if (in_data[0]) {LED_ON(LED2);} else {LED_OFF(LED2);}
              if (in_data[1]) {LED_ON(LED3);} else {LED_OFF(LED3);}
      }
     send_timer--;
   }
  }
}
```

Este código muestra un ciclo que muestrea, almacena en memoria y envia los datos cada que se juntan 63 muestras.

4.2.4. Comunicación USB

El puerto USB que presenta las siguientes ventajas:

- Velocidad de transferencia de hasta 480 Mbps.
- Un puerto USB permite conectar hasta 127 dispositivos.
- Los dispositivos USB suelen ser *plug and play*, esto es, al conectar el dispositivo USB a la computadora esta lo reconoce de forma automática pudiendo instalar y configurar los drivers que el dispositivo requiere para funcionar.
- El puerto brinda una polarización a los dispositivos conectados a él de 5 [V] y hasta 500 [mA].
- Es un puerto que actualmente se encuentra en la inmensa mayoría de las computadoras siendo soportado por la mayoría de los sistemas operativos actuales.
- Permite tener dispositivos con la modalidad *hotswap*, esto hace que el dispositivo pueda desconectarse y reconectarse físicamente de la computadora sin que sufra daños.

Como desventajas nos encontramos con que:

- La implementación de la interfaz USB es más compleja que la de otros puertos como el serie o el paralelo.
- El ancho de banda debe repartirse entre los dispositivos conectados a un mismo puerto USB.

Para la correcta comunicación mediante el puerto usb fue necesario trabajar en el desarrollo de tres aspectos sobre la interfaz:

1. El *hardware*. Se refiere a la implementación física: conectores, cables, etc.
2. El *software*. Está conformado por el programa desarrollado en Visual Basic que actúa como interfaz dentro del sistema operativo Windows para que se pueda manipular la conexión de la tarjeta conectada al puerto USB.
3. El *firmware*. Es la parte conformada por el programa cargado dentro del pic 18f4550 en el que se implementa el protocolo USB para la interfaz.

El tipo de dispositivo USB empleado para el proyecto, es un dispositivo HID con número de fabricante de microchip &H461 y número de producto &H20.

Debido a que el puerto USB puede entregar hasta 500mA, se usa la misma alimentación del puerto para alimentar a la interfaz que captura las señales de audio.

Las etapas descritas hasta este punto son implementadas electrónicamente en una tarjeta de desarrollo propio, el diagramo completo de la tarjeta desarrollada para el proyecto se presenta en las figuras 4.8 y 4.9.

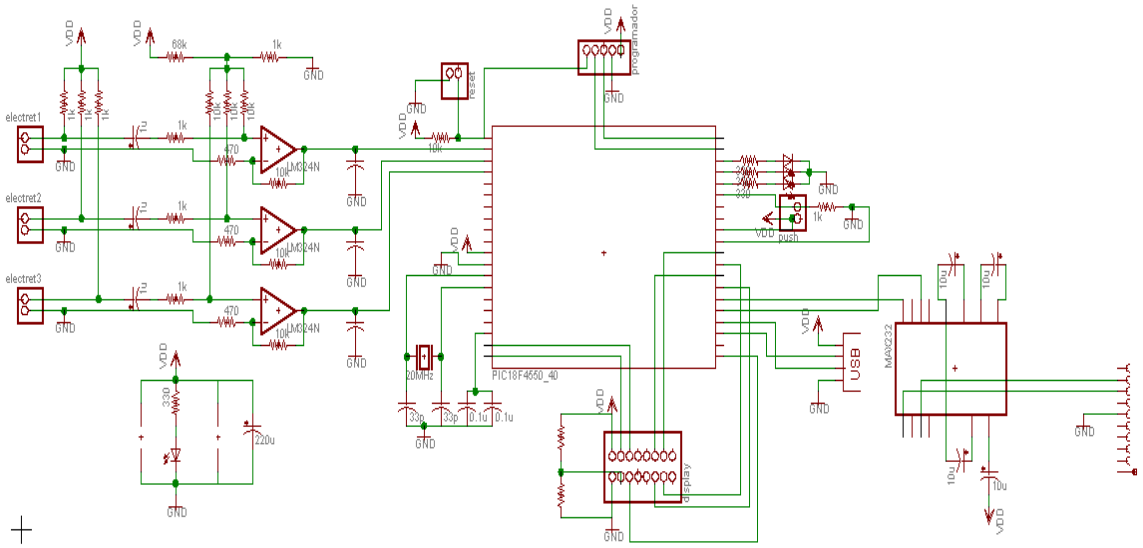


Figura 4.8. Diagrama eléctrico de la tarjeta.

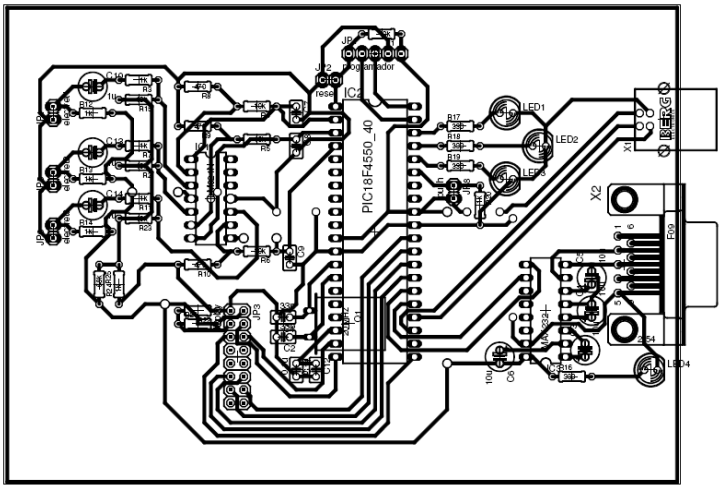


Figura 4.9. Diagrama del PCB.

4.2.5. Filtro para eliminar componente de directa

Debido a que en la etapa de acondicionamiento de la señal, se le agrega una componente de directa para poder ser muestreada por el CAD del PIC, es necesario eliminar esta componente para trabajar únicamente con las señales de audio. Debido a lo anterior, el primer paso, antes de procesar más las señales, es implementar digitalmente un filtro tipo Butterworth paso altas que elimine la componente de directa.

Los filtros digitales presentan ventajas sobre los filtros analógicos como son: alta inmunidad al ruido, gran precisión (limitada únicamente por errores de redondeo), se pueden modificar fácilmente, su costo es muy bajo.

La implementación digital de un filtro Butterworth implica un filtro IIR (Infinite Impulse Response), que al contrario de los filtros FIR (Finite Impulse Response), son considerablemente

más fáciles al llevarlos a la práctica, y con un orden de filtro, relativamente menor al orden de los filtros FIR, se puede obtener una respuesta óptima.

Se eligió un filtro Butterworth para esta etapa del procesamiento, ya que nos proporciona una ganancia prácticamente constante en la banda de paso y en la banda de rechazo, aunque la banda de transición no sea muy pequeña.

El diseño clásico de un filtro digital, tal como se ha descrito, requiere primero del diseño de un filtro analógico, para lo que se deben especificar las características principales de su respuesta en frecuencia.

El diseño analógico se realiza a partir de unas especificaciones como las dadas en la figura.

δ_1 es el rizado de la banda de paso.
δ_2 es el rizado de la banda de rechazo.
f_p es la frecuencia límite de la banda de paso.
f_s es la frecuencia límite de la banda de rechazo.

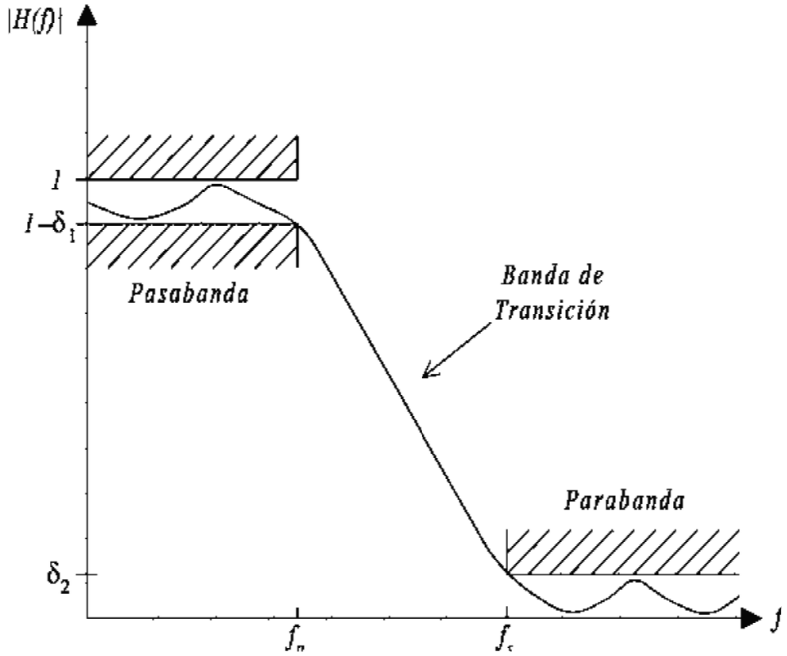


Figura 4.10. Características de la respuesta en frecuencia de un filtro paso bajas.

Partimos de un prototipo de filtro pasobajas normalizado en el que usamos una frecuencia n normalizada. Para otro tipo de filtro se requerirá la consiguiente transformación de frecuencia. Para ese filtro pasobajas normalizado la función de Transferencia es

$$H(s) = \frac{G_0}{\prod_{k=0}^{N-1} \frac{s - s_k}{\Omega_c}}$$

Los polos son de la forma

$$s_k = \Omega_c e^{j\frac{\pi}{2}} e^{j\left(\frac{(2k+1)\pi}{2N}\right)}, k = 0,1,\dots,N-1$$

Con las siguientes características:

– Los polos normalizados están sobre un círculo de radio 1 en el plano s.

–Los polos están equiespaciados p/n radianes con $q_k=(2k-1)p/2n$, donde q_k se mide con respecto al eje positivo del eje jω.

– Los polos nunca estarán sobre el eje jω (2k-1 nunca puede ser par).

– Si n es impar, siempre hay un par de polos reales en s=±1.

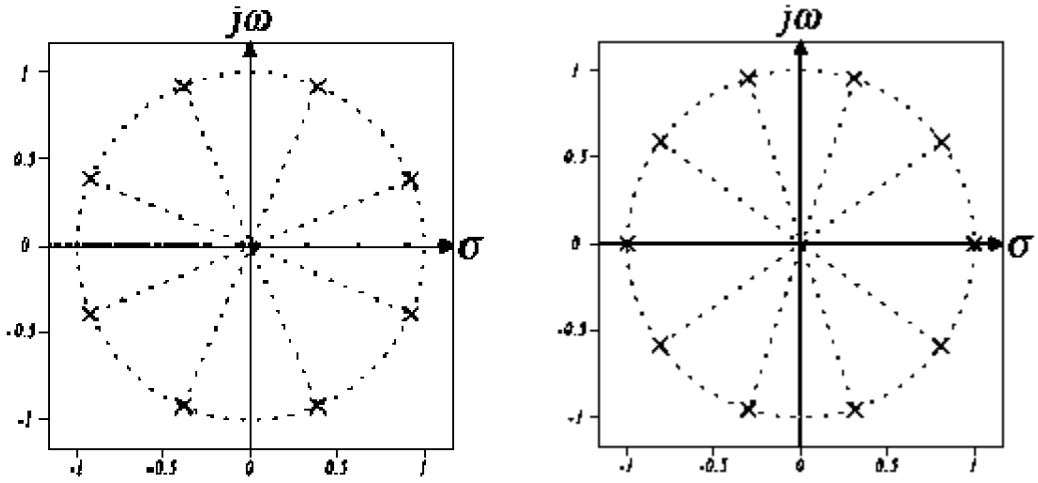


Figura 4.11. Distribución de polos para filtros Butterworth de orden n=4 y n=5 respectivamente.

De los 2n polos de que consta H(s)H(-s) sólo estamos interesados en los polos de la parte izquierda del plano s, que son los que dan estabilidad al filtro.

Se debe convertir el filtro paso bajas en un filtro paso altas mediante las relaciones de mapeo

$$s = \frac{\omega_x}{s}, \omega_x = \frac{\omega_{nueva}}{\omega_{anterior}}$$

Una vez que se cuentan con los polos del filtro analógico, se mapean estos polos en polos en el plano Z por medio de la transformación bilineal.

$$s = \frac{2}{T}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right)$$

Una vez en el plano Z, se expresa la función de transferencia del filtro en términos de la entrada y salida actuales y anteriores, para expresar la salida de forma recursiva.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

$$Y(n) = -\sum_{K=1}^{N} a_k y(n-k) + \sum_{K=0}^{M} b_k x(n-k)$$

Teniendo todo esto en consideración, se empleó MATLAB como ayuda para el diseño de este filtro con ayuda de las siguientes funciones:

<Butter> diseña un filtro tipo Butterworth digital y analógico

[B,A] = BUTTER(N,Wn,'high') diseña un filtro Butterworth digital paso altas de orden N y regresa los coeficientes en vectores B (numerador) y A (denominador) de longitud N+1. Los coeficientes están listados en potencias descendientes de z. La frecuencia de corte Wn debe estar acotada entre 0.0<Wn<1.0 correspondiendo a la mitad de la frecuencia de muestreo.

<FREQZ> Obtiene la respuesta en frecuencia de un filtro digital
H=FREQZ(B,A,W) regresa el vector de n puntos complejos de respuesta en frecuencia H para las frecuencias designadas en el vector W en [rad/muestra] (normalizado entre 0 y π) del filtro cuyo vector de coeficientes de numerador es B y A es el vector de coeficientes del denominador de la función de transferencia H(z).

Así se procede a obtener los coeficientes del numerador y del denominador de la función de transferencia H(z) del filtro de 4º orden, paso altas, con frecuencia de corte en f_c=(0.1)(10.5kHz)=1050Hz.

```
>> [B,A]=butter(4,0.1,'high')
B =0.6620   -2.6481    3.9721   -2.6481    0.6620
A =1.0000   -3.1806    3.8612   -2.1122    0.4383
```

Y enseguida se obtiene la respuesta del filtro en MatLab

```
>>W=0:0.0001:3.14;
>>H=freqz(B,A,W);
>>W=0:10500/len(W):10499.9;
>>plot(W,abs(H))
```

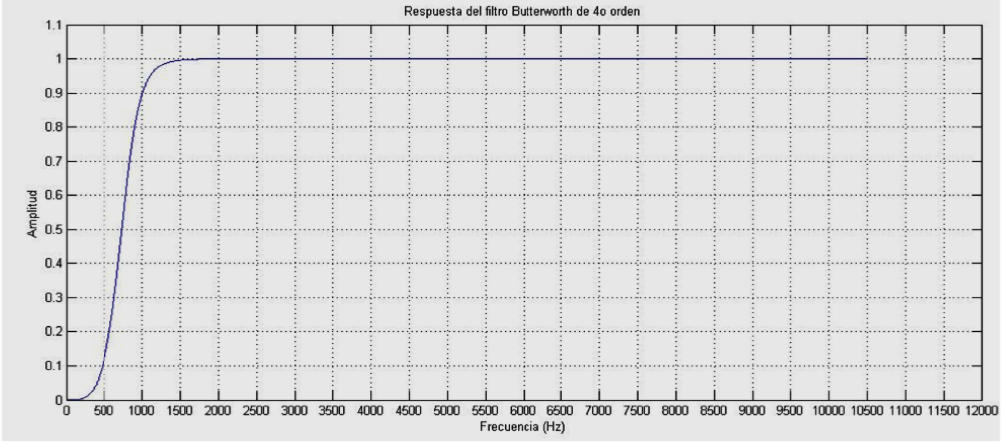


Figura 4.12. Respuesta en frecuencia del filtro Butterworth de cuarto orden.

Al usar las instrucciones descritas anteriormente en MATLAB, los coeficientes del filtro son:

```
B =0.6620   -2.6481    3.9721   -2.6481    0.6620
A =1.0000   -3.1806    3.8612   -2.1122    0.4383
```

Esto quiere decir:

$$H(z) = \frac{0.662z^4 - 3.3441z^3 + 4.2389z^2 - 2.4093z + 0.5175}{z^4 - 3.1806z^3 + 3.8612z^2 - 2.1122z + 0.4383}$$

O expresada en potencias negativas de z

$$H(z) = \frac{0.662 - 3.3441z^{-1} + 4.2389z^{-2} - 2.4093z^{-3} + 0.5175z^{-4}}{1 - 3.1806z^{-1} + 3.8612z^{-2} - 2.1122z^{-3} + 0.4383z^{-4}}$$

Y en términos de n

$$y(n) = 3.1806y(n-1) - 3.8612y(n-2) + 2.1122y(n-3) - 0.4383y(n-4) + 0.662x(n) - 3.3441x(n-1) + 3.8612x(n-2) - 2.1122x(n-3) + 0.5175x(n-4)$$

Esta es la versión recursiva del filtro IIR Butterworth, que depende de salidas anteriores y entradas anteriores, lo que no representa ningún problema, ya que se almacenan las señales de entrada, por lo que se tienen en memoria todos los valores de las señales para todos los tiempos nT donde T es el periodo de muestreo (1/Fs) y las condiciones iniciales del filtro son ceros.

En visual Basic, el código empleado se muestra a continuación.

