

“EFECTO DE LA PALABRA FINITA EN LA
IMPLEMENTACION DE FILTRO DIGITAL”

CARLOS ANDRES PEREZ PAEZ
DIANA PATRICIA TOSCANO QUINTERO

UNIVERSIDAD PONTIFICIA BOLIVARIANA
FACULTAD DE INGENIERIA ELECTRONICA
ESCUELA DE CIENCIAS BASICAS
BUCARAMANGA
2008

EFFECTO DE LA PALABRA FINITA EN LA
IMPLEMENTACION DE FILTO DIGITAL”

por

CARLOS ANDRES PEREZ PAEZ
DIANA PATRICIA TOSCANO QUINTERO

Tesis propuesta para obtener el titulo de ingeniero
electrónico

Asesor

EDGAR BARRIOS UREÑA
Msa Ingeniero Eléctrico

UNIVERSIDAD PONTIFICIA BOLIVARIANA
FACULTAD DE INGENIERIA ELECTRONICA
ESCUELA DE CIENCIAS BASICAS
BUCARAMANGA
2008

Nota de Aceptación

Firma del Presidente del Jurado

Firma del Jurado

Firma del Jurado

Bucaramanga, Julio de 2008

A nuestros padres

AGRADECIMIENTOS

A todas aquellas personas que de alguna u otra manera creyeron en nosotros para el desarrollo de nuestra tesis al igual que nuestro crecimiento como profesionales de bien,

A nuestro director de tesis el cual es nuestro modelo como ingeniero y padre dentro de la institución, por su grandísimo apoyo y dedicación en el transcurso de todos estos meses de dura investigación.

A todos los amigos que nos apoyaron y ayudaron en los momentos más difíciles del desarrollo de la tesis. A nuestras familias los cuales estuvieron presentes y con su apoyo nos dieron fuerzas en el momento en la que ya no teníamos

Al ingeniero Jesús Vega por su tiempo y conocimientos en el uso de matlab. Y al ingeniero Juan Carlos Villamizar por su apoyo técnico.

CONTENIDO

	Pág.
INTRODUCCION	1
1. FILTRO DIGITAL	2
1.1 CONCEPTO GENERAL	2
1.2 REPRESENTACION NUMERICA DE LOS FILTROS	2
1.3 TIPOS DE FILTROS	4
1.4 FILTROS FIR	4
1.5 FILTROS IIR	5
1.6 FILTROS ADAPTATIVOS	7
2. ESTRUCTURA DE LOS FILTROS DIGITALES	8
2.1 ESTRUCTURA 1D	8
2.2 ESTRUCTURA 2D	9
2.3 ESTRUCTURA 3D	10
2.4 ESTRUCTURA 4D	11
2.5 ESTRUCTURA DE ACOPLAMIENTO 1X y 2X	12
3. REALIZACION DE FILTROS	14
3.1 FILTROS EN CASCADA	14
3.2 FILTROS EN PALALELO	15
3.3 FILTROS EN ESCALERA	17
4. EFECTO DE LA PALABRA FINITA	18
4.1 SISTEMA NUMERICO DE PUNTO FIJO	18
4.1.1 SISTEMA NUMERICO SIGNO Y MAGNITUD	18
4.1.1.1 CUANTIZACION POR TRUNCAMIENTO	19
4.1.1.2 CUANTIZACION REDONDEO	21
4.1.1.3 CUANTIZACION LSB-1	23
4.1.1.4. RELACION DE LOS TRES TIPOS DE CUANTIZACION	25
4.1.1.5. RANGO DINAMICO	26
4.1.2. SISTEMA DE COMPLEMENTO A DOS	27
4.1.2.1 CUANTIZACION POR TRUNCAMIENTO	27
4.1.2.2. CUANTIZACION ROUND-OFF	28
4.1.2.3. CUANTIZACION LSB-1	29
4.1.2.4 RELACION DE LOS TRES TIPOS DE CUANTIZACION	30
4.1.2.5 RANGO DINAMICO	31

4.2 COEFCIENTE DE CUANTIZACION	32
4.2.1 Ubicación de polos y ceros	32
4.3 ANALISIS DE LA SEÑAL DE CUANTIZACION	35
4.3.1 CUANTIZACION DE LA ENTRADA DEL FILTRO	35
4.3.2 CUANTIZACION DE LA VARIABLES INTERNAS	38
4.3.3 RUIDO DE CUANTIZACION	41
4.4 CICLO LÍMITE	45
4.4.1 DEFINICION DE CICLO LÍMITE	45
4.4.2 CLASIFICACION DE ERRORES DE CUANTIZACION	50
4.4.3 OSCILACIONES DEL DESBORDAMIENTO	50
4.4.4 CONDICIÓN PARA AUSENCIA DE OSCILACIONES DE DESBORDAMIENTO	50
4.5 FILTROS DE SEGUNDO ORDEN EN CASCADA	51
4.5.1 ESCALAMIENTO DE LA SEÑAL	52
4.5.1.1. ESCALAMIENTO DEL LIMITE SUPERIOR	52
4.5.2 OPTIMIZACION	53
4.5.3 PARIEDAD DE POLOS Y CEROS	54
4.5.4 ORDENAMIENTO	55
4.6 FILTROS DE SEGUNDO ORDEN EN PARALELO	57
5. ANALISIS DE UN COTROLADOR PID UTILIZANDO EL FDATAOL	61
5.1 EJEMPLO	62
5.2 VENTANA DE DISEÑO DE FILTRO Y GRAFICA DE ESPECIFICACIONES DEL FILTRO	63
5.3. VENTANA DE IMPORTACION DESDE EL WORKSPACE Y GRAFICA DE RESPUESTA EN MAGNITUD	64
5.4. VENTANA DE MODELACION Y OPTIMISACION DEL FILTRO CON GRAFICA DE FASE Y MAGNITUD	65
5.5 VENTANAS DE KIT DE CUANTIZACION	66
5.6. VENTANA DE TRANSFORMACIONDE FILTRO Y GRAFICA GRUPO DE ATRASO	69
5.7 GRAFICA DE ATRASO DE FASE	70
5.8. VENTA DE CREACION DE UN FILTRO MULTIRATE Y GRAFICA DE RESPUESTA AUN IMPULSO	71
5.9 GRAFICA DE RESPUESTA A UNA SEÑAL ESCALON	72
5.10. GRAFICA DE COMPORTAMIENTO DE POLOS Y CEROS	73
5.11 COEFICIENTES DEL FILTRO	74
5.12. REPORTE DEL FILTRO	75
5.13. GRAFICA DE RESPUESTA ESPERADA	76
5.14. GRAFICA DE POTENCIA DEL ESPECTRO DEL RUIDO	77
5.15 ANALISIS	78
CONCLUSIONES	79

REFERENCIA BIBLIOGRAFICA	80
REFERENCIA PAGINAS DE INTERNET	81
ANEXO A “TOOLBOX DE PUNTO FIJO”	I
ANEXO B “TOOLBOX DE DISEÑO DE FILTROS”	XCIX