```
   Open "ar1.txt" For Binary Access Read As #1
   Open "ar2.txt" For Binary Access Read As #2
   Open "ar3.txt" For Binary Access Read As #3
   Open "ar1out.txt" For Binary Access Write As #11
   Open "ar2out.txt" For Binary Access Write As #12
   Open "ar3out.txt" For Binary Access Write As #13

   sig = 1
   xn(0) = xn(1) = xn(2) = 0
   'Condicones iniciales en cero
   y(0) = y(1) = y(2) = y(3) = y(4) = 0
   x(0) = x(1) = x(2) = x(3) = x(4) = 0
   y1(0) = y1(1) = y1(2) = y1(3) = y1(4) = 0
   x1(0) = x1(1) = x1(2) = x1(3) = x1(4) = 0
   y2(0) = y2(1) = y2(2) = y2(3) = y2(4) = 0
   x2(0) = x2(1) = x2(2) = x2(3) = x2(4) = 0
   While sig <= LOF(1)
   'lectura en el archivo1
   Get #1, sig, xn(0)
   x(0) = 50 * xn(0)
   y(0) = 0.662 * x(0) - 2.6481 * x(1) + 3.9721 * x(2) - 2.6481 * x(3) + 0.662
* x(4) + 3.1806 * y(1) - 3.8612 * y(2) + 2.1122 * y(3) - 0.4383 * y(4)
   y(1) = y(0)
   y(2) = y(1)
   y(3) = y(2)
   y(4) = y(3)
   x(1) = x(0)
   x(2) = x(1)
   x(3) = x(2)
   x(4) = x(3)

   'lectura en el archivo2
   Get #2, sig, xn(1)
   x1(0) = 50 * xn(1)
   y1(0) = 0.662 * x1(0) - 2.6481 * x1(1) + 3.9721 * x1(2) - 2.6481 * x1(3) +
0.662 * x1(4) + 3.1806 * y1(1) - 3.8612 * y1(2) + 2.1122 * y1(3) - 0.4383 *
y1(4)
   y1(1) = y1(0)
   y1(2) = y1(1)
   y1(3) = y1(2)
   y1(4) = y1(3)
   x1(1) = x1(0)
   x1(2) = x1(1)
   x1(3) = x1(2)
   x1(4) = x1(3)

   'lectura en el archivo3
   Get #3, sig, xn(2)
   x2(0) = 50 * xn(2)
   y2(0) = 0.662 * x2(0) - 2.6481 * x2(1) + 3.9721 * x2(2) - 2.6481 * x2(3) +
0.662 * x2(4) + 3.1806 * y2(1) - 3.8612 * y2(2) + 2.1122 * y2(3) - 0.4383 *
y2(4)
   y2(1) = y2(0)
   y2(2) = y2(1)
   y2(3) = y2(2)
   y2(4) = y2(3)
   x2(1) = x2(0)
   x2(2) = x2(1)
   x2(3) = x2(2)
   x2(4) = x2(3)

   'se gurada dato en archivo 1
      save = 5 * Int(y(0))
      Put #11, , save
   'se gurada dato en archivo 2
      save = 5 * Int(y1(0))
      Put #12, , save
   'se gurada dato en archivo 3
      save = 5 * Int(y2(0))
      Put #13, , save

   sig = sig + 1


      xx = 100 * sig / LOF(1)
      If xx = Int(xx) Then
            precentworm Picture1, xx
      End If
   Picture1.Cls

   Wend
   Text1.Text = "Beendet123"
   'se cierran los archivos usados
   Close #1
   Close #11

   Close #2
   Close #12

   Close #3
   Close #13
```

A continuación se presentan los efectos del filtro en las señales de audio capturadas.

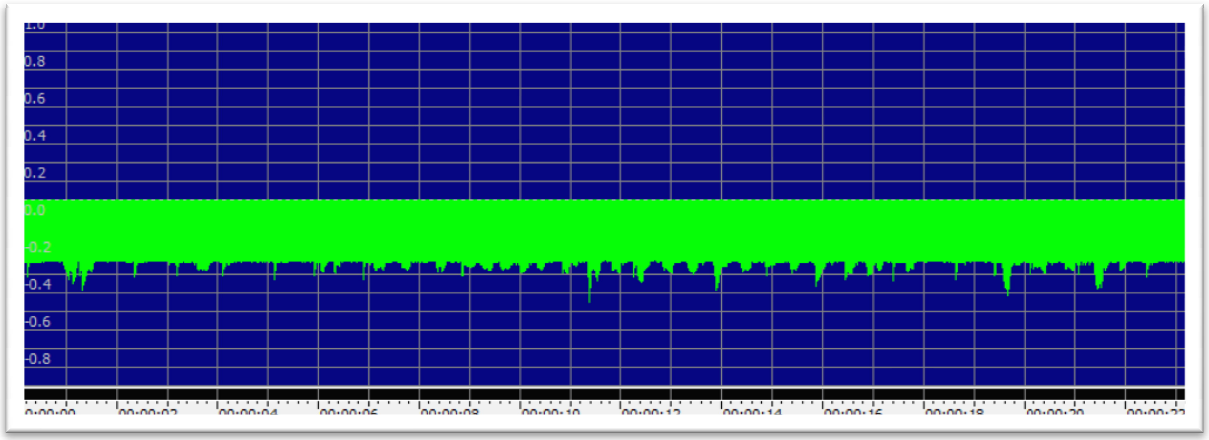


Figura 4.13. Señal 1 (señal de interés más ruido) antes de filtrar.

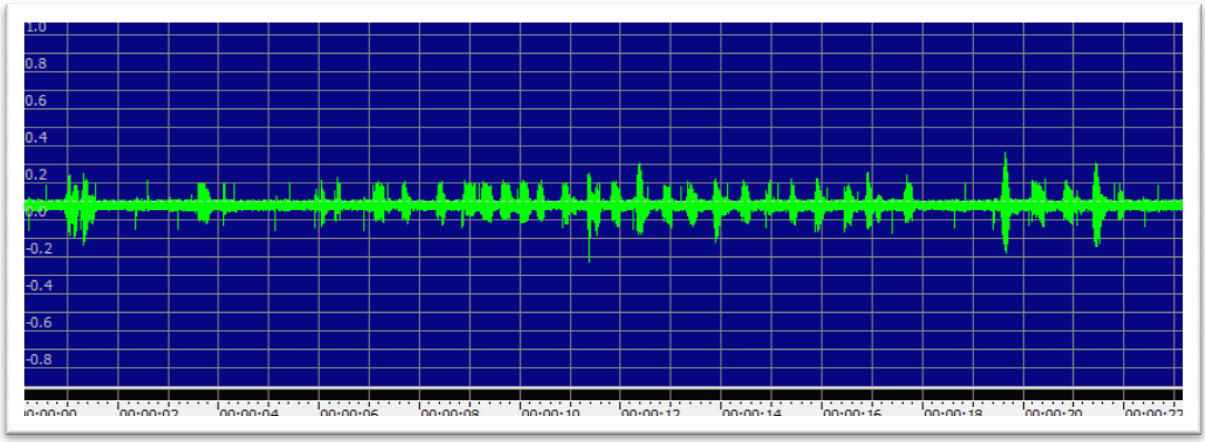


Figura 4.14. Señal 1 (señal de interés más ruido) después de filtrar.

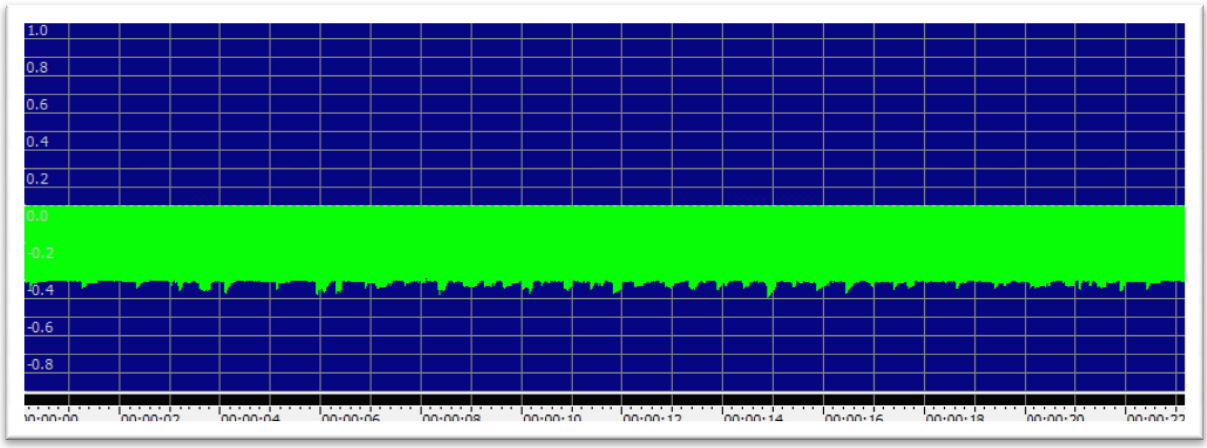


Figura 4.15. Señal 2 antes de filtrar.

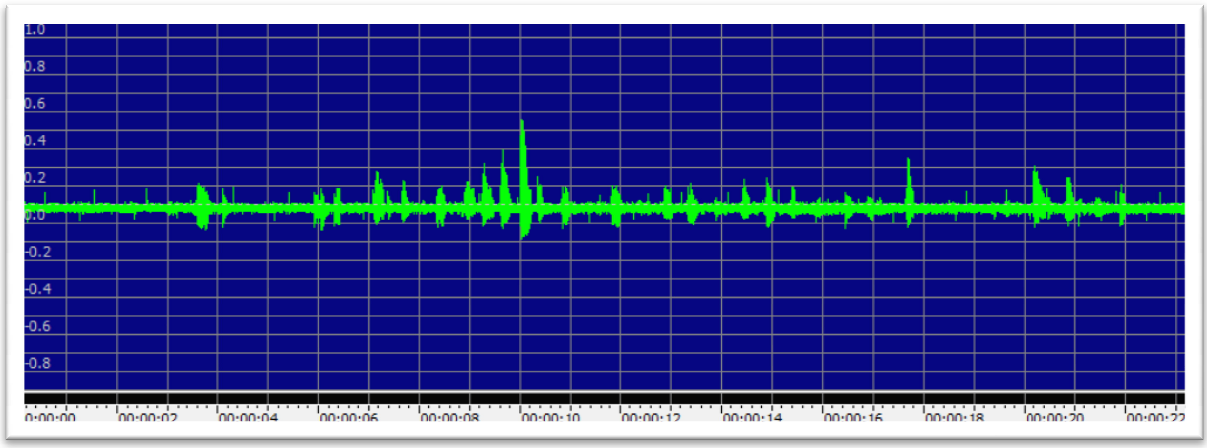


Figura 4.16. Señal 2 después de filtrar.

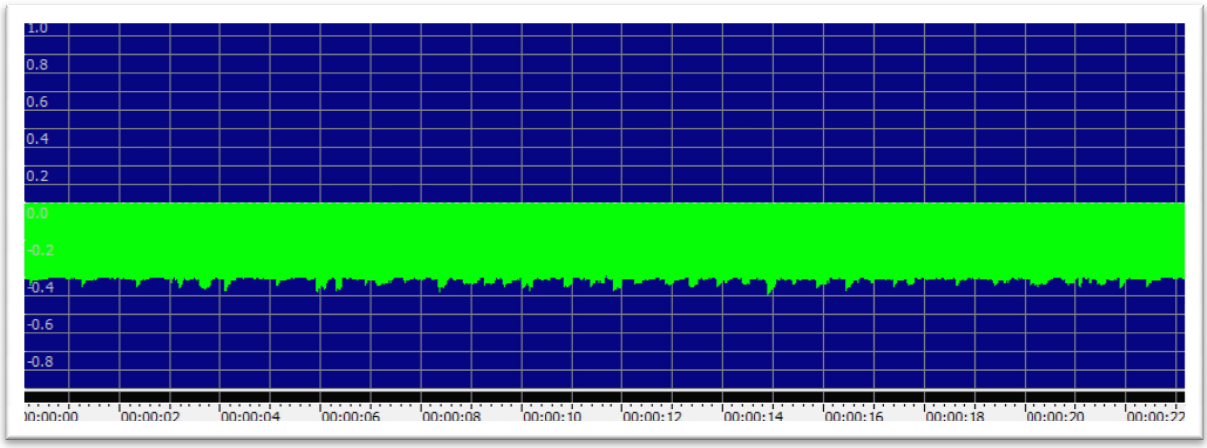


Figura 4.17. Señal 3 antes de filtrar.

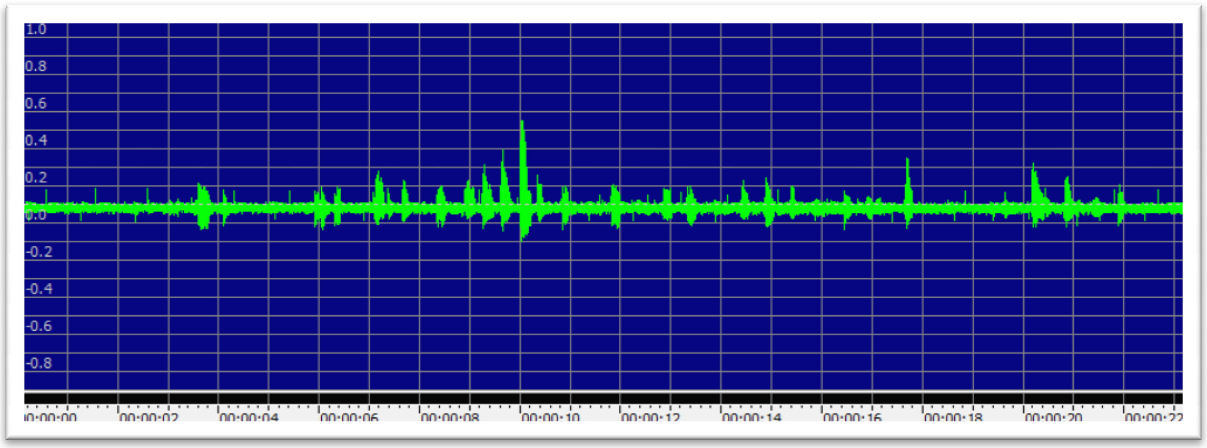


Figura 4.18. Señal 3 después de filtrar.

4.3. Cancelador adaptable de ruido

Diagrama general del cancelador adaptable de ruido.

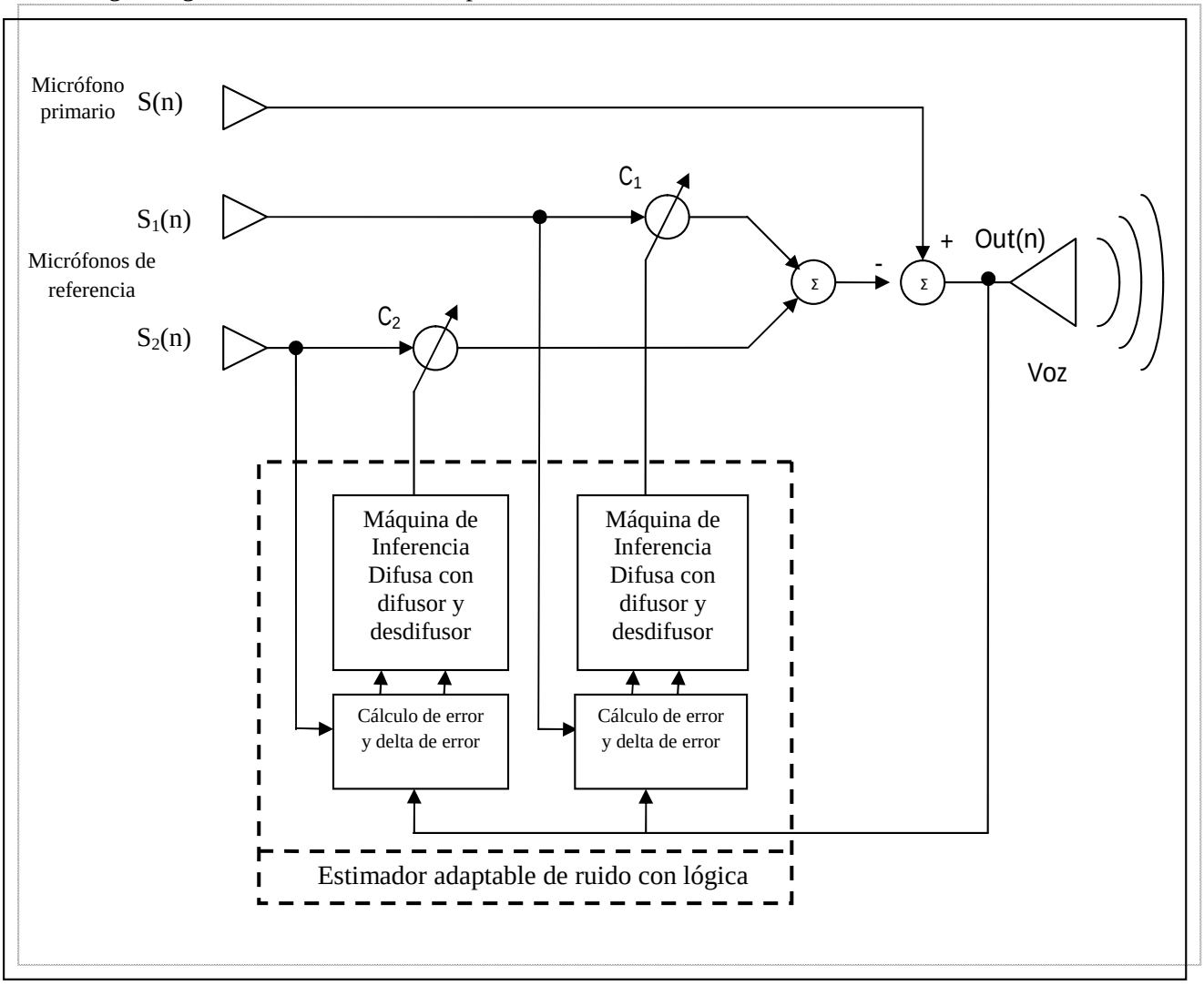


Figura 4.19. Cancelador Adaptable de ruido con lógica difusa.

4.3.1. Correlación cruzada de las señales

Como se observa en el diagrama, la salida del sistema estará dada por:

$Out(n)=S(n)-C_1S_1(n)-C_2S_2(n)$

Donde S(n) es la señal de interés mezclada con ruido aditivo del entorno.

$S_1(n)$ y $S_2(n)$ son las señales capturadas por los micrófonos de referencia, que contienen muestras del ruido del entorno del sistema.

C_1 y C_2 son los coeficientes modificados por los valores dc_1 y dc_2 (delta en coeficiente 1 y 2) que entrega el estimador adaptable de ruido con lógica difuso implementado, y estos valores a su vez se obtienen de acuerdo a las entradas de las máquinas de inferencia difusa (error y delta de error) por medio de la máquina de inferencia difusa, asimismo, se calculan cada milisegundo, lo que comprende un total de N=21 muestras de las señales de audio de los micrófonos.

$C_1=C_1+dc_1$
$C_2=C_2+dc_2$

Así cuando dc_1 y dc_2 resulten aproximadamente 0, los coeficientes C_1 y C_2 permanecerán estables en su valor, reduciendo al máximo el ruido de la salida.

El cálculo de estos coeficientes se define como el grado de parecido de la señal Out(n) con las señales de los micrófonos de referencia, para lo cual hacemos uso del concepto de correlación cruzada y ya que el parecido también depende de la amplitud, se usa el cociente de las energías de las señales como sigue:

$$e_1 = \left(\frac{energía\ out(n)}{energía\ s_1(n)}\right) \text{maxomin}\ [\rho_{(out(n)s_1(n))}(m)]$$

$$donde\ \rho_{(out(n)s_1(n))}(m) = \frac{\sum_{n=0}^{N-1} out(n)s_1(n-m)}{[\sum_{n=0}^{N-1} out^2\ (0) \sum_{n=0}^{N-1} {s_1}^2\ (0)]^{\frac{1}{2}}}$$

$$e_2 = \left(\frac{energía\ out(n)}{energía\ s_2(n)}\right) \text{maxomin}\ [\rho_{(out(n)s_2(n))}(m)]$$

$$donde\ \rho_{(out(n)s_2(n))}(m) = \frac{\sum_{n=0}^{N-1} out(n)s_2(n-m)}{[\sum_{n=0}^{N-1} out^2\ (0) \sum_{n=0}^{N-1} {s_2}^2\ (0)]^{\frac{1}{2}}}$$

Se entiende entonces, que mientras mayor sea el parecido de la señal out(n) con $s_1(n)$ el resultado de e_1 será mayor, y mientras mayor sea el parecido de out(n) con $s_2(n)$ el valor de e_2 será mayor también. Para valores negativos de e_1 la interpretación será una relación inversa entre los valores de $s_1(n)$ y out(n), y de forma análoga con e_2, $s_2(n)$ y out(n). Mientras que cuando los valores de e_1 y e_2 sean aproximadamente cero, se entenderá que la similitud entre la señal de salida es prácticamente nula con las señales provenientes de los micrófonos de referencia, por lo que la señal de salida out(n) será prácticamente la señal de interés sin los efectos del ruido estimado del entorno, esto implica que la razón señal a ruido de la señal de salida ha aumentado, como se indica en el capítulo 2 dentro del apartado 2.3.

El cociente de energías se usa como indicador de semejanza ya que la correlación cruzada es independiente a este aspecto, pero para propósitos de restar señales parecidas, la amplitud queda expresada en la energía de las señales.

Cabe mencionar que por la definición de la correlación cruzada normalizada, los valores de estas variables quedan limitados al rango de -1 a 1, de forma similar se entiende que el cociente de energía queda comprendido en el mismo rango.

Los valores de los deltas de errores (que sirven de entrada para las máquinas de inferencia difusa) se obtienen como sigue:

$$de_1(k)=e_1(k) - e_1(k-1)$$
$$de_2(k)=e_2(k) - e_2(k-1)$$

Es decir, son las diferencias del $error_i$ actual con el $error_{i-1}$ (anterior), esta variable es proporcional a la derivada del error, por lo que su comportamiento creciente o decreciente nos permitirá saber si el error a la salida de nuestro sistema esta convergiendo al mínimo (en caso de que el valor sea negativo), o si se aleja de este valor (en caso de que el valor sea positivo), o bien si tiende a permanecer estático (en caso de que el valor sea aproximadamente cero).

Es importante destacar que de acuerdo a las definiciones de e_1, e_2, de_1 y de_2 los valores de las últimas dos variables quedan limitados al rango de -2 a 2.

La sección de código en visual basic que realiza lo antes mencionado se presenta a continuación:

```
'resta de s1-f1s2-f2s3
    'Se calcula la energía de las señales para lacorr normalizada
     N = 21
     For j = 0 To 2
        energia(j) = 0
     Next
     For j = 0 To N - 1
        energia(0) = energia(0) + 0.001 * output(j) * output(j)
        energia(1) = energia(1) + 0.001 * s2(j) * s2(j)
        energia(2) = energia(2) + 0.001 * s3(j) * s3(j)
     Next
     'corr12(0) = corr12(1) = 0 inicializacion en ceros
     For l = -10 To 10
        rho12(l) = 0
        rho13(l) = 0
     Next
     'paral<0
     For l = -10 To -1
     For N = (0) To (20 + l)
        rho12(l) = rho12(l) + 0.001 * s2(N - l) * output(N)
        rho13(l) = rho13(l) + 0.001 * s3(N - l) * output(N)
     Next
     Next
     'para l>0
     For l = 0 To 10
     For N = l To (20)
        rho12(l) = rho12(l) + 0.001 * s2(N - l) * output(N)
        rho13(l) = rho13(l) + 0.001 * s3(N - l) * output(N)
     Next
     Next
     For j = -10 To 10 'Normalizacion
        rho12(j) = rho12(j) / Sqr(energia(0) * energia(1))
        rho13(j) = rho13(j) / Sqr(energia(0) * energia(2))
     Next
     'obtencion del valor máximo o minimo de los elementos de la correlacion
     For j = 0 To 2
       errmax(j) = 0
       idem(j) = 0
     Next
     For j = -10 To 10
      If Abs(rho12(j)) > errmax(0) Then
        errmax(0) = Abs(rho12(j))
        idem(0) = j
      End If
      If Abs(rho13(j)) > errmax(1) Then
        errmax(1) = Abs(rho13(j))
        idem(1) = j
       End If
     Next
     errmax(0) = rho12(idem(0)) * energia(0) / energia(1)
     errmax(1) = rho13(idem(1)) * energia(0) / energia(2)
```

4.4. Definición de la máquina de inferencia difusa con difusor y desdifusor

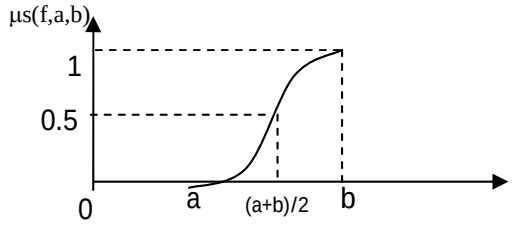
4.4.1. Máquina de inferencia difusa

Debido a que el proceso de las dos máquinas de inferencia difusa descritas en el apartado anterior tienen el mismo objetivo de obtener el valor de los deltas de los coeficientes dc_1 y dc_2, se usara una única máquina de inferencia difusa con difusor y desdifusor, con la diferencia de que dependiendo los valores de entrada, si corresponden a la señal $s_1(n)$ se obtendrá el valor de dc_1, o si corresponden a la señal $s_2(n)$ se obtendrá el valor de dc_2. No se usa una máquina de inferencia difusa con 4 entradas (e_1, e_2, de_1 ,de_2) y 2 salidas (dc_1 y dc_2) ya que se considera que las señales $s_1(n)$ y $s_2(n)$ son independientes, por lo que el coeficiente dc_1 no dependerá de los valores e_2 y de_2 ni viceversa.

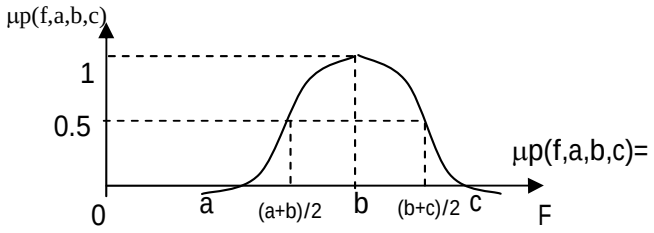
4.4.2. Difusor

Como se mencionó en el capítulo 3, existen distintas formas base para definir los conjuntos difusos de entrada, sin embargo para medios ruidosos, tal como es el caso del presente proyecto, las curvas suaves proporcionan un desempeño óptimo. Por lo que se decidió usar las siguientes formas para los conjuntos difusos

Con base a las funciones de pertenencia



$$\mu s(f,a,b)=\begin{cases}0, & f < a \\ 2\left(\dfrac{f-a}{b-a}\right)^2, & a \le f \le \dfrac{a+b}{2} \\ 1-2\left(\dfrac{f-b}{b-a}\right)^2, & \dfrac{a+b}{2} < f < b \\ 1, & f > b\end{cases}$$



$$\mu p(f,a,b,c)=\begin{cases}0, & f \le a \\ 2\left(\dfrac{f-a}{b-a}\right)^2, & a < f \le \dfrac{a+b}{2} \\ 1-2\left(\dfrac{f-b}{b-a}\right)^2, & \dfrac{a+b}{2} < f \le b \\ 1-2\left(\dfrac{f-b}{b-c}\right)^2, & b < f \le \dfrac{b+c}{2} \\ 2\left(\dfrac{f-c}{b-c}\right), & \dfrac{b+c}{2} < f \le c \\ 0, & f > c\end{cases}$$

Es indispensable definir en primer lugar el Universo del Discurso para las dos variables de entrada que maneja el difusor, es decir el error y el delta de error, por lo que recurrimos al rango de valores que limita la definición de cada una de estas variables:

Error: $e \in [-1,1]$
Delta de error: $de \in [-2,2]$

Una vez definido el Universo del Discurso, se definen los conjuntos difusos de entrada para cada variable, que son los distintos conjuntos en los que se representará la información de las entradas de error y delta de error, se notará que las siguientes definiciones nos proporcionan una mayor sensibilidad para valores cercanos a cero que para los valores más alejados

El código de visual basic para este tipo de funciones se enlista a continuación:
Para función tipo "S"

```
Public Function mus(err As Double, lima As Double, limb As Double) As Double
   Dim mimu As Double


   If lima < limb Then
      If err >= limb Then
         mimu = 1
      End If
      If err <= lima Then
         mimu = 0
      End If
      If (err <= ((lima + limb) / 2)) And (err > lima) Then
         mimu = 2 * ((err - lima) / (limb - lima)) ^ 2
      End If
      If (err < limb) And (err > ((lima + limb) / 2)) Then
         mimu = 1 - 2 * ((err - limb) / (limb - lima)) ^ 2
      End If
   End If
   If lima > limb Then
      If err <= limb Then
         mimu = 1
      End If
      If err >= lima Then
         mimu = 0
      End If
      If (err >= ((lima + limb) / 2)) And (err < lima) Then
         mimu = 2 * (((err - lima) / (limb - lima)) ^ 2)
      End If
      If (err > limb) And (err < ((lima + limb) / 2)) Then
         mimu = 1 - 2 * (((err - limb) / (limb - lima)) ^ 2)
      End If
```

```
    End If
    mus = mimu
    Exit Function
End Function
```

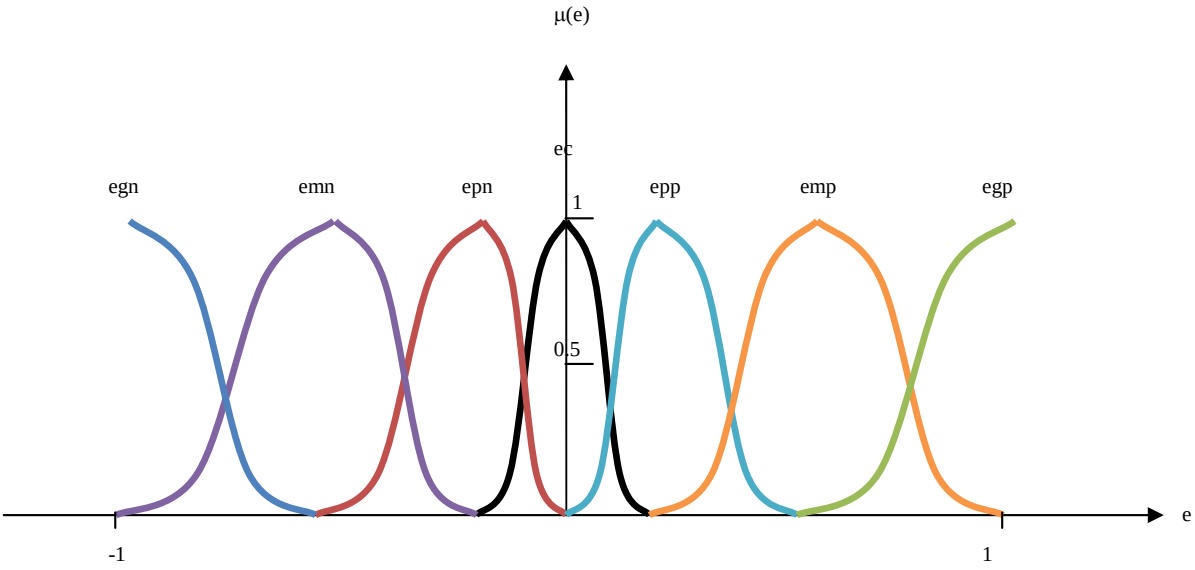
Para función tipo “pi”

```
Public Function mup(err As Double, lima1 As Double, limb As
Double, lima2) As Double
  Dim mimu As Double

    If (err >= lima2) And (err < lima1) Then
       mimu = 0
    End If
    If (err >= lima1) And (err < ((lima1 + limb) / 2)) Then
       mimu = 2 * ((err - lima1) / (limb - lima1)) ^ 2
    End If
    If (err >= ((lima1 + limb) / 2)) And (err < limb) Then
       mimu = 1 - 2 * ((err - limb) / (limb - lima1)) ^ 2
    End If
    If (err >= limb) And (err < ((lima2 + limb) / 2)) Then
       mimu = 1 - 2 * ((err - limb) / (limb - lima2)) ^ 2
    End If
    If (err >= ((lima2 + limb) / 2)) And (err <= lima2) Then
       mimu = 2 * ((err - lima2) / (limb - lima2)) ^ 2
    End If

  mup = mimu
  Exit Function
End Function
```

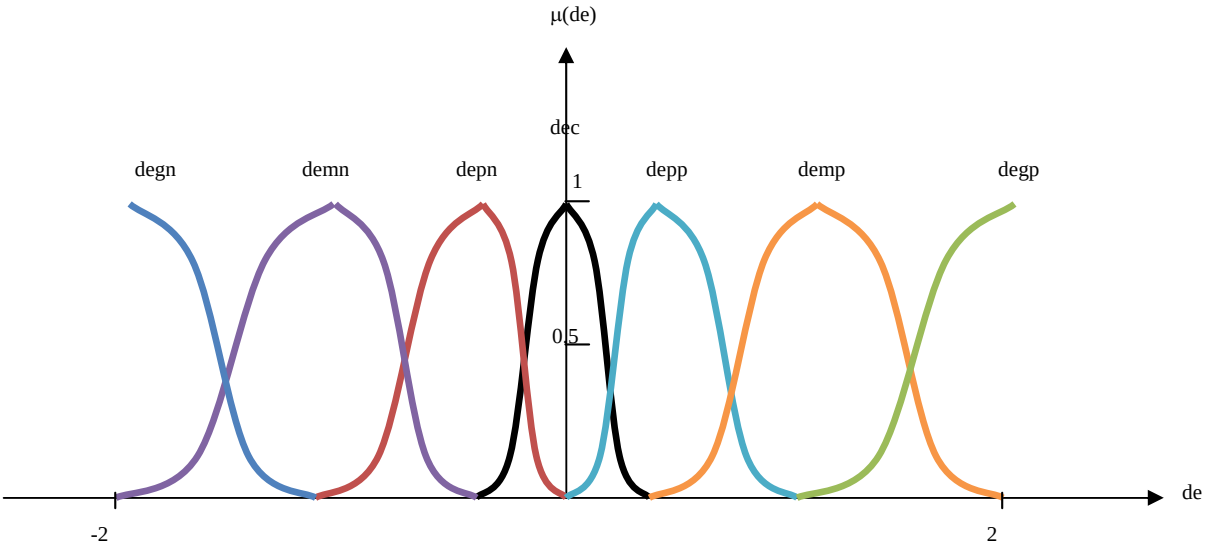
Conjuntos difusos de entrada para el Error.



Funciones de pertenencia para los conjuntos difusos de entrada de Error.

Error grande negativo:	μegn=μs(e,-0.5,-1)
Error mediano negativo:	μemn=μp(e,-1,-0.5,-0.2)
Error pequeño negativo:	μepn=μp(e,-0.5,-0.2,0)
Error cero:	μec=μp(e,-0.2,0,0.2)
Error pequeño positivo:	μepp=μp(e,0,0.2,0.5)
Error mediano positivo:	μemp=μp(e,0.2,0.5,1)
Error grande positivo:	μegp=μs(e,0.5,1)

Conjuntos difusos de entrada para Delta de error.



Funciones de pertenencia para los conjuntos difusos de entrada de Delta de error.

Delta Error grande negativo: μdegn=μs(e,-1,-2)
Delta Error mediano negativo: μdemn=μp(e,-2,-1,-0.4)
Delta Error pequeño negativo: μdepn=μp(e,-1,-0.4,0)
Delta Error cero: μdec=μp(e,-0.4,0,0.4)
Delta Error pequeño positivo: μdepp=μp(e,0,0.4,1)
Delta Error mediano positivo: μdemp=μp(e,0.4,1,2)
Delta Error grande positivo: μdegp=μs(e,1,2)

4.4.3. Desdifusor

Debido a que la mejor aproximación para obtener una salida definida a partir de conjuntos difusos de salida definidos por la base de reglas difusas es el centroide, se decidió implementar una versión discreta del centroide sin importar el tiempo de procesamiento, ya que no se maneja tiempo real en el proyecto, y así obtener un coeficiente correspondiente a las reglas propuestas.

Método de desdifusión, aproximación de centroide

$$Z^* = \frac{\int \mu(z) z dz}{\int \mu(z) dz} \approx \frac{\sum \mu(z_i) z_i \Delta z}{\sum \mu(z_i) \Delta z}$$

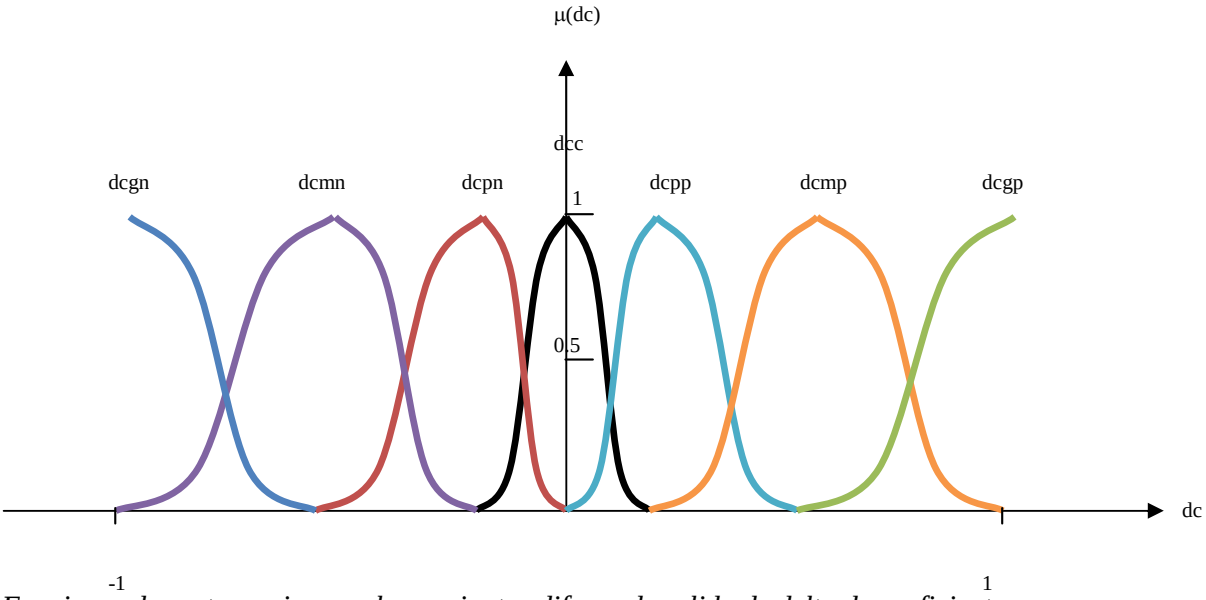
Sabiendo que mientras menor sea Δz, la aproximación será mejor, aunque el proceso será más tardado.

El Universo del discurso para la variable de salida (delta de coeficiente) se define del rango de -1 a 1, con la teoría de la correlación cruzada, ya que la señal de salida no puede contener componentes amplificadas de las señales $s_1(n)$ ni $s_2(n)$, y con estos valores se pueden eliminar efectos directos e inversos de las señales de los micrófonos de referencia en la salida del sistema.

Delta de coeficiente: dc∈[-1,1]

Bajo las mismas funciones descritas para el difusor para las funciones de pertenencia de los conjuntos difusos, se definen los siguientes conjuntos difusos de salida, a los que se les aplicará al final la desdifusión.

Conjuntos difusos de salida para delta de coeficiente.



Funciones de pertenencia para los conjuntos difusos de salida de delta de coeficiente.

Delta coeficiente grande negativo:	μdcgn=μs(e,-0.5,-1)
Delta coeficiente mediano negativo:	μdcmn=μp(e,-1,-0.5,-0.2)
Delta coeficiente pequeño negativo:	μdcpn=μp(e,-0.5,-0.2,0)
Delta coeficiente cero:	μdcc=μp(e,-0.2,0,0.2)
Delta coeficiente pequeño positivo:	μdcpp=μp(e,0,0.2,0.5)
Delta coeficiente mediano positivo:	μdcmp=μp(e,0.2,0.5,1)
Delta coeficiente grande positivo:	μdcgp=μs(e,0.5,1)

El código en visual basic que realiza la desdifusión y considerando los conjuntos difusos de salida enlistados anteriormente se presenta a continuación:

```
mudelta = 0
num = 0
den = 0
zcaract = 0
delta = 1 / 1001
For i = 0 To ((2 / delta) - 2)
   mudelta = 0
   zeta = -1 + delta * ((i + 1))
   'se calcula el valor de mu(z)
   If mufact(0) <> 0 Then
      mudelta = max((mufact(0) * mus(zeta, -0.5, -1)), mudelta)
   End If
   If mufact(1) <> 0 Then
      mudelta = max((mufact(1) * mup(zeta, -1, -0.5, -0.21)),
mudelta)
   End If
   If mufact(2) <> 0 Then
      mudelta = max((mufact(2) * mup(zeta, -0.5, -0.2, 0)),
mudelta)
   End If
   If mufact(3) <> 0 Then
      mudelta = max((mufact(3) * mup(zeta, -0.2, 0, 0.2)),
mudelta)
   End If
   If mufact(4) <> 0 Then
      mudelta = max((mufact(4) * mup(zeta, 0, 0.2, 0.5)),
mudelta)
   End If
   If mufact(5) <> 0 Then
      mudelta = max((mufact(5) * mup(zeta, 0.2, 0.5, 1)),
mudelta)
   End If
   If mufact(6) <> 0 Then
      mudelta = max((mufact(6) * mus(zeta, 0.5, 1)), mudelta)
   End If
   num = num + mudelta * zeta * 2 * delta
   den = den + mudelta * 2 * delta
Next
   zcaract = num / den
```