LISTA DE TABLAS

	PAG.
TABLA 2.1 PROPIEDADES DE LAS ESTRUCTURA DIRECTAS	7
TABLA 3.1 PROPIEDADES DE LAS ESTRUCTURAS DIRECTAS EN CASCADAS	15
TABLA 3.2 PROPIEDADES DE LAS ESTRUCTURAS DIRECTAS EN PARALELO	16
TABLA 4.1 VALORES DE LOS COEFICIENTES DEL FILTRO 1D DE PRIMER ORDEN.	46
TABLA 4.2 VALORES DE LOS COEFICIENTES DEL FILTRO 1D DE SEGUNDO ORDEN.	48
TABLA 4.3 VALORES DE LOS COEFICIENTES DEL FILTRO 1D DE SEGUNDO ORDEN.	49
TABLA 4.4 CLASIFICACION DE LOS ERRORS DE CUANTIZACION.	50

LISTA DE FIGURAS

	Pág.
Figura 1.1 Sistema de filtrado	2
Figura 1.2 Ejemplo de palabra binaria de 3 bits	3
Figura 1.3 Representación de una palabra binaria de 8 bits.	3
Figura 1.4 Estructura del Filtro FIR	5
Figura 1.5 Estructura de Filtro IIR	6
Figura 2.1 Estructura 1D	9
Figura 2.2 estructura 2D	10
Figura 2.3 estructura 3D	11
Figura 2.4 Estructura 4D	12
Figura 2.5 Estructura 1X.	13
Figura 2.6 Estructura 1X.	14
Figura 3.1 Conexión en Cascada	15
Figura 3.2 Conexión en Paralelo.	16
Figura 4.1 Característica de cuantización para sistema numérico de signo y magnitud	20
Figura 4.2 Error de probabilidad de la cuantización de truncamiento para un sistema signo-magnitud.	21
Figura 4.3 Característica de cuantización por redondeo para sistema numérico de signo y magnitud.	22
Figura 4.4 Error de probabilidad de la cuantización de redondeo para un sistema signo-magnitud.	23
Figura 4.5 Característica de cuantización por LSB-1 para sistema numérico de signo y magnitud.	24
Figura 4.6 Error de probabilidad de la cuantización de LSB-1 para un sistema signo-magnitud.	25
Figura 4.7 Equivalente de cuantizadores de redondeo	25
Figura 4.8 Equivalente de cuantizadores de LSB-1	26
Figura 4.9 Característica de overflow para el sistema signo-magnitud	26
Figura 4.10 Característica de cuantización por truncamiento para el sistema de complemento a dos.	27
Figura 4.11 Error de probabilidad de la cuantización por truncamiento para el sistema de complemento a dos.	28
Figura 4.12 Error de probabilidad de la cuantización por redondeo para el sistema de complemento a dos.	28
Figura 4.13 Característica de cuantización por redondeo para el sistema de complemento a dos.	29

Figura 4.14 Error de probabilidad de la cuantización por lsb-1 para el sistema de complemento a dos.	29
Figura 4.15 Característica de cuantización por lsb-1 para el sistema de complemento a dos.	30
Figura 4.16 Equivalente de cuantizadores de redondeo.	31
Figura 4.17 Equivalente de cuantizadores de LSB-1redondeo.	31
Figura 4.18 Equivalente de cuantizadores de LSB-1redondeo.	33
Figura 4.19 Estructura 1X.	35
Figura 4.20 Conversor A/D	36
Figura 4.21 Conversor A/D de aproximaciones sucesivas	36
Figura 4.22 Conversor A/D de aproximaciones sucesivas bias.	37
Figura 4.23 Característica de overflow para el conversor A/D	37
Figura 4.24 Modelo de conversor A/D con una señal de ruido.	38
Figura 4.25 Caso ideal de la cuantización interna de variables.	39
Figura 4.26 Caso físicamente realizable de la cuantización interna de variables.	39
Figura 4.27 Error de la probabilidad de densidad de redondeo	41
Figura 4.28 Modelo del ruido del filtro	42
Figura 4.29 Modelo del ruido de cada termino interno del filtro	42
Figura 4.230 Modelo del ruido de cada termino interno del filtro	43
Figura 4.31 Filtro 1D de primer orden.	46
Figura 4.32 Filtro 1D con cuantización múltiple.	48
Figura 4.33 Oscilaciones de Overflow.	51
Figura 4.34 Pariedad de polos y ceros	55
Figura 4.35 Modelo de ordenamiento para un modulo 1D	56
Figura 4.36 Modelo de implementación general de paralelo	58
Figura 4.37 Modelo de implementación general de paralelo para módulos 1D y 4D	59
Figura 5.1 Diagrama de bloques en simulink	62
Figura 5.2 Respuesta al sistema	62
Figura 5.3 FDATAOOL. Especificaciones del filtro.	63
Figura 5.4 FDATAOOL. Respuesta en magnitud del filtro.	64
Figura 5.5 FDATAOOL. Respuesta en magnitud y respuesta en fase del filtro.	65
Figura 5.6 FDATAOOL. Parámetros de cuantización	66
Figura 5.7 FDATAOOL. Parámetros de cuantización	67
Figura 5.8 FDATAOOL. Parámetros de cuantización	68
Figura 5.9 FDATAOOL. Grupo de retardo	69
Figura 5.10 FDATAOOL. Retardo de fase	70
Figura 5.11 FDATAOOL. Respuesta al impulso	71
Figura 5.12 FDATAOOL. Respuesta a la señal escalón	72
Figura 5.13 FDATAOOL. Ubicación de polos y ceros	73
Figura 5.14 FDATAOOL. Coeficientes del filtro. Ubicación de polos y ceros.	74
Figura 5.15 FDATAOOL. Reporte del filtro.	75

Figura 5.16 FDATOOL. Respuesta en magnitud estimada.	76
Figura 5.17 FDATOOL. Espectro de potencia del ruido redondeado	77
RESUMEN GENERAL DE TRABAJO DE GRADO	

RESUMEN GENERAL DE TRABAJO DE GRADO

TITULO: EFECTO DE LA PALABRA FINITA EN LA IMPLEMENTACION DE FILTRO DIGITAL

AUTOR(ES): DIANA PATRICIA TOSCANO QUINTERO
CARLOS ANDRES PEREZ PAEZ

FACULTAD: Facultad de Ingeniería Electrónica

DIRECTOR(A): EDGAR BARRIOS UREÑA

RESUMEN

Los filtros digitales pueden presentar problemas en su aplicación debido a efectos de la aritmética finita (longitud finita de las palabras digitales en los procesadores y en las memorias) que producen ruido a la salida del filtro, desplazamiento de los valores de los parámetros del filtro, oscilaciones ante algunos tipos de entrada, inestabilidad y oscilaciones de overflow. El número de bits con que se opera produce errores debido a truncamientos o redondeos que generan los problemas mencionados. Los problemas encontrados en la implementación de la palabra finita en filtros digitales son, en buena medida, derivados del uso empírico que se hace de ellos debido al poco manejo que se tiene del tema en nuestro medio; estos problemas se pueden reducir con la profundización del análisis y la investigación del fenómeno. Se adquirió un conocimiento más profundo del tema y se busca difundirlo en la comunidad académica. El objetivo de esta tesis fue la de estudiar los efectos de la longitud finita de la palabra y los métodos para atenuarlos y aplicar esos métodos en el diseño de un filtro digital PID. Para ello se realizó la investigación de las estructuras de filtros digitales, el efecto de la palabra finita en la representación de los filtros digitales; se estudió el funcionamiento y la aplicación del toolbox de diseño de filtros y el toolbox de punto fijo de MATLAB para la implementación los métodos de reducción de los efectos de la palabra finita en un filtro digital PID.

PALABRAS CLAVES: EFECTO DE PALABRA FINITA. TOOLBOX DE DISEÑO DE FILTROS

Vº Bº DIRECTOR DE TRABAJO DE GRADO

RESUMEN GENERAL DE TRABAJO DE GRADO

TITULO: **EFFECTS OF THE WORD FINITE IN THE
IMPLEMENTATION OF DIGITAL FILTERS**

AUTOR(ES): DIANA PATRICIA TOSCANO QUINTERO
CARLOS ANDRES PEREZ PAEZ

FACULTAD: Facultad de Ingeniería Electrónica

DIRECTOR(A): EDGAR BARRIOS UREÑA

RESUMEN

The digital filters can present problems in its implementation due to the effects of finite arithmetic (finite length of the words in digital processors and memories) that produce noise to the output filter, shifting the parameters of the filter, oscillations before some types of entry, instability and fluctuations overflow. The number of bits with which operates produces errors due to rounding truncation or that generate the problems mentioned. The problems encountered in implementing the finite word in digital filters are largely derived from empirical use made of them because of the little that is handling the issue in our environment, these problems can be reduced with the deepening of analysis and research of the phenomenon. It acquired a deeper understanding of the subject and was looking for spread in the academic community. The objective of this thesis was to study the effects of finite length of the floor and methods to mitigate and implement these methods in the design of a digital filter PID. This was carried out research into the structures of digital filters, the effect of finite word in the representation of digital filters; studied the operation and implementation of the toolbox design filters and the toolbox of fixed point for MATLAB implementation methods for reducing the effects of finite word in a digital filter PID.

PALABRAS CLAVES: Effects of the word finite. Toolbox design filters.

Vº Bº DIRECTOR DE TRABAJO DE GRADO

INTRODUCCION

Un filtro digital es un algoritmo matemático expresable como una ecuación en diferencias e implementable en hardware y software. Su objetivo es dar una señal de control dependiente de la última entrada, de entradas y salidas anteriores y de los parámetros del filtro.

Al usar un filtro digital se observan una serie de factores que sirven para la calificación de los filtros; dichos factores derivan de la implementación del software y del hardware del filtro digital, tales como la complejidad computacional, la capacidad de memoria y la longitud de palabra.

Los filtros digitales pueden presentar problemas en su aplicación debido a efectos de la aritmética finita (longitud finita de las palabras digitales en los procesadores y en las memorias) que producen ruido a la salida del filtro, desplazamiento de los valores de los parámetros del filtro, oscilaciones ante algunos tipos de entrada, inestabilidad y oscilaciones de overflow. El número de bits con que se opera produce errores debido a truncamientos o redondeos que generan los problemas mencionados.

En todo filtro digital, especialmente un PID, el funcionamiento observado de su salida tiene varios problemas que no siempre son tomados en cuenta en el diseño del filtro; estos problemas producen errores que, aunque insignificantes, influyen en el comportamiento del filtro; por esto se realizó este estudio que busca mostrar métodos para corregir estos problemas en el filtro digital PID.

Los problemas encontrados en la implementación de la palabra finita en filtros digitales son, en buena medida, derivados del uso empírico que se hace de ellos debido al poco manejo que se tiene del tema en nuestro medio; estos problemas se pueden reducir con la profundización del análisis y la investigación del fenómeno. Se adquirió un conocimiento más profundo del tema y se busca difundirlo en la comunidad académica.

El objetivo de esta tesis fue la de estudiar los efectos de la longitud finita de la palabra y los métodos para atenuarlos y aplicar esos métodos en el diseño de un filtro digital PID. Para ello se realizó la investigación de las estructuras de filtros digitales, el efecto de la palabra finita en la representación de los filtros digitales; se estudió el funcionamiento y la aplicación del toolbox de diseño de filtros y el toolbox de punto fijo de MATLAB para la implementación los métodos de reducción de los efectos de la palabra finita en un filtro digital PID.

1. FILTRO DIGITAL

1.1 CONCEPTO GENERAL

El filtro es un sistema que realiza el proceso de discretización de una señal de entrada y produce una salida dependiente de la entrada y de parámetros de construcción. Tiene como entrada una señal analógica o digital y a su salida es otra señal analógica o digital, pudiendo cambiar la amplitud, la frecuencia o la fase, dependiendo de las características del filtro. Los filtros digitales se usan para desarrollar controladores en control digital, el tratamiento digital de la imagen o para tratamiento del sonido digital.



Figura 1.1 Sistema de filtrado

La denominación de filtro digital proviene de las características de funcionamiento interno más que del tipo de señal a filtrar; así se podría llamarse filtro digital a un filtro que realiza el procesamiento de señales digitales o que procesa señales analógicas.

El valor de la entrada actual y algunas anteriores (previamente almacenadas) son multiplicados por coeficientes. También se puede tomar valores de la salida de instantes anteriores y multiplicarlos, también, por coeficientes. Los resultados de todas estas multiplicaciones son sumadas, dando una salida para el instante actual. Esto implica que internamente tanto la salida como la entrada del filtro sean digitales por lo que puede ser necesaria una conversión analógico/digital o digital/analógico para el uso de filtros digitales en señales analógicas.

1.2 REPRESENTACION NUMERICA DE LOS FILTROS

La representación numérica en los filtros digitales se hace aplicando el código binario que usa un conjunto de bits para representar valores enteros y fraccionarios, ya sean valores positivos o negativos.

Los números binarios es un sistema de numeración, donde las cifras son representadas por 0 (cero) y 1(unos); por ejemplo la representación binaria de una palabra de 3 bits se muestra en la figura 1.2

REPRESENTACION DECIMAL	REPRESENTACION BINARIA
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Figura 1.2 Ejemplo de palabra binaria de 3 bits

El número de bits utilizado define la precisión de los números representados haciendo que la diferencia entre el número representado y el valor que lo representa sea menor mientras mayor sea el número de bits. A continuación se presenta un ejemplo de una palabra binaria

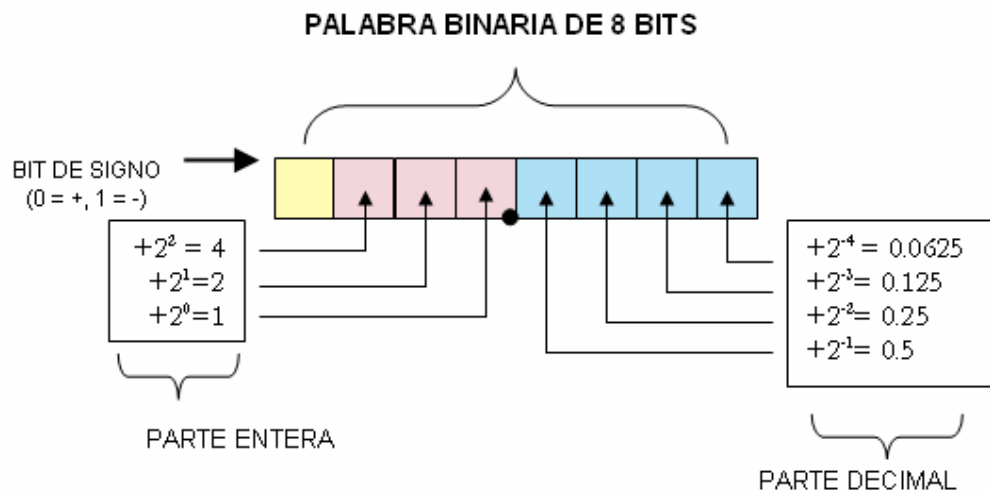


Figura 1.3 Representación de una palabra binaria de 8 bits.