```
   If ((mufact(3) = 1) And (mufact(0) * mufact(1) * mufact(2) *          zcaract = 0
mufact(4) * mufact(5) * mufact(6) = 0)) Then                          End If
```

4.4.4. Base de Reglas difusas

Para establecer la base de reglas difusas, es preciso interpretar el significado tanto de las entradas de la máquina de inferencia difusa, como su salida.

Conjunto de reglas
Para definir el conjunto de reglas es necesario interpretar el significado de los conjuntos difusos de entrada y salida. Por lo que se hacen las siguientes consideraciones.

- Un valor positivo del coeficiente anula errores positivos mientras que un valor negativo de coeficiente anula errores negativos
- Es deseable un valor del error cero, ya que un error positivo o negativo quiere decir que hay parecido entre las señales ya sea por razones de las mismas señales o por sobrepaso del coeficiente
- El valor de delta de coeficiente positivo incrementa el valor del coeficiente
- Entre mayor sea el valor del delta del coeficiente, el valor del coeficiente se incrementará más rápido
- Un valor negativo de delta de coeficiente decrementa el valor del coeficiente
- Entre menor (negativo) sea el valor del delta de coeficiente, el valor del coeficiente disminuye
- Un valor de delta de coeficiente cercano a cero prácticamente no modifica el valor del coeficiente
- Un valor positivo de delta de error cuando el error es positivo involucra que el error tiende a aumentar
- Un valor negativo de delta de error cuando el error es positivo involucra que el error tiende a disminuir
- Un valor positivo de delta de error cuando el error es negativo significa que el error negativo disminuye
- Un valor negativo de delta de error cuando el error es negativo significa que el error aumenta negativamente

El comportamiento del delta de error para distintos casos se muestra en la figura 4.20.

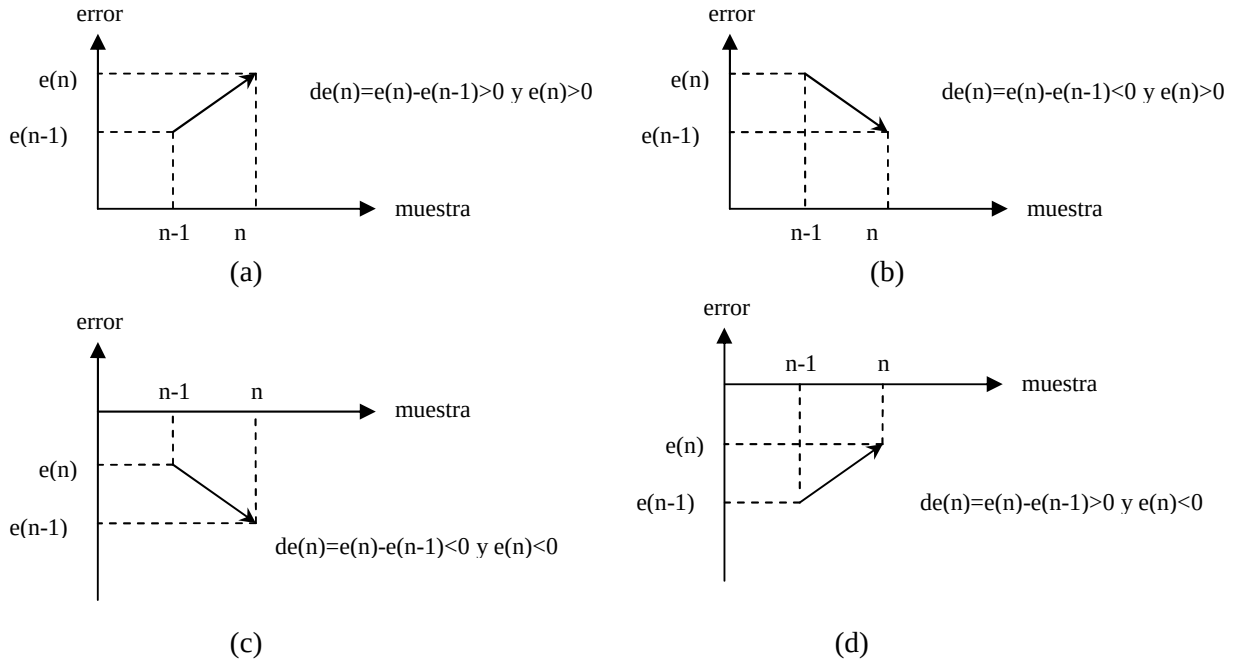


Figura 4.20. Posibles casos para el error y delta de error:
(a) si el error es positivo y delta de error es positivo, entonces el error tiende a alejarse de cero
(b) si el error es positivo y el delta de error es negativo, entonces el error tiende a cero
(c) si el error es negativo y delta de error positivo, entonces el error tiende a cero
(d) si el error es negativo y delta de error es negativo, entonces el error tiende a alejarse de cero

La tabla 4.1 muestra la implicación entre conjuntos difusos de entrada y de salida

	egn	**emn**	**epn**	**ec**	**epp**	**Emp**	**egp**
degn	dcgn	dcgn	dcmn	dcc	dcc	Dcmp	dcmp
demn	dcgn	dcgn	dcmn	dcc	dcc	Dcmp	dcgp
depn	dcgn	dcmn	dcpn	dcc	dcpp	Dcmp	dcgp
dec	dcgn	dcmn	dcpn	dcc	dcpp	Dcmp	dcgp
depp	dcgn	dcmn	dcpn	dcc	dccp	Dcmp	dcgp
demp	dcgn	dcmn	dccc	dcc	dcmp	Dcgp	dcgp
degp	dcmn	dcmn	dccc	dcc	dcmp	Dcgp	dcgp

Tabla 4.1. Conjunto de reglas para el sistema de inferencia difusa.

En forma de reglas Si… Entonces

1. Si egn y degn entonces dcgn
2. Si egn y demn entonces dcgn
3. Si egn y depn entonces dcgn
4. Si egn y dec entonces dcgn
5. Si egn y depp entonces dcgn
6. Si egn y demp entonces dcgn
7. Si egn y degp entonces dcmn
8. Si emn y degn entonces dcgn
9. Si emn y demn entonces dcgn
10. Si emn y depn entonces dcmn
11. Si emn y dec entonces dcmn
12. Si emn y depp entonces dcmn
13. Si emn y demp entonces dcmn
14. Si emn y degp entonces dcmn
15. Si epn y degn entonces dcmn
16. Si epn y demn entonces dcmn
17. Si epn y depn entonces dcmn
18. Si epn y dec entonces dcpn
19. Si epn y depp entonces dcpn
20. Si epn y demp entonces dcc
21. Si epn y degp entonces dcc
22. Si ec y degn entonces dcc
23. Si ec y demn entonces dcc
24. Si ec y depn entonces dcc
25. Si ec y dec entonces dcc
26. Si ec y depp entonces dcc
27. Si ec y demp entonces dcc
28. Si ec y degp entonces dcc
29. Si epp y degn entonces dcc
30. Si epp y demn entonces dcc
31. Si epp y depn entonces dcpp
32. Si epp y dec entonces dcpp
33. Si epp y depp entonces dcpp
34. Si epp y demp entonces dcmp
35. Si epp y degp entonces dcmp
36. Si emp y degn entonces dcmp
37. Si emp y demn entonces dcmp
38. Si emp y depn entonces dcmp
39. Si emp y dec entonces dcmp
40. Si emp y depp entonces dcmp
41. Si emp y demp entonces dcgp
42. Si emp y degp entonces dcgp
43. Si egp y degn entonces dcmp
44. Si egp y demn entonces dcgp

45. Si egp y depn entonces dcgp
46. Si egp y dec entonces dcgp
47. Si egp y depp entonces dcgp
48. Si egp y demp entonces dcgp
49. Si egp y degp entonces dc

En código de visual basic:

```
   'Inicializacion de salida en ceros
For i = 0 To 6
   mufact(i) = 0
Next

mufact(0) = consec(muerr(0), mupunto(0), mufact(0)) 'regla1
mufact(0) = consec(muerr(0), mupunto(1), mufact(0)) 'regla2
mufact(0) = consec(muerr(0), mupunto(2), mufact(0)) 'regla3
mufact(0) = consec(muerr(0), mupunto(3), mufact(0)) 'regla4
mufact(0) = consec(muerr(0), mupunto(4), mufact(0)) 'regla5
mufact(0) = consec(muerr(0), mupunto(5), mufact(0)) 'regla6
mufact(1) = consec(muerr(0), mupunto(6), mufact(1)) 'regla7
mufact(0) = consec(muerr(1), mupunto(0), mufact(0)) 'regla8
mufact(0) = consec(muerr(1), mupunto(1), mufact(0)) 'regla9
mufact(1) = consec(muerr(1), mupunto(2), mufact(1)) 'regla10
mufact(1) = consec(muerr(1), mupunto(3), mufact(1)) 'regla11
mufact(1) = consec(muerr(1), mupunto(4), mufact(1)) 'regla12
mufact(1) = consec(muerr(1), mupunto(5), mufact(1)) 'regla13
mufact(1) = consec(muerr(1), mupunto(6), mufact(1)) 'regla14
mufact(1) = consec(muerr(2), mupunto(0), mufact(1)) 'regla15
mufact(1) = consec(muerr(2), mupunto(1), mufact(1)) 'regla16
mufact(2) = consec(muerr(2), mupunto(2), mufact(2)) 'regla17
mufact(2) = consec(muerr(2), mupunto(3), mufact(2)) 'regla18
mufact(2) = consec(muerr(2), mupunto(4), mufact(2)) 'regla19
mufact(3) = consec(muerr(2), mupunto(5), mufact(3)) 'regla20
mufact(3) = consec(muerr(2), mupunto(6), mufact(3)) 'regla21
mufact(3) = consec(muerr(3), mupunto(0), mufact(3)) 'regla22
mufact(3) = consec(muerr(3), mupunto(1), mufact(3)) 'regla23
mufact(3) = consec(muerr(3), mupunto(2), mufact(3)) 'regla24
mufact(3) = consec(muerr(3), mupunto(3), mufact(3)) 'regla25
mufact(3) = consec(muerr(3), mupunto(4), mufact(3)) 'regla26
mufact(3) = consec(muerr(3), mupunto(5), mufact(3)) 'regla27
mufact(3) = consec(muerr(3), mupunto(6), mufact(3)) 'regla28
mufact(3) = consec(muerr(4), mupunto(0), mufact(3)) 'regla29
mufact(3) = consec(muerr(4), mupunto(1), mufact(3)) 'regla30
mufact(4) = consec(muerr(4), mupunto(2), mufact(4)) 'regla31
mufact(4) = consec(muerr(4), mupunto(3), mufact(4)) 'regla32
mufact(4) = consec(muerr(4), mupunto(4), mufact(4)) 'regla33
mufact(5) = consec(muerr(4), mupunto(5), mufact(5)) 'regla34
mufact(5) = consec(muerr(4), mupunto(6), mufact(5)) 'regla35
mufact(5) = consec(muerr(5), mupunto(0), mufact(5)) 'regla36
mufact(5) = consec(muerr(5), mupunto(1), mufact(5)) 'regla37
mufact(5) = consec(muerr(5), mupunto(2), mufact(5)) 'regla38
mufact(5) = consec(muerr(5), mupunto(3), mufact(5)) 'regla39
mufact(5) = consec(muerr(5), mupunto(4), mufact(5)) 'regla40
mufact(6) = consec(muerr(5), mupunto(5), mufact(6)) 'regla41
mufact(6) = consec(muerr(5), mupunto(6), mufact(6)) 'regla42
mufact(5) = consec(muerr(6), mupunto(0), mufact(5)) 'regla43
mufact(6) = consec(muerr(6), mupunto(1), mufact(6)) 'regla44
mufact(6) = consec(muerr(6), mupunto(2), mufact(6)) 'regla45
mufact(6) = consec(muerr(6), mupunto(3), mufact(6)) 'regla46
mufact(6) = consec(muerr(6), mupunto(4), mufact(6)) 'regla47
mufact(6) = consec(muerr(6), mupunto(5), mufact(6)) 'regla48
mufact(6) = consec(muerr(6), mupunto(6), mufact(6)) 'regla49
'dónde la función consec se define como
Public Function consec(prec1 As Double, prec2 As Doubl
e, valini As Double) As Double
   consec = min(prec1, prec2)
   consec = max(consec, valini)
   Exit Function
End Functio
```

Los métodos usados para la sección de lógica difusa son:
Mínimo como método AND
Máximo como método OR
Producto como método de Implicación
Máximo como método de Agregación
Ya que estos métodos son fácilmente implementables.

4.4.5. Máquina de inferencia difusa en MATLAB

Bajo las especificaciones anteriores, se realizó una simulación en MATLAB del funcionamiento de la máquina de inferencia difusa con difusor y desdifusor del proyecto.

Como se aprecia en la figura 4.21, se nota que el sistema cuenta con dos entradas (error y delta de error) y una salida (delta de coeficiente) teniendo por centro una base de reglas difusas.

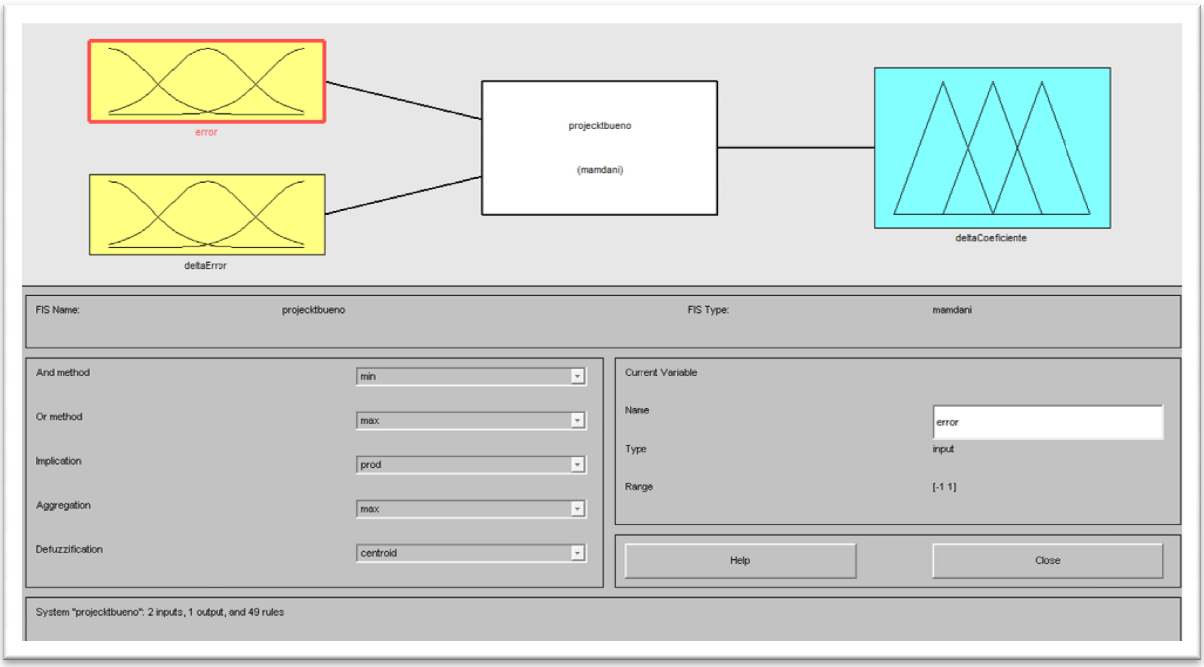


Figura 4.21. Configuración del sistema de inferencia difusa con difusor y desdifusor.

Como ya se detalló anteriormente, la forma de los conjuntos difusos de entrada y de salida son curvas suaves, ya que presentan un desempeño óptimo en sistemas ruidosos. Las figuras 4.22, 4.23 y 4.24 muestran la forma y distribución de los conjuntos difusos de entrada y salida.

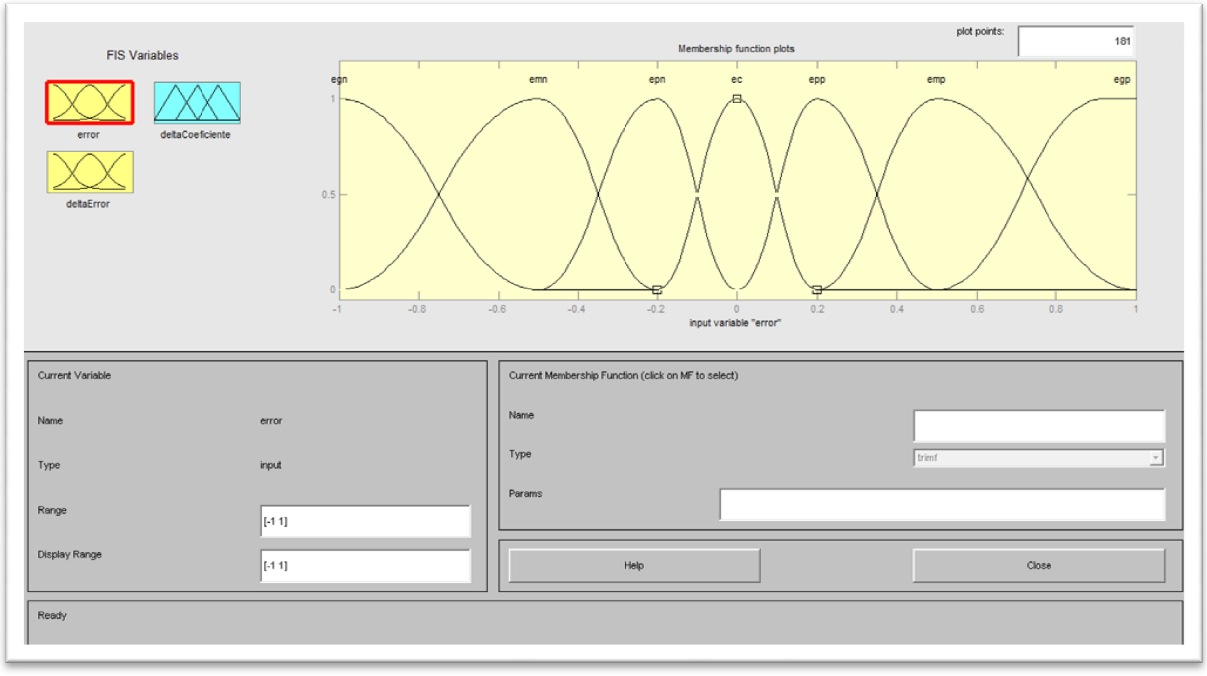


Figura 4.22.Conjuntos difusos para la variable de entrada “error”.

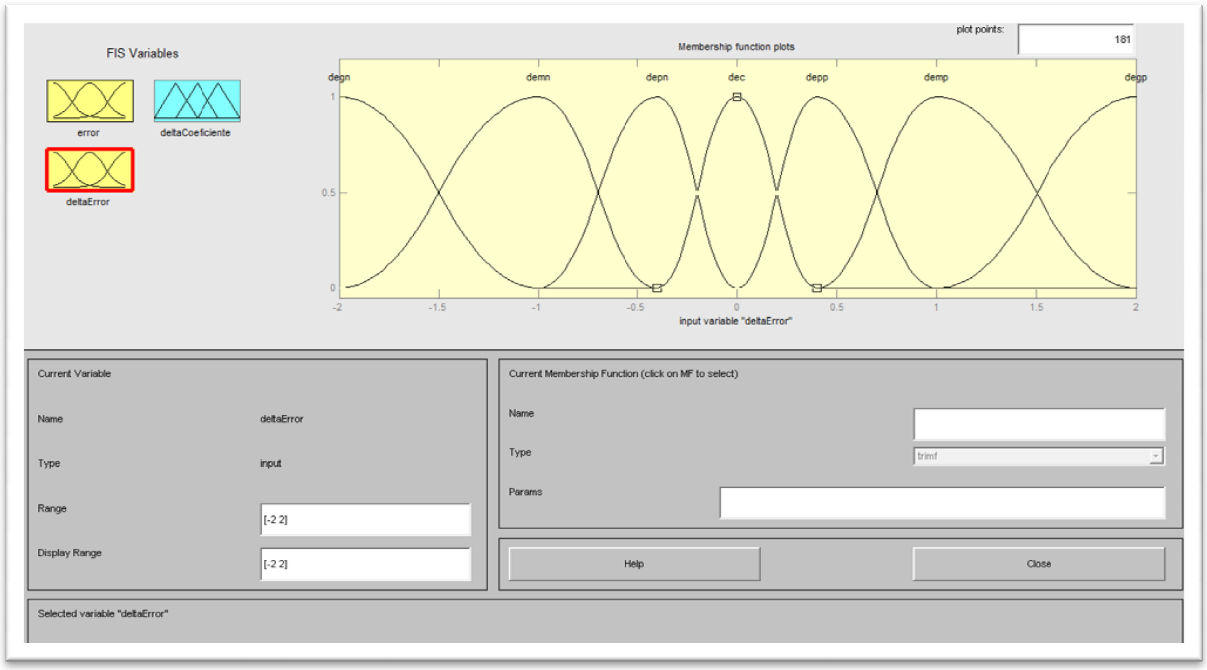


Figura 4.23. Conjuntos difusos para la variable de entrada “delta de error”.

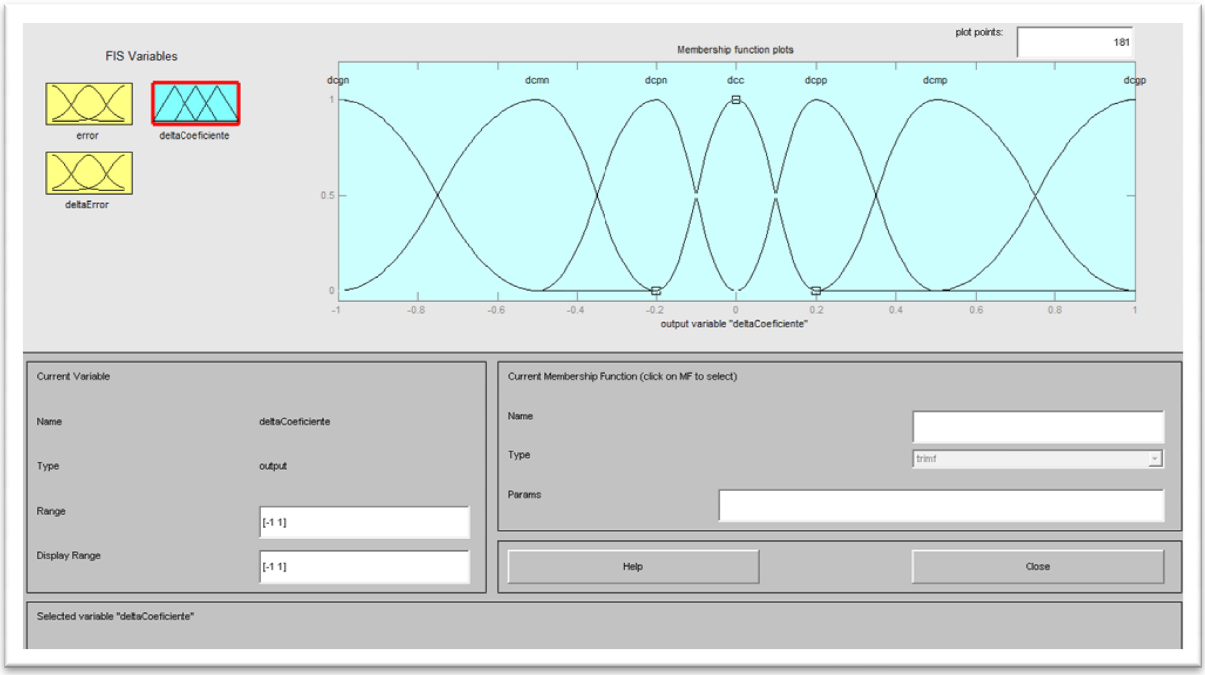


Figura 4.24. Conjuntos difusos para la variable de salida “delta de coeficiente”.

El conjunto de reglas difusas descrito para la simulación de MATLAB, que incluye las reglas especificadas anteriormente, se muestran en la figura 4.25.

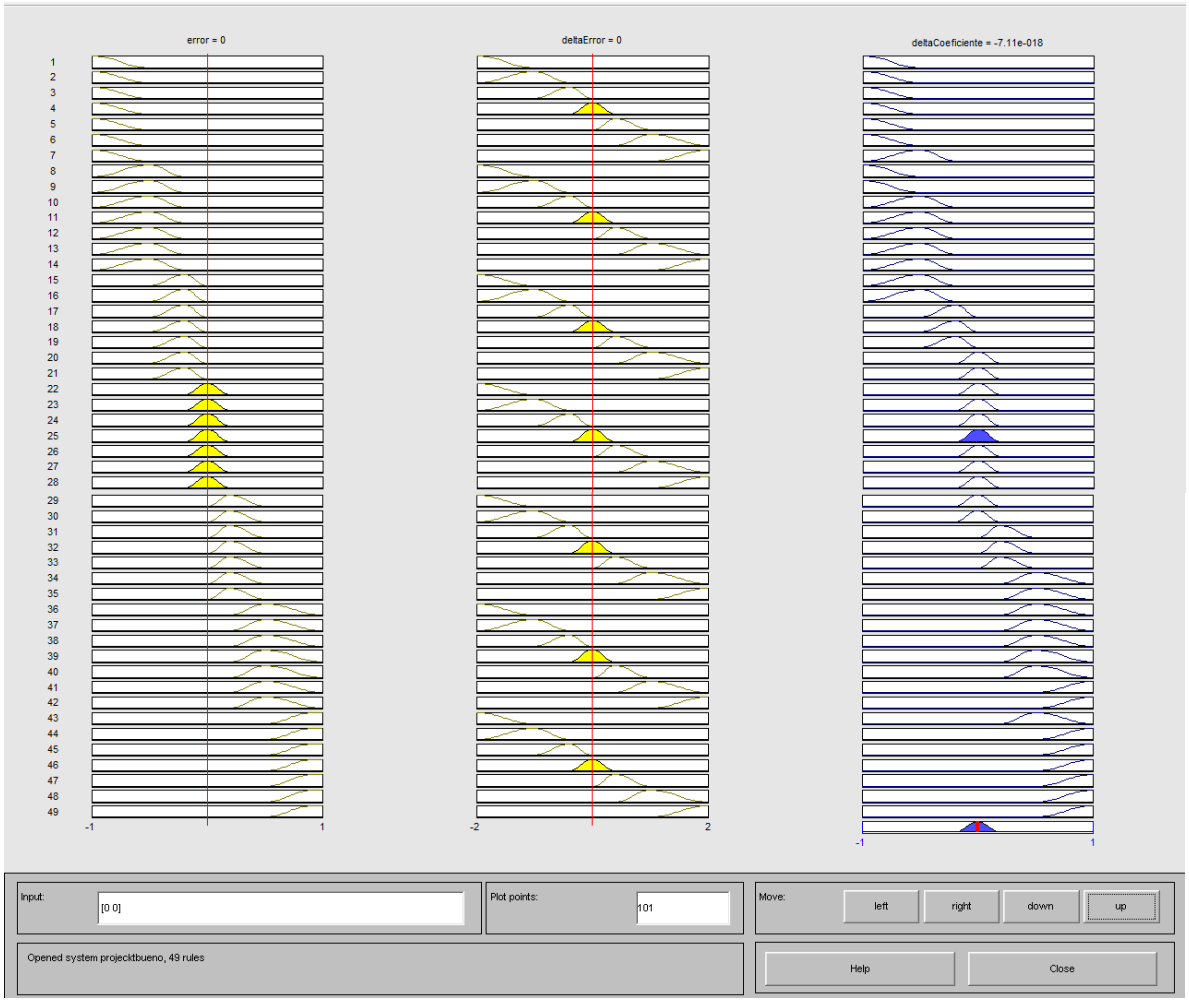


Figura 4.25. Conjunto de reglas del sistema de inferencia difusa.

La superficie de control, que se genera con los conjuntos difusos y las reglas especificadas, se muestra en la figura 4.26 y es la relación existente entre las dos variables de entrada y la salida del sistema de inferencia difusa con difusor y desdifusor.

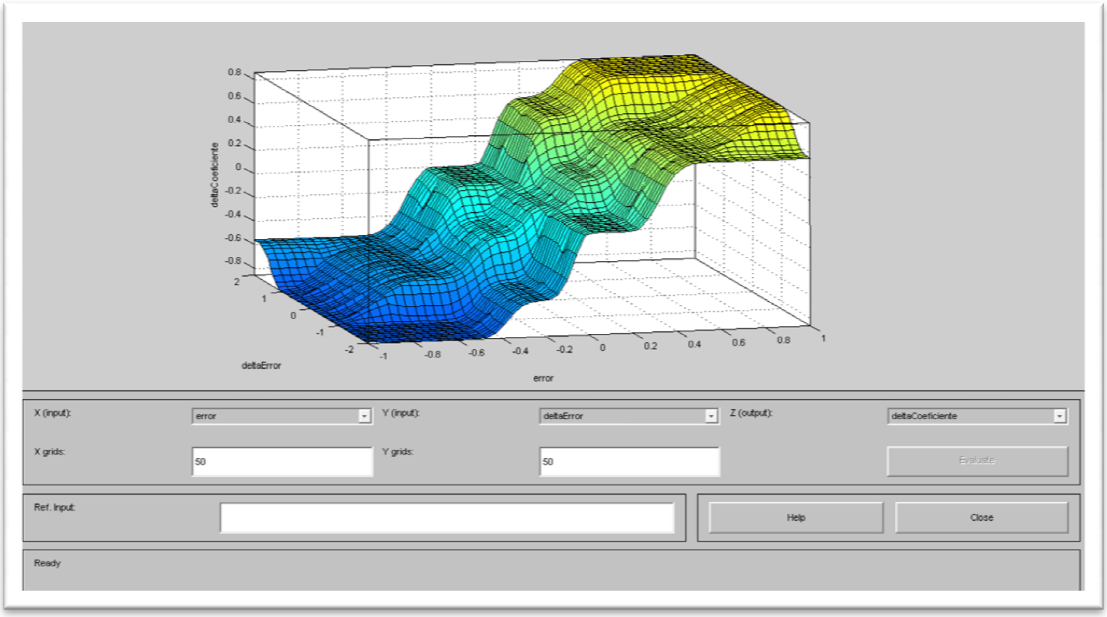


Figura 4.26. Superficie de control para el sistema de inferencia difusa.

La figura 4.27 presenta la traza de la superficie de control para el caso de delta de error igual a cero, por lo que muestra el comportamiento del delta del coeficiente con respecto al error.

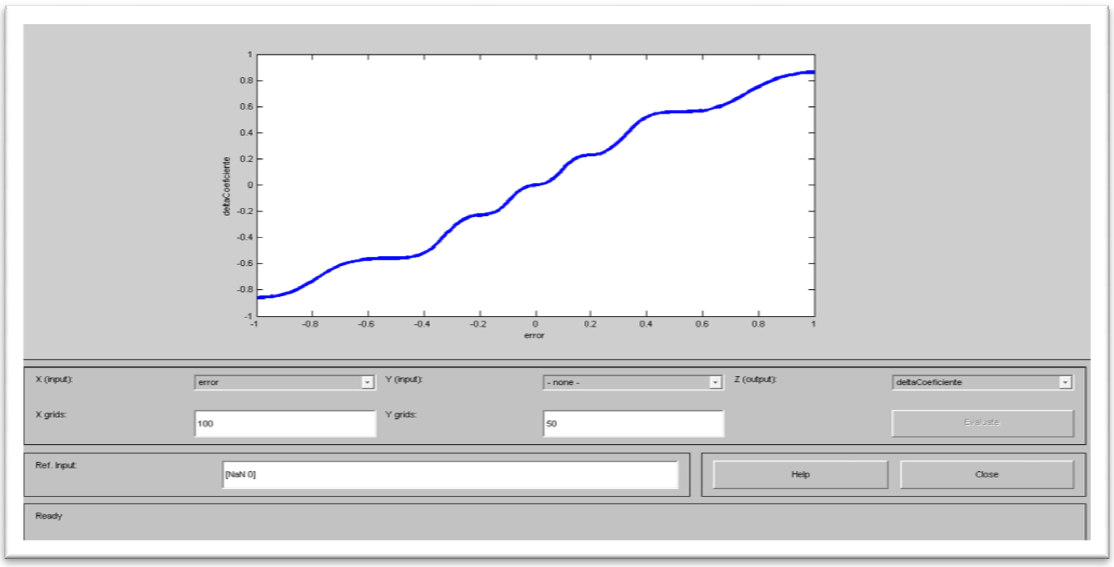


Figura 4.27. Traza de la superficie de control para delta de error cero.

Capítulo 5

Pruebas

En este capítulo se presentan los resultados obtenidos en diversas pruebas realizadas con el sistema implementado, tanto con señales de prueba creadas digitalmente, como con señales adquiridas con la interfaz USB implementada. Se comentará cada ejemplo y la relevancia del mismo para asimilar el comportamiento del cancelador de ruido adaptable con lógica difusa.

Las gráficas de las señales serán mostradas por medio del software GOLDWAVE, que es un editor de onda que puede abrir los archivos generados por el programa realizado en Visual Basic.

5.1. Prueba 1

Esta prueba fue realizada completamente usando señales capturadas por medio de la interfaz elaborada, sin embargo existe el detalle que los micrófonos de referencia no se encuentran bien ubicados para captar ruido del entorno, ni lo suficientemente espaciados del micrófono principal, por lo que parte de la señal de interés es también capturada por los micrófonos de referencia.

La ventana inicial del programa Visual Basic se presenta en la figura 5.1, donde se observa las opciones para capturar los datos que recibe en el puerto USB, desconectar el puerto USB y guardar los datos ya recibidos, inicio del proceso de filtrado y, dos opciones más para observar los módulos de correlación y lógica difusa de forma independiente del proceso conjunto. La figura 5.2 muestra una captura de pantalla del proceso de filtrado en proceso, primero se elimina la componente de directa de los archivos guardados y/o después se realiza el proceso de filtrado adaptable con lógica difusa.



Figura 5.1. Captura de pantalla inicial del programa realizado en VISUAL BASIC.

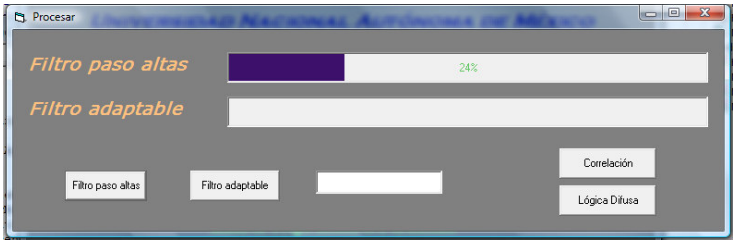


Figura 5.2. Captura de pantalla del programa procesando las señales.

Las figuras 5.3, 5.4 y 5.5 presentan las señales capturadas antes de ser filtradas con el filtro paso altas butterworth de 4º orden.

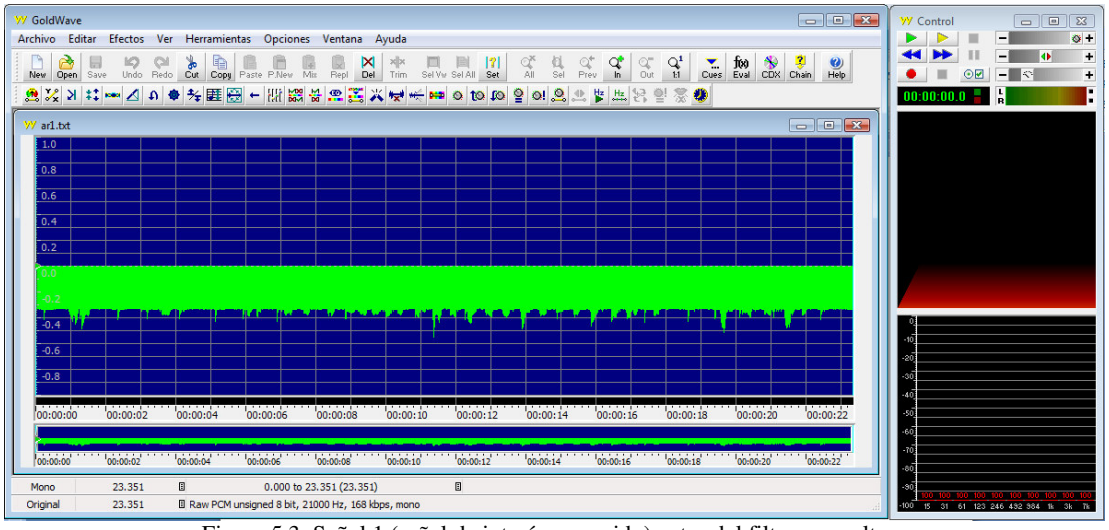


Figura 5.3. Señal 1 (señal de interés con ruido) antes del filtro paso altas.

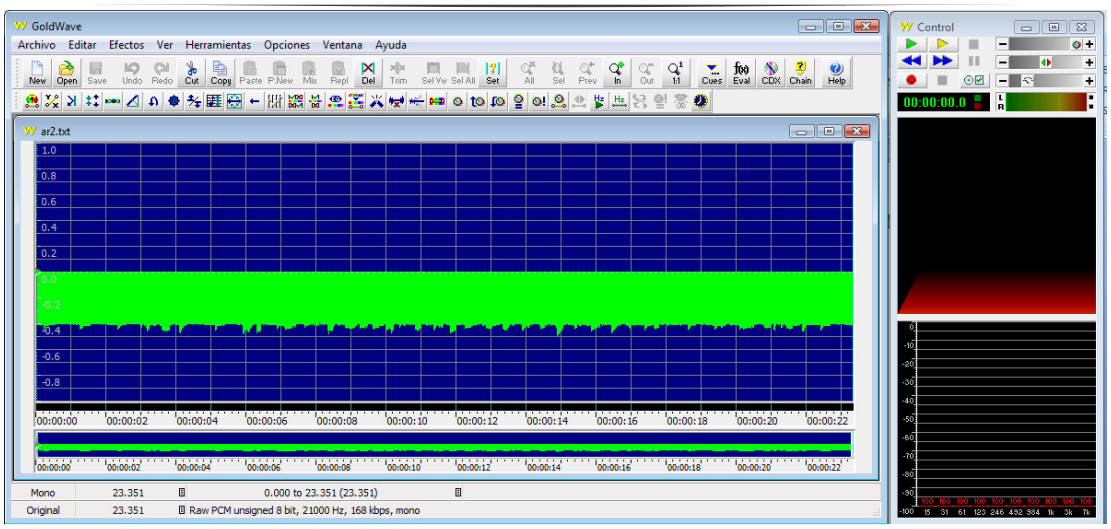


Figura 5.4. Señal 2 (señal de estimación de ruido) antes del filtro paso altas.

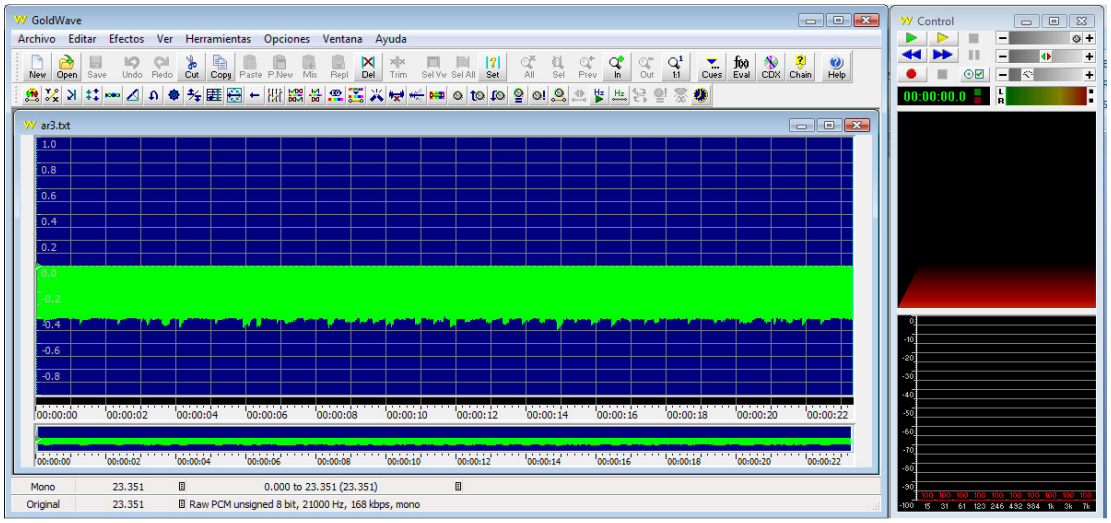


Figura 5.5. Señal 3 (señal de estimación de ruido) antes del filtro paso altas.

A continuación se presentan las mismas señales a la salida del filtro paso altas antes mencionado en las figuras 5.6, 5.7 y 5.8.

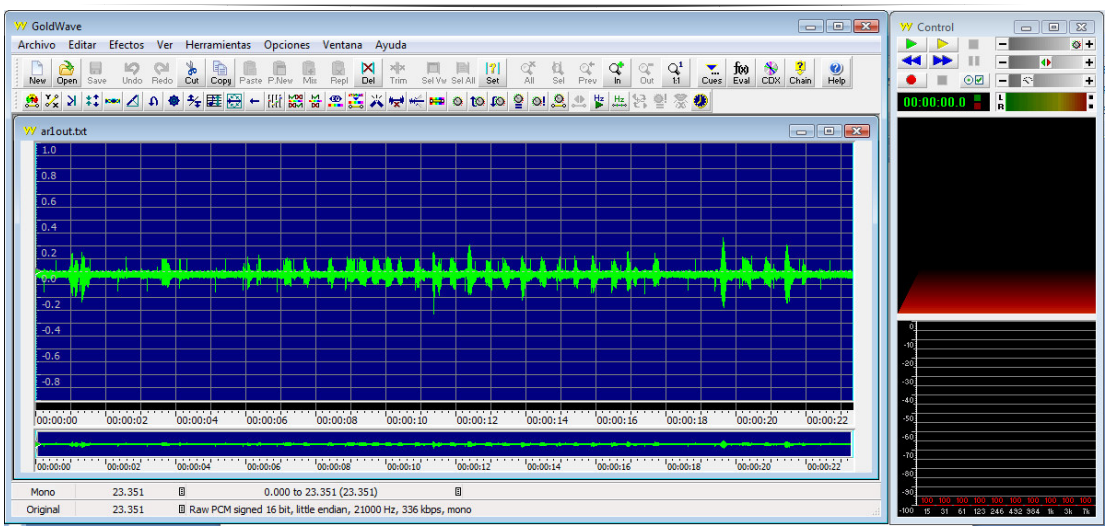


Figura 5.6. Señal 1 (señal de interés con ruido) después del filtro paso altas.

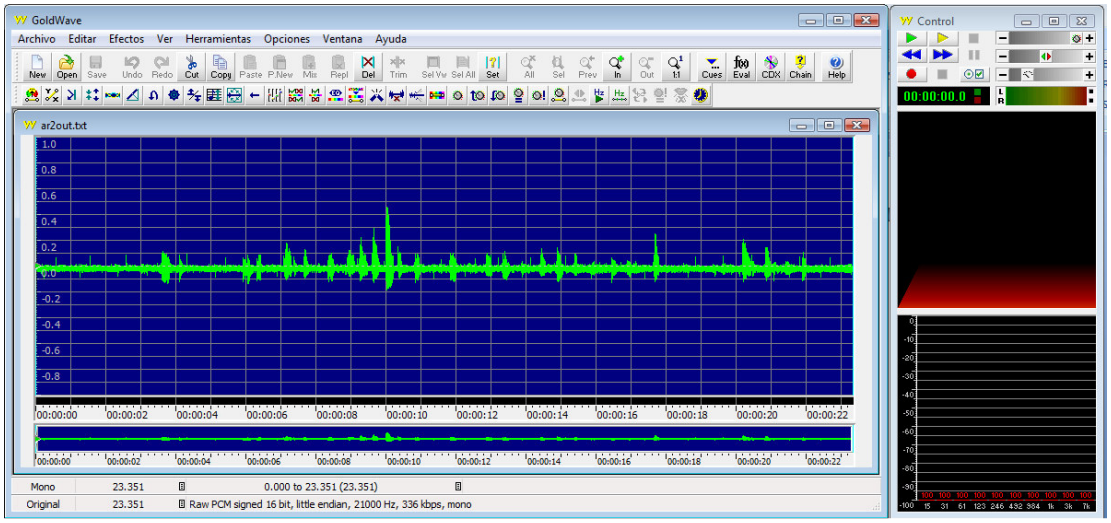


Figura 5.7. Señal 2 (señal de estimación de ruido) después del filtro paso altas.

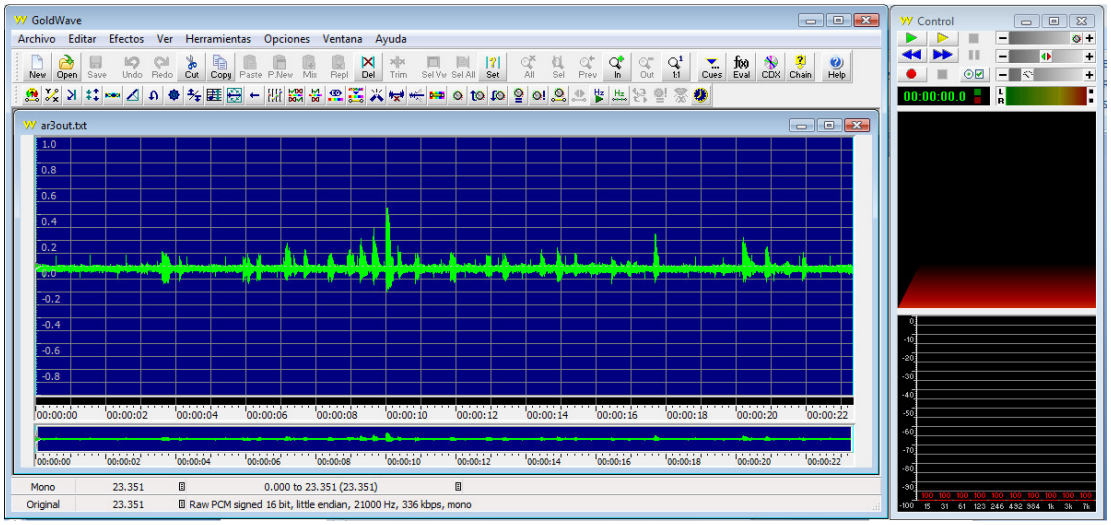


Figura 5.8. Señal 2 (señal de estimación de ruido) después del filtro paso altas.

Por último la señal de salida del filtro se presenta en la figura 5.9.

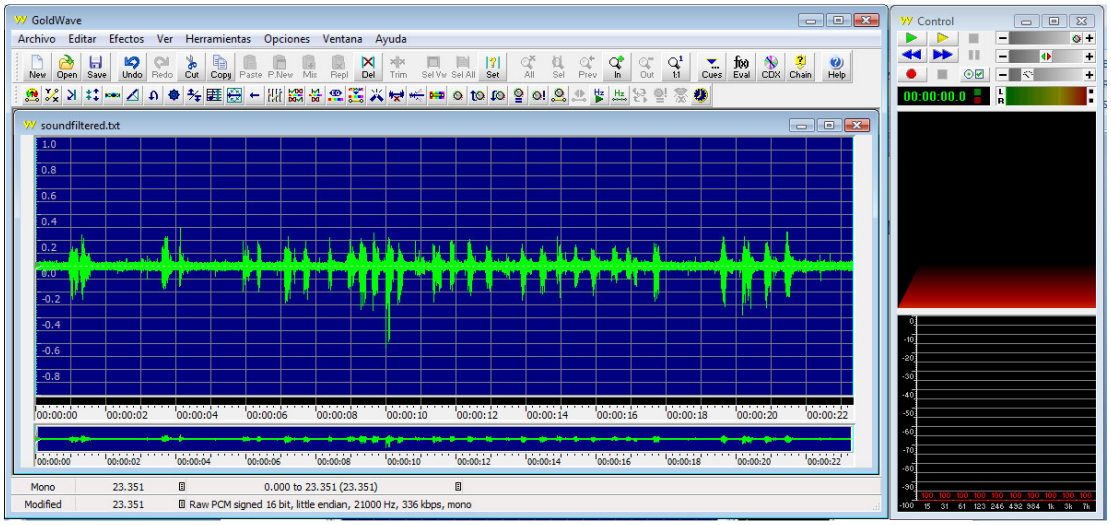


Figura 5.9. Salida del filtro adaptable con lógica difusa para la prueba 3.

5.2. Prueba 2

En esta prueba se colocan de forma estratégica los tres micrófonos, de tal forma que el micrófono principal capte tanto la señal de interés como música de fondo, mientras que los dos micrófonos restantes se ubican para percibir la música de fondo sin captar la señal de interés. A continuación se presentan las gráficas de las señales en las figuras 5.10 a 5.15.

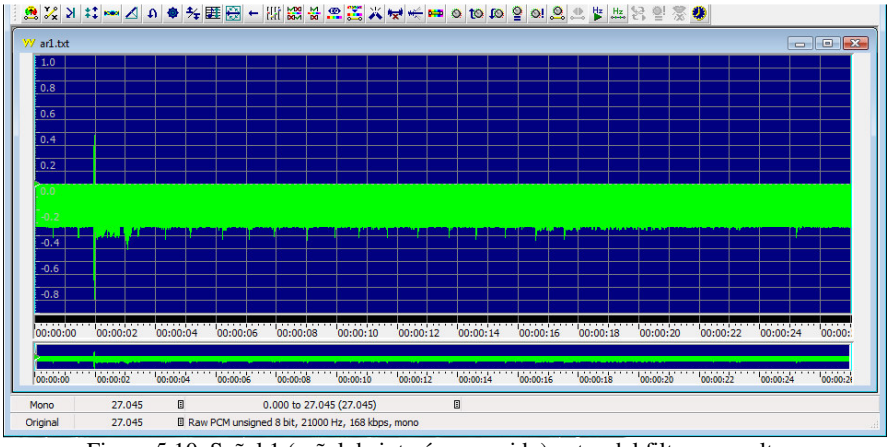


Figura 5.10. Señal 1 (señal de interés con ruido) antes del filtro paso altas.

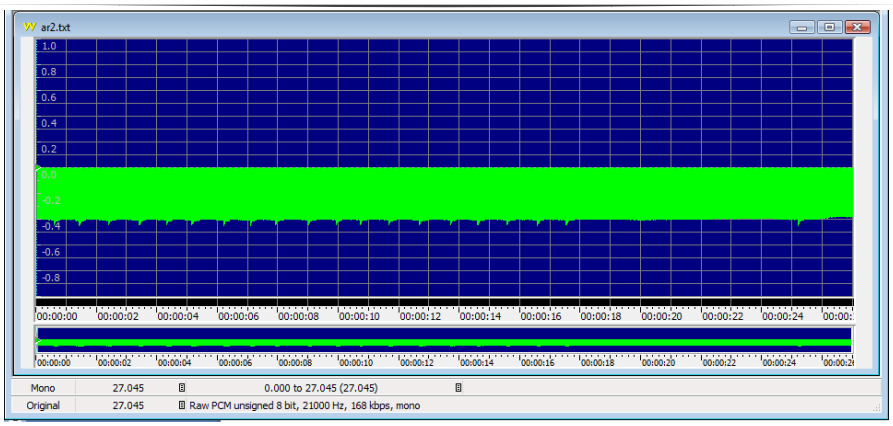


Figura 5.11. Señal 2 (señal de estimación de ruido) antes del filtro paso altas.

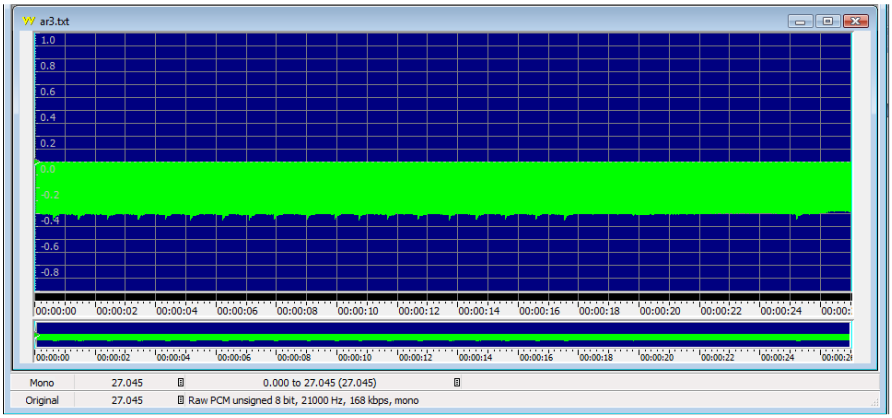


Figura 5.12. Señal 3 (señal de estimación de ruido) antes del filtro paso altas.

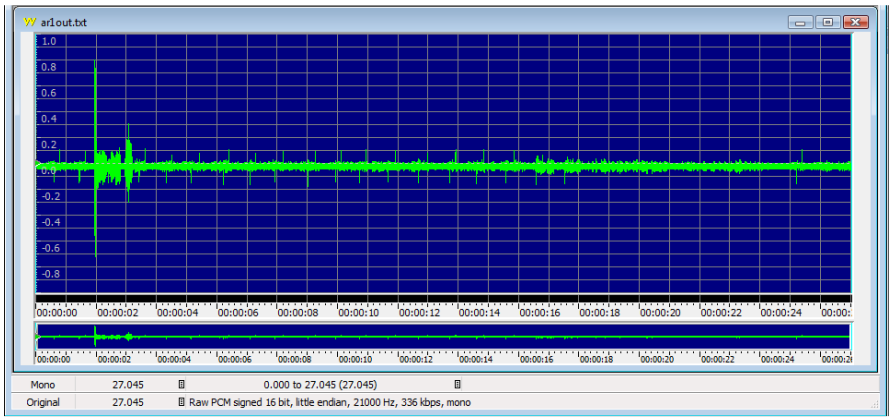


Figura 5.13. Señal 1 (señal de interés con ruido) después del filtro paso altas.

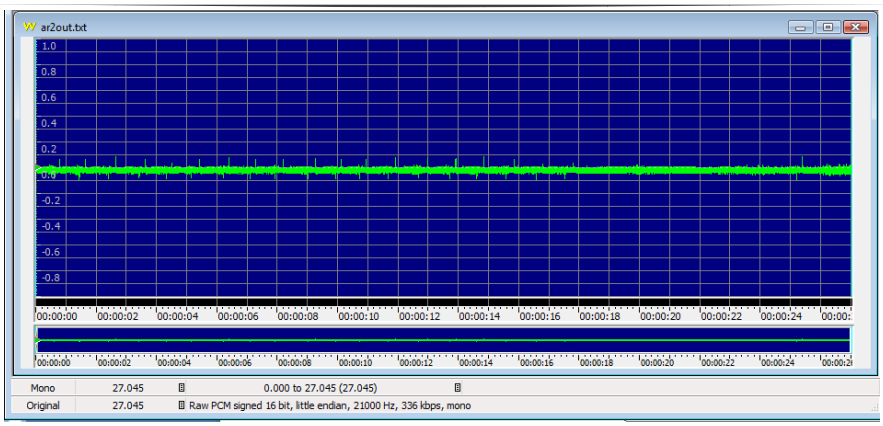


Figura 5.14. Señal 2 (señal de estimación de ruido) después del filtro paso altas.

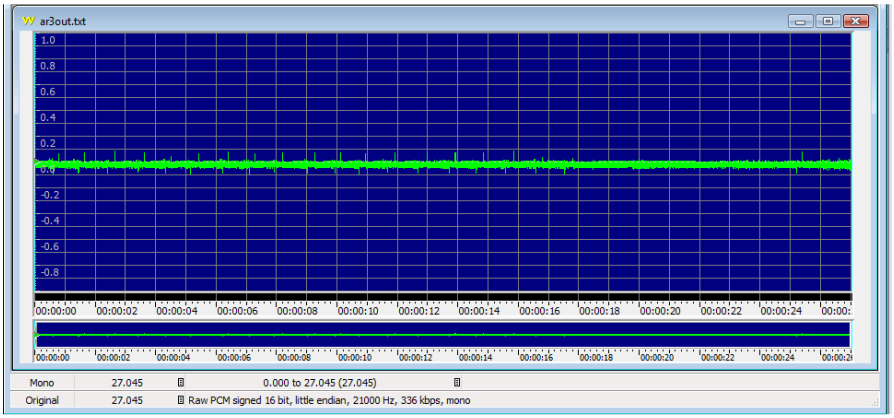


Figura 5.15. Señal 3 (señal de estimación de ruido) después del filtro paso altas.

La salida del filtro adaptable para la prueba 2 se muestra en la figura 5.25.

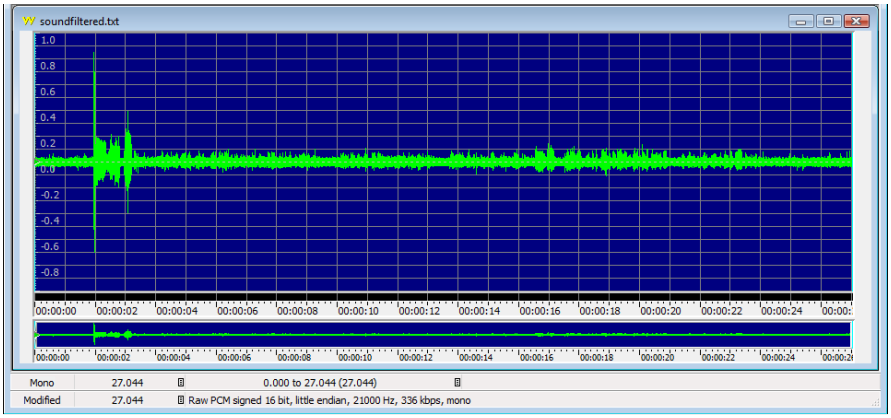


Figura 5.16. Salida del filtro adaptable con lógica difusa para la prueba 4.

5.3. Prueba 3

Esta prueba usa como entrada señales generadas digitalmente con el software GOLDWAVE, dónde la señal de interés es exactamente la misma que las señales de referencia, es decir, se supone que los micrófonos reciben únicamente una señal uniforme que es considerada ruido. Ésta señal es una senoidal con frecuencia de 500 Hz.

Las figuras 5.17, 5.18 y 5.19 muestran las señales creadas con las características ya mencionadas que sirven de entrada para el filtro.

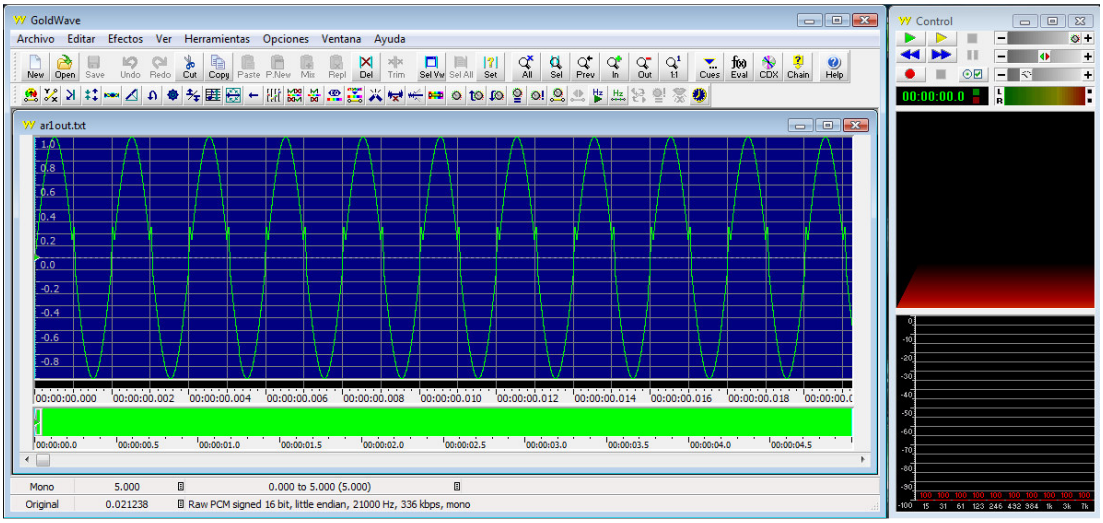


Figura 5.17. Señal 1 (señal de interés con ruido).

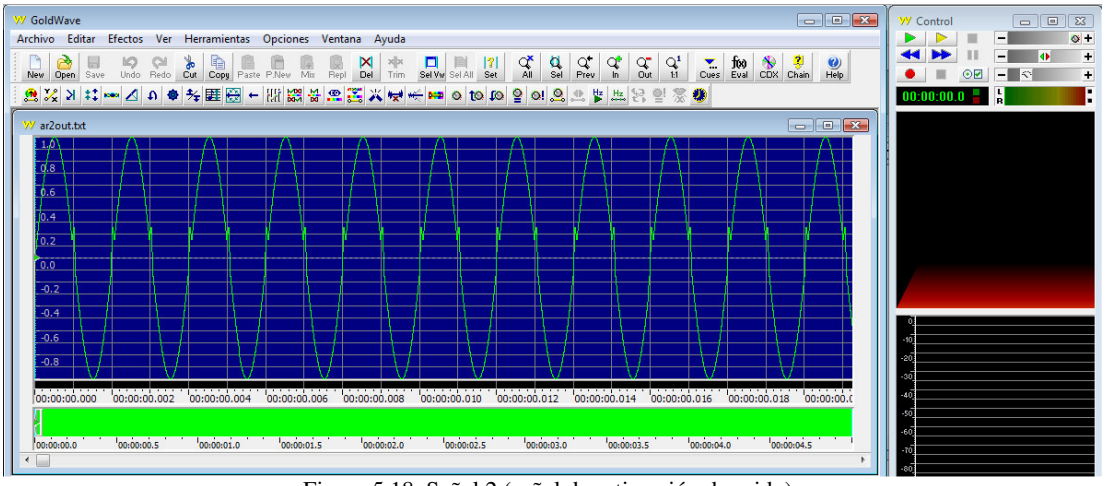


Figura 5.18. Señal 2 (señal de estimación de ruido).

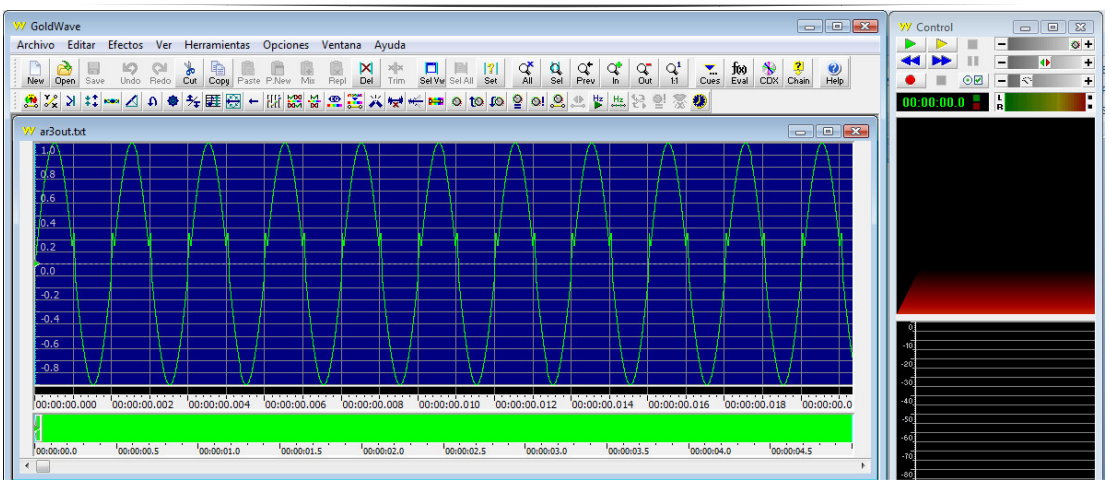


Figura 5.19. Señal 3 (señal de estimación de ruido).

La figura 5.20 muestra la señal de salida, es decir, la señal 1 de interés, filtrada, y que, como se esperaba, es silencio total, ya que las señales de referencia para estimar el ruido del medio, son exactamente iguales a la señal que se tiene como de interés.

Al inicio se aprecia la modificación rápida del coeficiente para cancelar el ruido estimado con las señales 2 y 3.

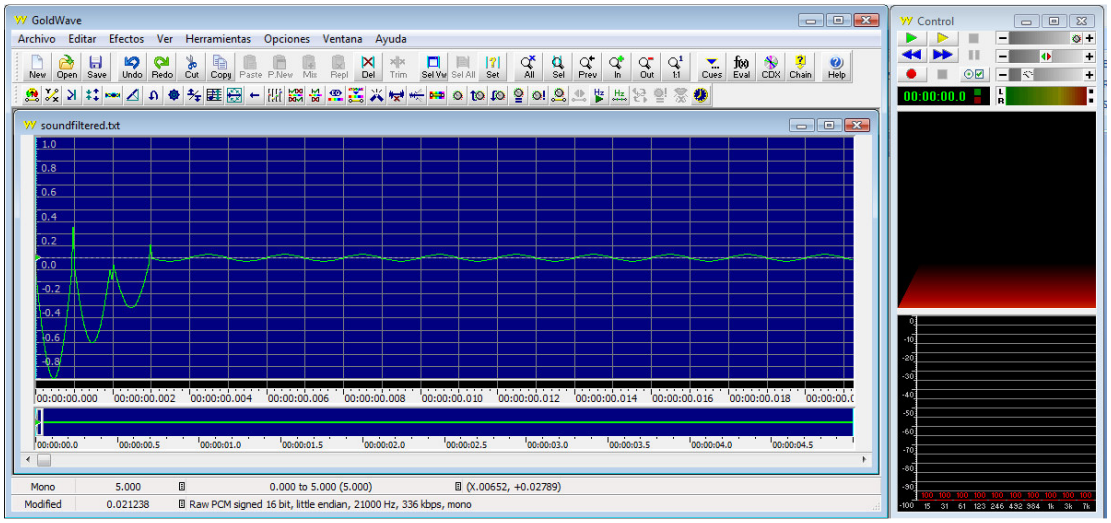


Figura 5.20. Señal de salida del filtro adaptable con lógica difusa para la prueba 1.

5.4. Prueba 4

La siguiente prueba también se realizó con señales generadas con ayuda del software GOLDWAVE, teniendo como señal de interés una mezcla de una melodía corta, una señal senoidal con frecuencia de 440Hz más, con ruido blanco y una senoidal de frecuencia de 500 Hz.

Como señales de referencia se le proporcionan al filtro una señal con ruido blanco (señal 2) y otra señal con la senoidal que oscila en 500 Hz (señal3).

Las figuras 5.21, 5.22 y 5.23 muestran las señales de entrada del filtro.

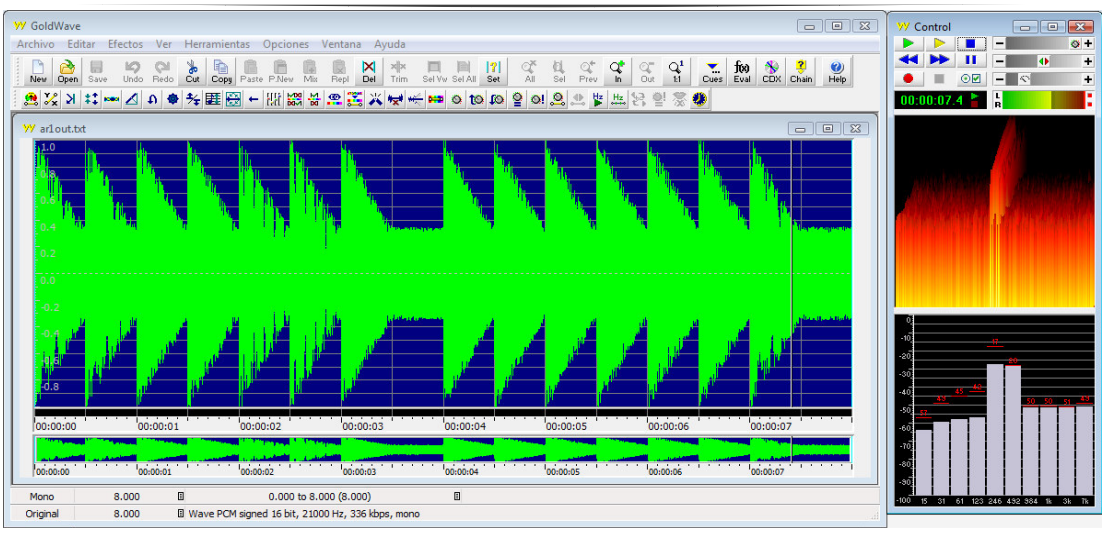


Figura 5.21. Señal 1 (señal de interés con ruido).

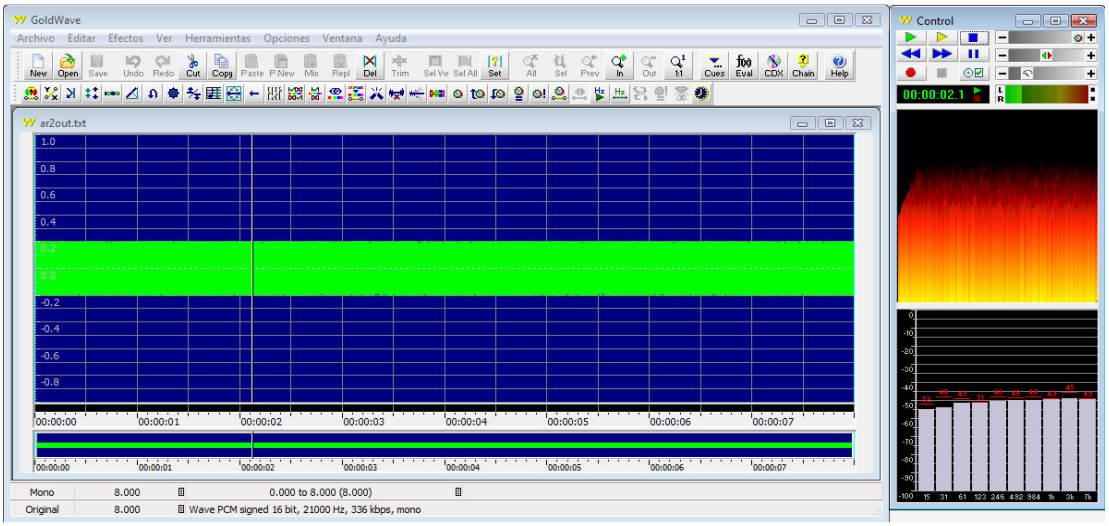


Figura 5.22. Señal 2 (señal de estimación de ruido con ruido blanco).

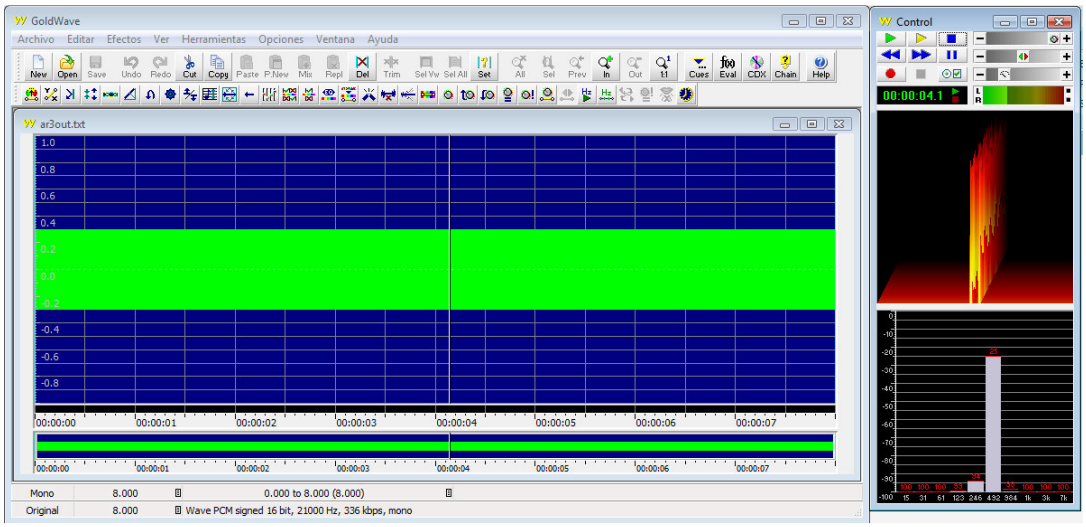


Figura 5.23. Señal 3 (señal de estimación de ruido con señal senoidal)

La figura 5.24 muestra la señal de salida del filtro, donde se puede apreciar que gran parte del ruido que se tiene en las señales de referencia pudo ser eliminado de la señal de interés.

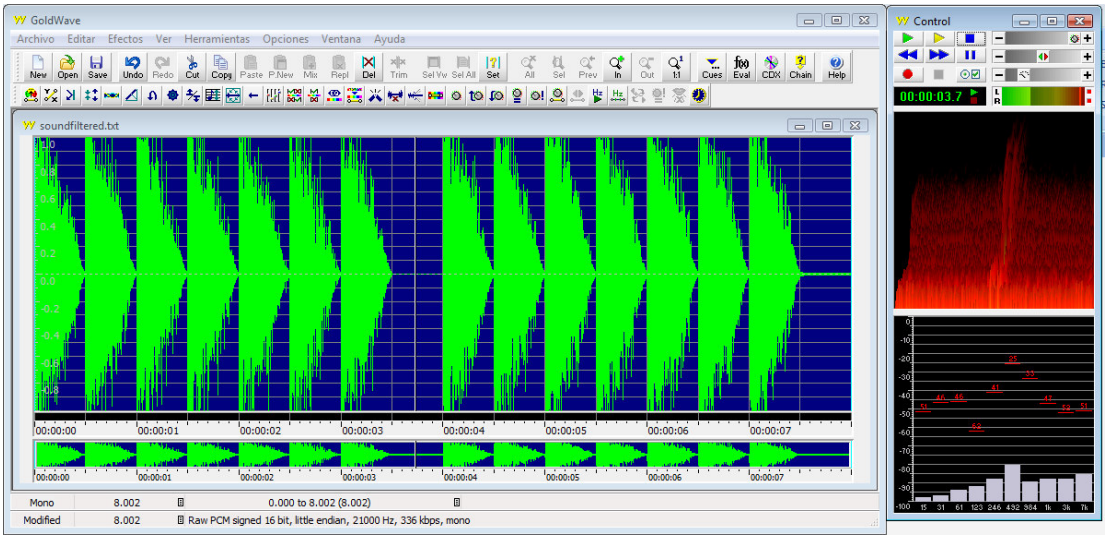


Figura 5.24. Señal de salida del filtro adaptable con lógica difusa para prueba 2.

La figura 5.25 muestra la melodía original (señal de interés) que fue contaminada con ruido en la señal 1.

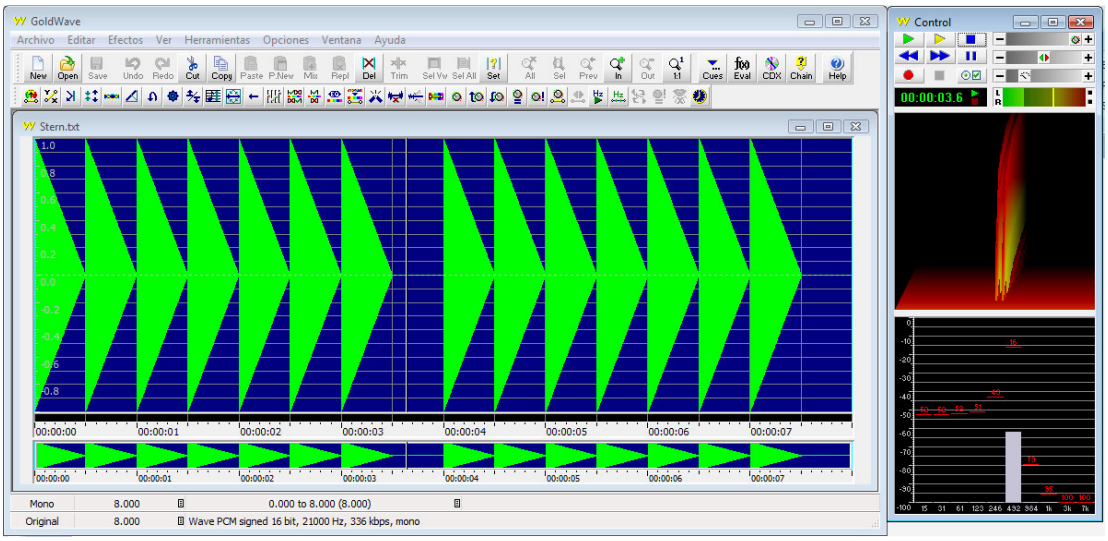


Figura 5.25. Señal de interés sin ruido.

Conclusiones

Para concluir el proyecto, es necesario analizar el desempeño de las etapas, por lo que comenzaremos con la etapa de captura de las señales, la cual es bastante eficiente, ya que la tarjeta elaborada con los amplificadores y el PIC se alimentan directamente del puerto USB de la computadora, por medio del cual se transfieren los datos a una interfaz hecha en Visual Basic, cumpliendo con el criterio de Nyquist para que las tres señales de voz capturadas sean entendibles. Esta primera etapa cuenta con la desventaja de que el convertidor analógico digital del PIC está configurado para enviar sólo 8 bits por cada muestra, por medio de una cuantización uniforme, lo que le resta un poco de calidad al audio capturado, sin embargo podemos afirmar que las señales son entendibles.

En cuanto a la etapa digital, la programación realizada en Visual Basic ha superado las limitaciones de los tipos de variables que este lenguaje de programación puede usar, ya que el tipo *double* implica números de 64 bits, con las que nos limitamos a hacer las operaciones necesarias para el procesamiento.

El tiempo de procesamiento, tanto para el filtro paso altas como para el filtro adaptable con lógica difusa, es mucho, lo que impide su aplicación en tiempo real. Por lo anterior, se capturan las señales en archivos binarios, y la respuesta es igualmente guardada en un archivo binario. Esta velocidad se debe a la cantidad de operaciones con variables de tipo *double* que se realizan para obtener la respuesta a una serie de 21 muestras de las señales y, principalmente, al proceso de desdifusión que es bastante tardado ya que se realizan aproximaciones de integrales.

Analizando las respuestas de las pruebas 3 y 4 del capítulo 5, se observa que el filtro estima y elimina eficientemente el ruido aditivo presente en la señal de interés, ya que las señales generadas por medio del software GOLDWAVE son señales cuyas muestras son capturadas en el mismo instante de tiempo, es decir, no se usa un sistema de muestreo con multiplexión en tiempo; además la amplitud de estas señales no están limitadas por un cuantificador de 8 bits/muestra, lo que involucra un menor ruido de cuantificación en comparación con las señales obtenidas por la tarjeta elaborada.

Los resultados de las pruebas 3 y 4 son altamente satisfactorios, con las cuales podemos comprobar la validez del procesamiento desarrollado, siendo de particular interés la prueba uno, dónde se

muestra una señal senoidal que se cancela al estar presente en las señales que estiman el ruido. Se puede pensar en esta senoidal como una armónica que compone una señal cualquiera (por análisis de Fourier) por lo que es factible pensar que se puede cancelar en cierta medida (dependiendo de las energias) la componente de esta armónica en la señal de interés.

Aunque el filtro está pensado para emplearse exclusivamente con señales de voz, la prueba número 2 confirma que las bases teóricas pueden ser aplicadas para señales de audio en general. Y aunque no elimina completamente el ruido aditivo en la señal de interés, podemos afirmar que lo reduce en gran medida.

Finalmente, las pruebas 1 y 2 presentan deficiencias con respecto a las pruebas anteriores, principalmente debido a las características de la forma de adquisición de las señales con gran ruido de cuantificación y el pequeño desfasamiento que existe entre muestras de distintos micrófonos al usar un sistema de multiplexión en tiempo para el muestreo. Por lo que se presenta la posibilidad de mejorar estas características para trabajos posteriores.

Los resultados del proyecto son satisfactorios en general, susceptible de futuras mejoras y de la utilización de mejores herramientas para realizar el análisis del desempeño, sin embargo, el objetivo se completó, haciendo uso de los aspectos teóricos ya descritos y de herramientas de software diversas. Se realizó una tarjeta que involucra electrónica analógica con interfaz a una computadora, representando un sistema completo que se relaciona con variables físicas del ambiente.

Es recomendable, para futuras aplicaciones similares, emplear microprocesadores más aptos para las tareas presentadas en ésta tesis, tales como DSP (procesadores digitales de señales) ya que cuentan con características favorables, como es su velocidad de procesamiento.

Bibliografía

[1] H. Taub y D. L. Schilling. *Principles of Communication Systems.* New York, EUA. 1971.

[2] C. L. Shannon. Comnunication in the presence of noise, Reimpreso en: *Proceedings of The IEEE*, vol. 72, págs. 1191-1201. 1984.

[3] A. J. Jerri. The Shannon sampling theorem - Its various extensions and applications, *Proceedings oí The IEEE*, vol. 65. págs. 1565-1596, noviembre 1977.

[4] A. Gersho. Principles of quantization, *IEEE Trans. on Circuit and Systems*, vol. CAS-25, págs. 427-435, Julio 1978.

[5] G.B. Clayton, *Data converters*. Nueva York, EUA. 1982.

[6] O.D Grace y S.P. Pitt. *Sampling and interpolation of bandlimited signals by quadrature methods,* The Journal of the Acoustical Society of América, vol. 48, págs. 1311-1318, noviembre 1969.

[7] A. V. Oppenheim, Ronald W. Schafer con John R. Buck. *Discrete-Time Signal Processing.* New Jersey, EUA. 1999.

[8] E. Soria, M. Martínez, J. V. Francés y G. Camps Valls. *Tratamiento digital de señales*. Madrid, España. 2003.

[9] A. V. Oppenheim y A. S. Willsky. *Señales y Sistemas.* México. 1998.

[10] A. Albiol, V. Naranjo, J. Prades. *Tratamiento digital de la señal, teoría y aplicaciones*. España, 1999.

[11] USB Implementers' Forum. *Device Class Definition for Human Interface Devices (HID), Firmware Specification Version 1.11*. Junio de 2001.
Disponible en Web: < http://www.usb.org/developers/devclass_docs/HID1_11.pdf>

[12] L. X Wang. *Adaptive Fuzzy Systems and Control, design and stability analysis*. New Jersey, EUA. 1994.

[13] J. G. Proakis, D. G. Manolakis. *Digital signal processing, principles, algorithms, and applications*. 2a edición. New Jersey, EUA. 1992.

[14] S. Haykin. *Adaptive Filter Theory*. 3ª Edición. EUA. 1995.

[15] A. D. Poularikas. Z. M. Ramadan. *Adaptive Filtering Primer with MATLAB*. Florida, EUA. 2006.

[16] Soporte técnico OEM de Fujitsu España. *Un paseo por USB-1.* Marzo de 200. Disponible en Web: < http://www.fujitsu.com/downloads/EU/es/soporte/discosduros/UnpaseoporUSB-1.pdf>

[17] González, Adolfo F. *Capacitores Parte I.* Universidad Tecnológica Nacional de Argentina. Argentina, 2004.

Tesis:

Névarez, Arturo. *Diseño y construcción de una interfaz USB para el control y monitoreo de instrumentación electrónica.* Tesis de Maestría. México, UNAM. 2006.

Pinzón, Julio A. y Figueroa, Armando. *Desarrollo de un sistema de control para servo motor con interfase USB.* Tesis de licenciatura. México, UNAM. 2007

Sitios web:

Sitio patrocinado por el USB Implementers Forum, Inc., creadores de la tecnología USB. Disponible en: <http://www.usb.org>

Sitio oficial de Microchip Technology Inc., fabricante y distribuidor de la familia de controladores PIC. Disponible en: <http://www.microchip.com>

Página oficial de Custom Computer Services Inc. Desarrolladores del compilador CCS de lenguaje C para PICs. Disponible en: <http://www.ccsinfo.com>

Página de Jaime Fdez-Caro Belmonte con proyectos, desarrollos, y ejemplos de programación de PICs. Disponible en: < http://www.hobbypic.com>

Página con teoría y múltiples ejemplos de proyectos en los que se involucran PICs.
Disponible en: < http://picmania.garcia-cuervo.net>

Guía de micrófonos electret de Hosiden Corporation.
Disponible en: < http://www.hosiden.co.jp/web/english/web/products/pdf/e_on06_mic.pdf>

Apuntes sobre diseño y características de filtros.
Disponible en: <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4040003/lecciones/> y en <http://www1.ceit.es/asignaturas/tratamiento%20digital/frametds5.html>

Apéndice A

Universal Serial Bus

La interfaz denominada *Universal Serial Bus* (USB) es un estándar para comunicaciones desarrollado en 1996 por las empresas IBM, Intel, Northern Telecom, Compaq, Microsoft, Digital Equipment Corporation y NEC, introduciendo al mercado un nuevo puerto que permite la conexión de diversos periféricos a una computadora y que ofrece comunicación bidireccional y semi-dúplex.

Interfaz USB

Para la comunicación mediante USB se usa un cable con 4 conductores transmitiendo una señal diferencial (D+ y D-) y alimentación (VCC = 5[V] y GND) de acuerdo a la siguiente tabla:

Nombre	Descripción	Color del cable
V_{CC}	+ 5 [V]	Rojo
D-	Señal Diferencial	Blanco
D+	Señal Diferencial	Verde
GND	Tierra	Negro

Tabla A.1. Descripción de los conductores de un cable USB.

Dependiendo de las características eléctricas del cable, la longitud máxima para una correcta comunicación es de 3[m] para la especificación USB1 y de hasta 5[m] para USB2.