Cuando se aplica diagrama anterior en forma directa se llama de punto fijo, refiriéndose al hecho que el punto no se mueve, es decir, cada bit siempre representa el mismo valor. La aritmética de punto fijo es usada generalmente cuando se tienen características importantes de diseño, tales como el costo económico del hardware, la velocidad del procesador y la complejidad del diseño a realizar.

Además de la representación del sistema numérico en punto fijo, una representación importante y útil en los filtros digitales es la representación numérica de complemento a dos, donde a la palabra se le aplica complemento a 1 y se le adiciona al bit menos significativo un 1 binario, sin tener en cuenta el overflow producido por la operación.

1.3 TIPOS DE FILTROS

Hay varios tipos de filtros y varias clasificaciones de ellos. Los filtros se clasifican de acuerdo con la parte del espectro que dejan pasar y la atenuación que tengan; están los filtros pasa bajo (*lowpass*), filtros pasa alto (*highpass*) y filtros pasa banda (*passband*). Según su orden se tiene filtros de primer y segundo orden. De acuerdo con el tipo de respuesta ante entrada unitaria están los filtros FIR (*Finite Impulse Response*), IIR (*Infinite Impulse Response*), TIIR (*Truncated Infinite Impulse Response*). De acuerdo con la estructura con que se implementa se encuentran los filtros *Lattice*, en cascada, en paralelo.¹

1.4 FILTROS FIR

La sigla FIR significa *Finite Impulse Response* (Respuesta Finita al Impulso); la entrada del filtro FIR es una señal impulso (delta de *Kronecker*) cuya salida es un valor limitado de combinaciones lineales de los valores anteriores y presentes de la señal de entrada. Este filtro es un filtro no recursivo, es decir su valor de salida se calcula a partir de los valores de entrada. La ecuación general de un filtro FIR es de la forma:

$$\begin{aligned} y[n] &= b_0 x[n] + b_1 x[n-1] + \dots + b_{M-1} x[n-M+1] \\ &= \sum_{k=0}^{M-1} b_k x[n-k] \end{aligned} \quad (1.1)$$

Donde:

- M = Longitud del filtro.
- $x[n]$ = Señal de entrada del filtro.
- $y[n]$ = Señal de salida del filtro.
- b_k = Coeficientes del filtro.
- $n-1 \dots n-M+1$ = Retardos en el tiempo de la señal ya sea de entrada o de salida.

La estructura fundamental de los filtros FIR se observa en la figura 1.4

¹ http://es.wikipedia.org/wiki/Filtro_digital

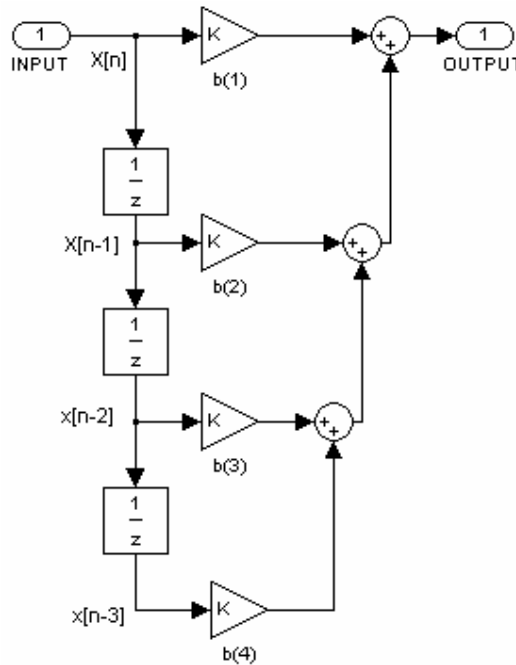


Figura 1.4 Estructura del Filtro FIR.

Ing. Lino Evgueni Coria Mendoza. Instituto tecnológico y de Estudios Superiores de Occidente. México.

Su función de transferencia es de la forma:

$$H(z^{-1}) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_M z^{-M}}{1 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_N z^{-N}} \quad (1.2)$$

La mayoría de los filtros FIR tiene los polos del sistema en el origen para hacer estable al sistema; los ceros se encuentran representados de manera equitativa en forma de pares, para establecer la linealidad de la fase, es decir su respuesta tiene un retraso en el tiempo constante; debido a dicha propiedad de respuesta lo filtros FIR tienen memoria finita.

Los filtros FIR transmiten todas las frecuencias de la señal de entrada con el mismo tiempo de retardo en el tiempo; esta característica no permite que haya ninguna etapa de distorsión en la señal de entrada. Esta propiedad es ventajosa en la transmisión de señales digitales.

1.5 FILTROS IIR

El significado de IIR es *Infinite Impulse Response* (Respuesta Infinita al Impulso). Si la entrada del filtros IIR es una señal impulso (delta de *Kronecker*) su salida nunca volverá a un estado de reposo específico; para obtener los valores de salida del filtro IIR se necesita los valores anteriores y actuales de la señal de

entrada y los valores anteriores de salida del filtro (estos son almacenados) son realimentados a la entrada de la señal.

La ecuación general de un filtro IIR es de la forma:

$$y[n] = b_0x[n] + b_1x[n-1] + \dots + b_{M-1}x[n-M+1]$$

$$= \sum_{k=0}^{M-1} b_k x[n-k] \quad (1.3)$$

Donde:

- M = Longitud del filtro.
- $x[n]$ = Señal de entrada del filtro.
- $y[n]$ = Señal de salida del filtro.
- b_k = Coeficientes del filtro.
- $n-1 \dots n-M+1$ = Retardos en el tiempo de la señal ya sea de entrada o de salida.

La estructura fundamental de los filtros FIR se muestra a continuación:

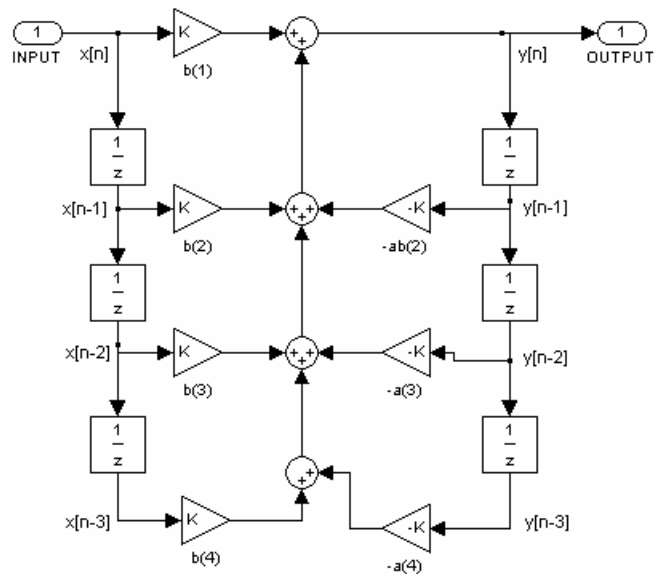


Figura 1.5 Estructura de Filtro IIR.