Dentro de la arquitectura USB se emplea una estructura piramidal de diversas capas o niveles (figura A.1). En el primer nivel se encuentra el controlador actuando como concentrador raíz, éste elemento reside dentro de la PC y es responsable de las comunicaciones entre los periféricos USB y la CPU de la PC. También es responsable de la admisión de los periféricos dentro del bus, tanto si se detecta una conexión como una desconexión. Para cada periférico añadido, el controlador determina su tipo y le asigna una dirección lógica para utilizarla siempre en las comunicaciones con el mismo. Si se producen errores durante la conexión, el controlador lo comunica a la CPU, que, a su vez, lo transmite al usuario. Una vez se ha producido la conexión correctamente, el controlador asigna al periférico los recursos del sistema que éste precise para su funcionamiento.

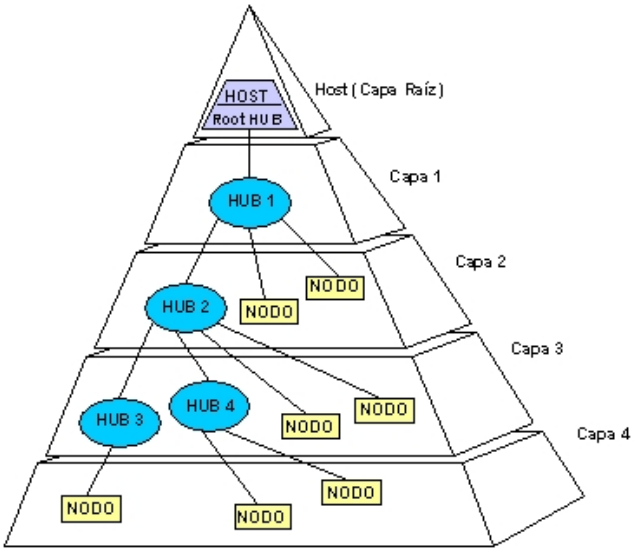


Figura A.1. Estructura piramidal de la arquitectura USB.

En los siguientes niveles se encuentran nodos y/o *hubs.* Los *hubs* o concentradores son los elementos que detectan la conexión y desconexión de dispositivos así como si son de alta o baja velocidad, también llevan tanto la alimentación como la comunicación hacia los dispositivos que se encuentran conectados de acuerdo a una topología de estrella, pudiendo tener solo un dispositivo conectado a cada nodo del hub.

De acuerdo al protocolo que se utiliza, el controlador USB transmite tokens (testigos) que incluyen la dirección del dispositivo destino, y el dispositivo que detecta su dirección en el token responde y lleva a cabo la transferencia de datos con el controlador. Así, el controlador USB maneja la parte más compleja del protocolo, generando tokens de transferencias de datos a 12 Mbps o a 1.5 Mbps y controlando la conexión lógica entre el sistema y las funciones internas de cada dispositivo. También es el controlador el que maneja el consumo de energía en el bus a través de las funciones suspender/continuar, por medio de las cuales controla los modos reposo/activo de los dispositivos. [16].

Enumeración del bus

La enumeración del bus es el proceso que identifica a los dispositivos conectados a él, para ello se utilizan dos identificadores:

- **VID.** Es un número de 16 bits que significa *Vendor Identification* o código que identifica al fabricante del hardware a conectar. En nuestro caso utilizamos el número **0461h**.
- **PID.** Es un número de 16 bits que significa *Product Identification* o código que identifica al dispositivo en concreto hardware a conectar. En nuestro caso utilizamos el número **0020h**.

Cuando el Sistema Operativo detecta la conexión con un dispositivo USB recibirá el VID y el PID especificados en el *firmware* y buscará entre sus *drivers* instalados para encontrar el que corresponde a esta identificación, si no la encuentra nos preguntará sobre donde ha de buscar un driver adecuado y deberemos indicarle su ubicación. Este *driver* deberá estar configurado para conectar con un *hardware* cuyos VID y PID sean los mismos. El driver para puertos serie ya existe en Windows, aquí solo debemos darle el enlace con el .inf para que conecte correctamente con el firmware que tiene el PIC.

Como el puerto USB permite la conexión y desconexión de dispositivos en cualquier momento, la enumeración se realiza de manera permanente en busca de nuevas dispositivos conectados dentro del bus.

Velocidad

Actualmente (octubre de 2008) existen tres niveles de velocidad posibles para los dispositivos USB:

Low speed. 1.5 Mbps, es soportada por el estándar de la versión 1 de USB, se utiliza debido a que en diversas ocasiones los periféricos tales como teclados y ratones que requieren cables flexibles siendo estos de baja velocidad. Utilizando dispositivos con este nivel de velocidad se puede tener un mayor ancho de banda para aplicaciones que así lo requieran.

Full speed. 12 Mbps, también fue definida en la versión 1 de USB, se pensó en que con esta velocidad la gama de aplicaciones sería mucho mayor permitiendo sustituir paulatinamente el uso de otros puertos como el paralelo y el serie.

High speed. 480 Mbps, esta velocidad solo es soportada por USB 2 por lo que no es soportada por todas las computadoras, ya que se requiere que esté equipada con hardware mas reciente.

Especificación HID

La especificación HID (*Human Interface Devices*) es una ampliación de la norma USB para facilitar a los fabricantes la construcción de este tipo de dispositivos. La norma especifica cómo deben extraer los datos de los dispositivos USB los drivers HID.

La definición HID surgió con la idea de permitir una mayor uniformidad en la manera de realizar la comunicación necesaria entre la computadora y los periféricos a través del puerto USB, que se requiere para trabajar con dispositivos con los que interactúan posibles usuarios humanos, tales como teclados, ratones y *joysticks*; de ahí su nombre.

El HID define dos roles, el de dispositivo de interfaz humana (HID) y el de anfitrión:

- **Dispositivo de interfaz humana (HID)** – Es el dispositivo que proporciona el servicio de entrada y salida de datos de procedencia humana desde y hacia el anfitrión.
- **Anfitrión o *host*** – El dispositivo que usa o solicita los servicios de un dispositivo de interfaz humana.

De acuerdo al protocolo HID los dispositivos definen sus paquetes de datos y luego presentan un descriptor HID al host. El descriptor HID es codificado como un arreglo de bytes que describen los paquetes de datos del dispositivo, la descripción se integra por: cuantos paquetes soporta el dispositivo, que tan grandes son los paquetes, y el propósito de cada byte y bit en el paquete. Por ejemplo, un joystick con un botón puede decirle al host que el estado de presionar/soltar ese botón, es almacenado en el primer bit del quinto byte en el paquete de datos número 2.

El host debe tomar el descriptor HID del dispositivo y llevar a cabo el *parcing*[1] necesario antes que se puede entablar la comunicación con el dispositivo. Realizar el *parcing* del descriptor HID puede ser complicado.

El mecanismo anterior describe los que es conocido como el modo de reporte HID. El HID también define el modo de arranque. En modo arranque, solo soporta algunos dispositivos con características especificas ya que se usan paquetes de datos ya definidos. Los únicos dispositivos soportados en modo arranque son:

- Teclado. Solo el núcleo de 104 teclas es soportado. Cualquier funcionalidad avanzada no está disponible.
- Ratón. Solo los ejes X y Y, la rueda y los primeros 3 botones están disponibles. Cualquier característica adicional en el ratón no funcionará.

Un uso común del modo arranque esta durante las primeras secuencias cuando la computadora arranca. La configuración del *setup* del BIOS (*Basic Input Output System*) de la computadora es a menudo hecha usando el modo de arranque.

La información referente al dispositivo USB se almacena en segmentos de ROM llamados descriptores. Un descriptor de interfaz puede identificar a un dispositivo como perteneciente a un número finito de clases. En el caso de un dispositivo USB/HID, se usa el *driver* correspondiente a las clase HID para direccionar toda la información trabajando en conjunto con el descriptor del dispositivo.

La clase HID utiliza diversos descriptores mostrados en la tabla A.2.

Descriptores estándar USB	De dispositivo
	De configuración
	De interfaz
	De *endpoint*
	De cadena
Descriptores específicos	HID
	De Reporte
	Físico

Tabla A.2. Descriptores de la clase HID

Estructura del descriptor de dispositivo

En el nivel superior, un descriptor incluye dos tablas de la información que se referidas como descriptor del dispositivo (*device descriptor*) y descriptor de cadena (*string descriptor*). Un descriptor de dispositivo USB estándar especifica el ID del producto y otra información sobre el dispositivo. Por ejemplo, los principales campos del descriptor de dispositivo pueden incluir:

- Clase
- Subclase
- Vendedor

[1] *Parcing.* Proceso de analizar una secuencia de símbolos a fin de determinar su estructura gramatical con respecto a una gramática formal dada. Formalmente es llamado análisis de sintaxis. Un parseador (*parser)* es un programa de computación que lleva a cabo esta tarea.

- Producto
- Versión

En el caso particular de HID:

- El tipo de clase no se define en el nivel del descriptor de dispositivo. El tipo de clase para un dispositivo de clase HID es definido por el descriptor de interfaz.
- La subclase campo se utiliza para identificar dispositivos de arranque.

Descriptor HID

El descriptor HID identifica la longitud y el tipo de descriptores subordinados al dispositivo. Sus partes son:

Parte	Offset / tamaño en bytes	Descripción
bLenght	0/1	Expresión numérica que es el tamaño total del descriptor HID
bDescriptorType	1/1	Constante que especifica el tipo de descriptor HID
bcdHID	2/2	Expresión numérica que especifica la versión de la especificación de la clase HID
bCountryCode	4/1	Expresión numérica que especifica el código del país donde se localiza el hardware.
bNumDescriptor	5/1	Expresión numérica que indica el número de descriptores de clase.
bDescriptorType	6/1	Constante que identifica el tipo de descriptor de clase.
wDescriptorLenght	7/2	Expresión numérica del tamaño total del descriptor de reporte.
[bDescriptorType]…	9/1	Constante que indica el tipo de descriptor opcional.
[wDescriptorLenght]…	10/2	Expresión numérica del tamaño total del descriptor opcional.

Si un descriptor es opcional especificado, una entrada de longitud correspondiente también debe ser especificada.

Descriptor de Reporte

Este descriptor es diferente de otros descriptores en el sentido de que no es simplemente una tabla de valores. La longitud y el contenido de un descriptor de informe pueden variar en función del número de campos de datos necesarios para el informe o los informes del dispositivo.

Está compuesto de los elementos que proporcionar información sobre el dispositivo. La primera parte de un elemento contiene tres campos: tipo de elemento, la etiqueta del elemento, el tamaño del elemento. Juntos estos campos identifican el tipo de información que proporciona el elemento. Hay tres tipos de elementos: *Main*, Global, y Local.

El descriptor de reporte proporciona una descripción de los datos proporcionados por cada control en un dispositivo. Cada etiqueta principal de elemento *main* (de entrada, de salida, o de funciones)

identifica el tamaño de los datos devueltos por un control particular, e identifica si los datos son absolutos o relativos, y otra información pertinente.

Un descriptor de reporte puede contener varios elementos principales pero incluir cada uno de los siguientes elementos para describir un control de datos (todos los demás elementos son opcionales):

- De entrada
- De uso
- De uso de página
- Mínimo lógico
- Máximo lógico
- Tamaño del reporte
- Cuenta del reporte

La siguiente es una muestra de codificación de los elementos que se utilizan para definir un ratón de 3 botones. En este caso, los elementos principales (*main*) son precedidos por elementos globales tales como Usage, Report Count o Report Size (cada línea es un nuevo elemento). [11]