Ing. Lino Evgueni Coria Mendoza. Instituto tecnológico y de Estudios Superiores de Occidente. México.

La función de transferencia del filtro IIR es:

$$H(z^{-1}) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_M z^{-M}}{1 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_N z^{-N}} \quad (1.4)$$

El término IIR en un filtro digital no es muy exacto ya que las respuestas reales del impulso de casi todos los filtros IIR se reducen generalmente a cero después de un tiempo.

Los filtros IIR son filtros recursivos, es decir, su salida es calculada a partir de los valores de entrada y además emplea como entrada valores anteriores de su salida, por lo tanto requiere menos coeficientes que el procesador evalúa; los filtros recursivos son de menor orden que los filtros no recursivos. Los filtros IIR, como los FIR, tienen memoria, pero a diferencia de ellos, tienen memoria infinita. En los filtros IIR el orden de estimación de los filtros IIR es exacta ya que derivan de las propiedades matemáticas clásicas para el diseño de filtros digitales; estas formulas son muy útiles para obtener el orden necesarios del filtro IIR para que se puedan cumplir las especificaciones de diseño deseado. Al diseñar los filtros IIR se realizan los cálculos necesarios para que el sistema sea estable, pero es muy común que al aplicar este filtro se vuelvan inestables.

1.6 FILTROS ADAPTATIVOS

Los filtros adaptativos son dispositivos utilizados para modelar la relación entre señales en tiempo real de manera iterativa; estos filtros se caracterizan por tener coeficientes que se adaptan a las condiciones de la señal de entrada, esto supone que se debe trabajar con señales aleatorias.

Los coeficientes de estos filtros son calculados durante el diseño y su valor cambia en cada iteración. La ecuación general de un filtro adaptativo es:

$$y(n) = \frac{b_0 x(n) + b_1 x(n-1) + \dots + b_M x(n-M)}{-a_1 y(n-1) - \dots - a_N y(n-N)} \quad (1.5)$$

Un filtro adaptativo tiene dos procesos básicos; el primero es el de filtrado que consiste en obtener un valor de salida con respecto al valor de entrada y el segundo consiste en ajustar los coeficientes del filtro de acuerdo al algoritmo específico. Estos dos procesos trabajan de manera iterativa cuando se aplica el filtro.²

² www.desi.iteso.mx/sys/dsp/filtros_digitales.ppt

2. ESTRUCTURA DE LOS FILTROS DIGITALES

Una forma sencilla de diseñar filtros digitales es seleccionar primero la estructura que va a representar dicho filtro. Entre las estructuras de los filtros digitales se encuentran las estructuras de forma directa, en las cuales los coeficientes del filtro aparecen en el diagrama de bloques en forma de multiplicadores (1D, 2D, 3D, 4D), las estructuras de acoplamiento (1X y 2X) y los módulos de primer y segundo orden. La Tabla 2.1 muestra las características más importantes que tienen todas las estructuras directas.

CARACTERISTICAS	ESTRUCTURAS			
	1D	2D	3D	4D
Elementos del retardo de tiempo	n	n	2n	2n
Multiplicadores	2n+1	2n+1	2n+1	2n+1
uniones de suma	2	n+1	1	.2n
Puntos de distribución de señal	n-1	2	2n	1

TABLA 2.1 PROPIEDADES DE LAS ESTRUCTURAS DIRECTAS

Ing. Lino Evgueni Coria Mendoza. Instituto tecnológico y de Estudios Superiores de Occidente. México.

Se puede observar que en las estructuras 1D y 2D conservan elementos de retardo y en las estructuras 1D y 3D se conservan las uniones. En todas las estructuras directas se observa cierta sensibilidad de los coeficientes al aumentar los valores de n, es decir si existen cambios grandes de n, los coeficientes a_i y b_i tiene cambios pequeños que a su vez producen cambios grandes de los polos y los ceros de la función de transferencia.

2.1 ESTRUCTURA 1D

Las estructuras 1D son de la forma:

$$D(z) = \frac{\sum_{i=0}^n a_i z^{-i}}{\sum_{i=0}^n b_i z^{-i}} = \frac{Y(z)}{X(z)} \quad (2.1)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.466

Cuando se realizan la representación gráfica de las estructuras 1D los retardos (z^{-1}) son representados por una caja rectangular, los multiplicadores por flechas, los sumadores por medio de elipses y los punto de bifurcación por puntos en negrilla.

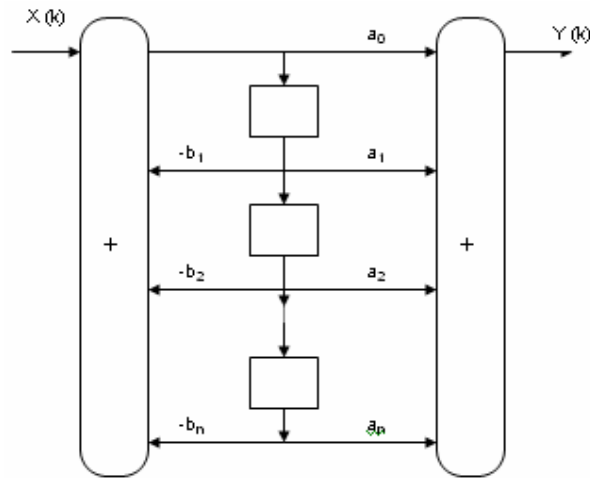


Figura 2.1 Estructura 1D

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.468

Las estructuras 1D son conocidas comúnmente como estructuras canónicas porque tiene solamente n elementos de atraso.

2.2 ESTRUCTURA 2D

Las estructuras 2D son la transposición de las estructuras 1D. También son estructuras canónicas. Su ecuación general es de la forma:

$$\begin{aligned}
 p_i(k) &= p_{i+1}(k-1) + a_i x(k) - b_i y(k), \quad i = 1, n-1 \\
 p_n(k) &= a_n x(k) - b_n y(k) \\
 y(k) &= a_0 x(k) + p_1(k-1)
 \end{aligned}
 \tag{2.2}$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.469

La estructura 2D es de la forma canónica, con un valor de coeficiente para $a_0=0$.

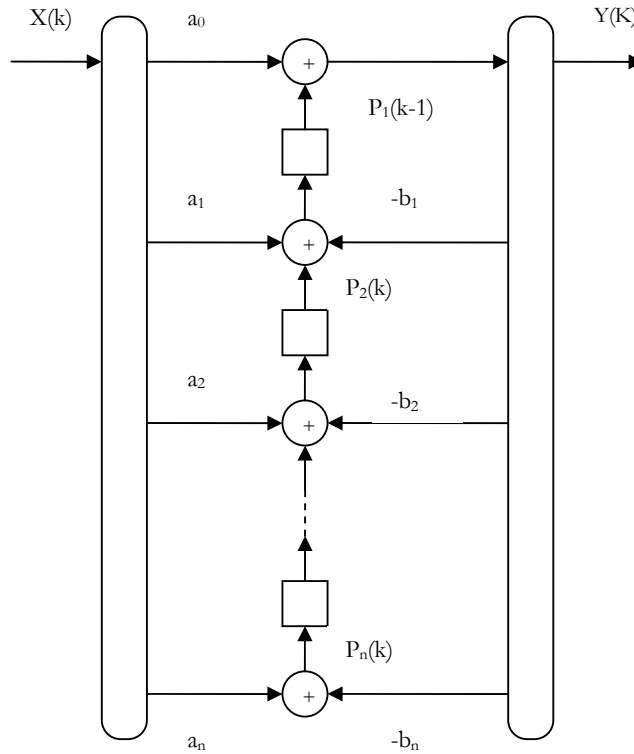


Figura 2.2 estructura 2D

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.468

2.3 ESTRUCTURA 3D

Su ecuación general es:

$$D(z) = \frac{\sum_{i=0}^n a_i z^{-i}}{\sum_{i=0}^n b_i z^{-i}} = \frac{Y(z)}{X(z)} \quad (2.3)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.469

El diagrama de bloques del filtro 3D es:

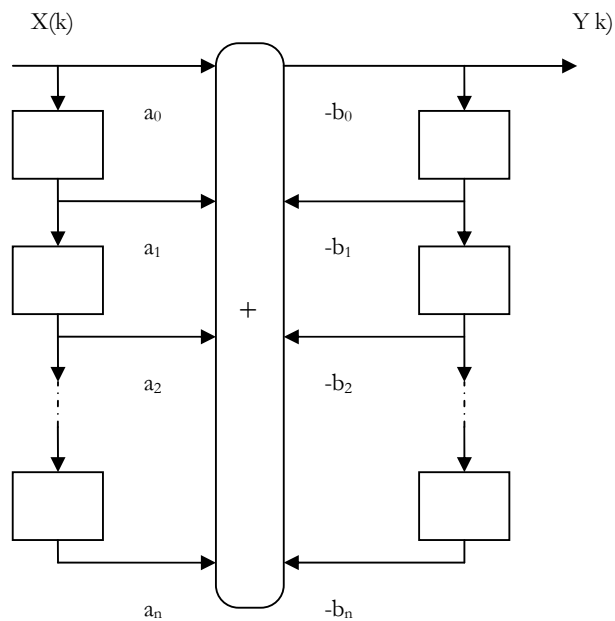


Figura 2.3 estructura 3D

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.468

La estructura 3D sólo tiene uniones de sumas pero tienen $2n$ elementos de retardo.

2.4 ESTRUCTURA 4D

La estructura 4D es el resultado de la transposición de la estructura 3D. Su ecuación general es:

$$\begin{aligned}
 r_o(k) &= x(k) + r_i(k-1) \\
 q_n(k) &= a_n r_o(k) \\
 r_n(k) &= -b_n r_o(k) \\
 q_i(k) &= a_i r_o(k) + q_{i+1}(k-1), \quad i = 1, n-1 \\
 r_i(k) &= -b_i r_o(k) + r_{i+1}(k-1) \\
 y(k) &= a_0 r_o(k) + q_1(k-1)
 \end{aligned} \tag{2.4}$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.470

El diagrama de bloques de la estructura 4D es:

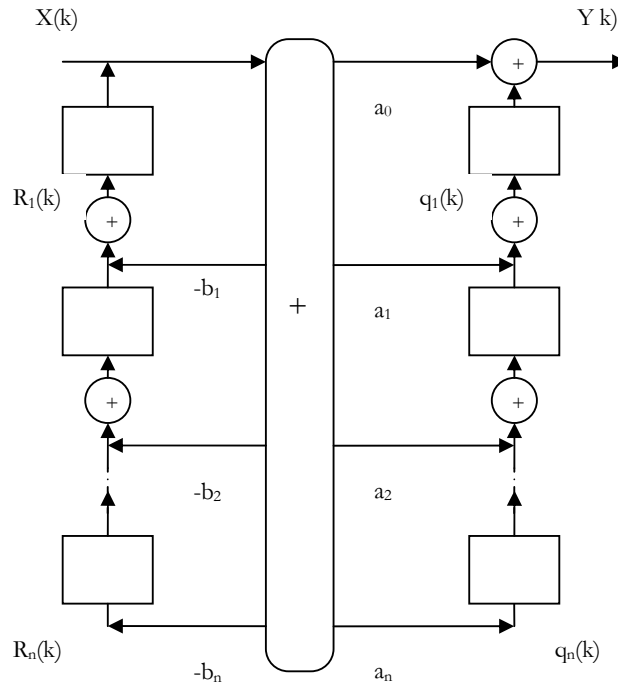


Figura 2.4 Estructura 4D

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.468

Esta estructura tiene un solo punto de distribución de señal pero tiene $2n$ ecuaciones diferencia.

2.5 ESTRUCTURA DE ACOPLAMIENTO 1X y 2X

La estructura de acoplamiento cruzado (1X) se usa para los casos de polos complejos y es una estructura de la forma canónica. La estructura de acople 2X es la forma transpuesta de la estructura de acople 1X.

Las ecuaciones diferencia de las estructuras de acople de 1X y 2X se pueden observar en (2.5) y (2.6) respectivamente. Al igual que en las estructuras directas las estructuras de acople también tienen diagramas de bloques característicos para cada una de sus estructuras; en las figuras 2.5 y 2.6 se muestra los diagramas de bloques de las estructuras 1X y 2X respectivamente.

$$\begin{aligned}
y(k) &= a_0 x(k) + s_2 x(k) \\
s_1(k) &= g_1 s_1(k-1) - g_2 s_2(k-1) + g_3 x(k) \\
s_2(k) &= g_1 s_2(k-1) + g_2 s_1(k-1) + g_4 x(k)
\end{aligned}
\tag{2.5}$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.474

$$\begin{aligned}
y(k) &= a_0 x(k) + g_3 t_1(k-1) + g_4 t_2(k-1) \\
t_1(k) &= g_1 t_1(k-1) + g_2 t_2(k-1) \\
t_2(k) &= x(k) + g_1 t_2(k-1) - g_2 t_1(k-1)
\end{aligned}
\tag{2.6}$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.474