```
Usage Page (Generic Desktop), ;Use the Generic Desktop Usage Page
Usage (Mouse),
Collection (Application), ;Start Mouse collection
Usage (Pointer),
Collection (Physical), ;Start Pointer collection
Usage Page (Buttons)
Usage Minimum (1),
Usage Maximum (3),
Logical Minimum (0),
Logical Maximum (1), ;Fields return values from 0 to 1
Report Count (3),
Report Size (1), ;Create three 1 bit fields (button 1, 2, & 3)
Input (Data, Variable, Absolute), ;Add fields to the input report.
Report Count (1),
Report Size (5), ;Create 5 bit constant field
Input (Constant), ;Add field to the input report
Usage Page (Generic Desktop),
Usage (X),
Usage (Y),
Logical Minimum (-127),
Logical Maximum (127), ;Fields return values from -127 to 127
Report Size (8),
Report Count (2), ;Create two 8 bit fields (X & Y position)
Input (Data, Variable, Relative), ;Add fields to the input report
End Collection, ;Close Pointer collection

End Collection ;Close Mouse collection
```

Esquema básico de comunicación de dispositivos HID

Los dispositivos de clase HID se comunican con el *driver* de la clase HID utilizando el pipe[2] de control o un pipe de interrupción como se observa en la siguiente figura:

[2] El modelo de transferencia de datos USB entre la fuente o destino y el punto de salida en un dispositivo es conocido como conducto o *pipe*.

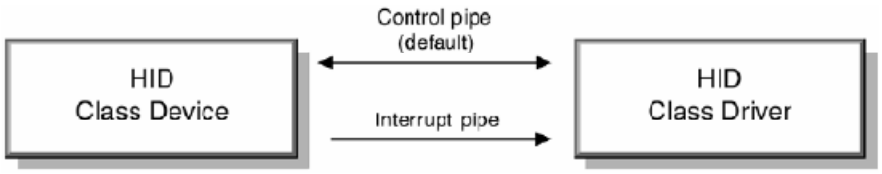


Figura. Comunicación entre el dispositivo HID y el *driver* HID.

El pipe de control se utiliza para:

- Recibir y responder las solicitudes de control USB y la clase de datos.
- Transmisión de los datos encuestados por el controlador de clase HID.
- Recepción de datos desde el host.

Mientras que el pipe de interrupción se usa para:

- Recepción asíncrona de datos desde el dispositivo.
- Transmisión de baja latencia de datos al dispositivo.