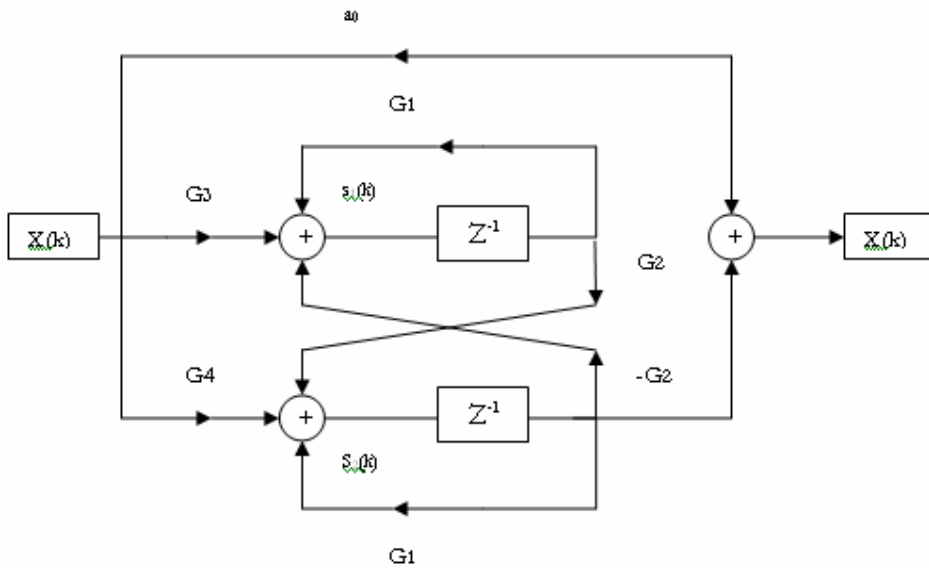


Figura 2.5 Estructura 1X.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.473

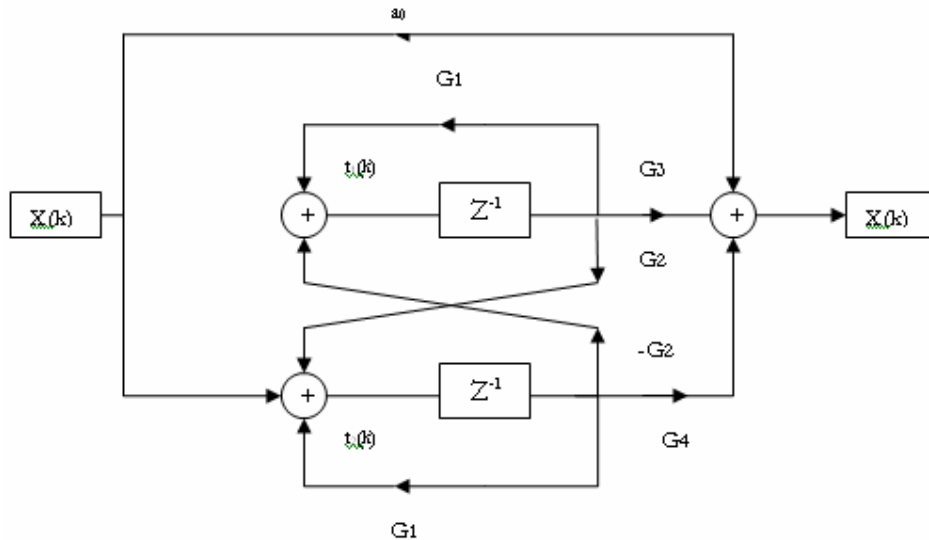


Figura 2.6 Estructura 1X.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.474

3. REALIZACION DE FILTROS

3.1 FILTROS EN CASCADA

Los filtros en cascada son aquellos en los cuales la función de transferencia es el resultado del producto de las funciones de transferencia individuales. Los filtros en cascada son utilizados para solucionar el problema de sensibilidad ocurrido con los coeficientes del filtro. Para ello se descompone $D(z)^3$ en factores de la forma:

$$D(z) = \frac{\prod_{i=1}^m (a_{i0} + a_{i1}z^{-1} + a_{i2}z^{-2})}{\prod_{i=1}^m (1 + a_{i3}z^{-1} + a_{i4}z^{-2})} \quad (3.1)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.475

Donde, m (entero) $\leq n/2$.

La figura 3.1 muestra la representación de una estructura directa de segundo orden, esta representación se utiliza tanto para estructuras directas como para estructuras de acoplamiento.

³ $D(z)$ es la función de transferencia del filtro digital.

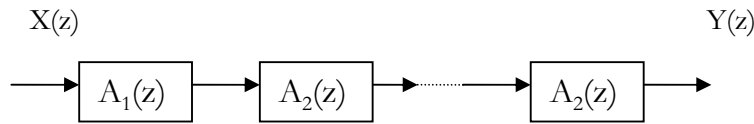


Figura 3.1 Conexión en Cascada.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.476

La tabla 3.1 muestra las características que tienen las estructuras 1D, 2D, 3D y 4D cuando se aplican a un módulo en cascada. Se puede observar que las estructuras 3D y 4D ahorran $n-2$ elementos, también se puede observar que se requieren $m-1$ bifurcaciones y puntos de mas.

PROPIEDADES	ESTRUCTURAS DIRECTAS			
	1D	2D	3D	4D
Elementos de retardo	$2m$ (n)	$2m$ (n)	$2m+2$ ($2n-(n-2)$)	$2m+2$ ($2n-(n-2)$)
Multiplicadores	$5m$ ($n-n+m$)	$5m$ ($n-n+m$)	$5m$ ($n-n+m$)	$5m$ ($n-n+m$)
Uniones de suma	$m-1$ ($2-m-1$)	$3m$ ($n+m$)	m	$3m+1$ ($2n-(m-1)$)
Puntos de distribución de señal	$3m$ ($n+m$)	$m+1$ ($2-m-1$)	$3m+1$ ($2n-(m-1)$)	m

Tabla 3.1 Propiedades de las estructuras directas en cascadas⁴

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.478

3.2 FILTROS EN PALALELO

Otra forma común de realizar las conexiones de un filtro digital es en paralelo, cuyo resultado es la suma de las funciones de transferencia individuales. Para este método se debe realizar una serie de procesos matemáticos en la función de transferencia; lo primero se realiza la factorización en el denominador y se luego la realización de fracciones parciales.⁵

$$D(z) = \beta_0 + \sum_{i=1}^m B_i(z) \quad (3.2)$$

$$B_i(z) = \frac{\beta_{i1}z^{-1} + \beta_{i2}z^{-2}}{1 + \beta_{i3}z^{-1} + \beta_{i4}z^{-2}} \quad (3.3)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.478,p.483.

⁴ m = numero entero $\leq n/2$.

⁵ Por medio de las fracciones parciales se realiza en el caso de que hayan polos diferentes.

En la figura 3.2 se observa la estructura general de la conexión en paralelo.

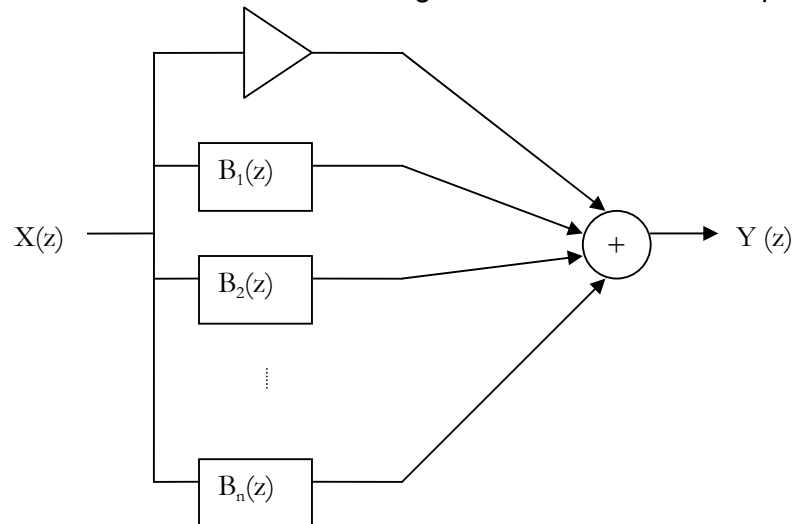


Figura 3.2 Conexión en Paralelo.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.478

La Tabla 3.2 contiene las propiedades de la conexión en paralelo, y con ella se puede hacer una comparación con la conexión en cascada y se puede decir que la estructura 3D y 4D en paralelo tiene $n-2$ elementos de retardo menos que la conexión en cascada y el número de multiplicadores en ambas conexiones son iguales; en la estructura 1D la conexión en paralelo requiere una unión de suma $m-1$ más que en la conexión en cascada; en la estructura 2D la conexión en paralelo necesita un punto de bifurcación $m-1$ más que en la conexión en cascada.

PROPIEDADES	ESTRUCTURAS DIRECTAS			
	1D	2D	3D	4D
Elementos de retardo	$2m$ (n)	$2m$ (n)	$2m+2$ ($2n-(n-2)$)	$2m+2$ ($2n-(n-2)$)
Multiplicadores	$4m+1$ ($2n+1$)	$4m+1$ ($2n+1$)	$4m+1$ ($2n+1$)	$4m+1$ ($2n+1$)
Uniones de suma	$m+1$ ($2+m-1$)	$2m+1$ ($n+1$)	$m+1$ ($1+m$)	$2m+3$ ($2n-(n-3)$)
Puntos de distribución de señal	$2m+1$ ($n+1$)	$m+1$ ($2+m-1$)	$2m+3$ ($2n-(n-3)$)	$m+1$ ($1+m$)

Tabla 3.2 Propiedades de las estructuras directas en paralelo⁶

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.478

⁶ $m = \text{numero entero} \leq n/2$.

3.3 FILTROS EN ESCALERA

Otro tipo de conexión para los filtros digitales es la de escalera que, al igual que los otros métodos, mejora la sensibilidad de los coeficientes de las estructuras directas de filtro. La conexión en escalera tiene como característica que los valores de los coeficientes para a_n y b_n son diferentes a cero de tal forma que se utiliza para filtros de orden superior como lo muestra la ecuación (3.5) y a su vez se puede expandir en fracciones como se observa en la ecuación (3.6)

$$D(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}}{1 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_n z^{-n}} \quad (3.4)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.485

$$D(z) = A_0 + \frac{1}{b_1 z^{-1} + \frac{1}{A_1 + \frac{1}{b_2 z^{-1} + \frac{1}{\ddots + \frac{1}{b_n z^{-n} + \frac{1}{A_n}}}}}} \quad (3.5)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.485

4. EFECTO DE LA PALABRA FINITA

En los sistemas digitales en los que se trabaja con variables discretas estas, al igual que los coeficientes, son números reales; los valores de las variables para estos sistemas se mantienen en el rango $(-\infty, \infty)$ de magnitud discreta

4.1 SISTEMA NUMERICO DE PUNTO FIJO

El sistema numérico de punto fijo afecta el comportamiento de cualquier filtro, donde los coeficientes y las variables del filtro pueden ser representadas con exactitud de acuerdo a las conveniencias de la cuantización que se usa. La exactitud de la representación puede medirse por el error:

$$e = Q(x) - x \quad (4.1)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.526

Suponiendo que x y $Q(x)$ están en el rango siguiente:

$$0 \leq |x| \leq 1 \quad (4.2)$$

$$0 \leq |x'| \leq C \quad (4.3)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.526

Por tal razón:

$$0 \leq |Q(x')| \leq C \quad (4.4)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.526

Para un $C > 1$, existe un valor de L , tal que:

$$Q(x) = 2^L Q(x') \quad (4.5)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.526

Por lo tanto:

$$0 \leq |Q(x)| \leq 1 \quad (4.6)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.526

4.1.1 SISTEMA NUMERICO SIGNO Y MAGNITUD

Este número de signo-magnitud del sistema representa los coeficientes del filtro digital y variables del signo en general. Su ecuación general es:

$$Q^b(x) = (s.m_1 | m_2 | m \cdots m_b) \quad (4.7)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.526

Donde:

- $Q^b(x)$: Cuantización de x
- S : bit de signo.
Para $S=0$ x es positivo.
Para $S=1$ x es negativo
- m_i : bits de magnitud.

4.1.1.1 CUANTIZACION POR TRUNCAMIENTO

En este caso $|x|$ es un número real positivo y es convertido a una fracción binaria, esto es, se trunca a b bits:

$$|x| = \underbrace{(m_1 | m_2 | m_b m_{b+1} \dots)_2}_{|Q_t^b(x)|} \quad (4.8)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.527

Donde:

- t : indica el truncamiento
- b : número de bits de cuantización de x

Por lo tanto el error, para $x \geq 0$, es:

$$e_t = |Q_t^b(x)| - |x| = -2^{-b} (m_{b+1} m_{b+2} \dots)_2 \quad (4.9)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.527

Donde:

$$0 \leq (m_{b+1} m_{b+2} \dots)_2 \leq 1 \quad (4.10)$$

$$0 \geq e_t > -2^{-b} \quad (4.11)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.527

Por lo tanto el error, para $x < 0$, es:

$$e_t = -\left[|Q_t^b(x)| - |x|\right] \quad (4.12)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.527

Donde:

$$0 \leq e_t < 2^{-b} \quad (4.13)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.527

La característica de la cuantización por truncamiento se observa en la figura 4.1

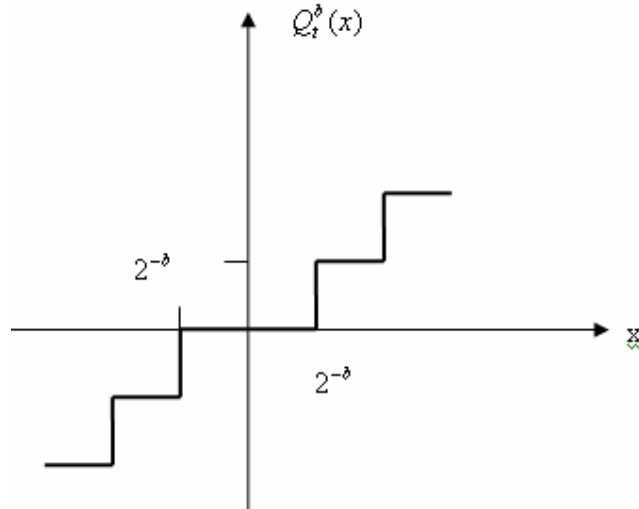


Figura 4.1 Característica de cuantización para sistema numérico de signo y magnitud.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.528

La varianza del ruido puede aproximarse por:

$$\sigma_{et}^2 \left\{ E[e_t]^2 - E^2[e_t] \right\}_{x>0} + \left\{ E[e_t]^2 - E^2[e_t] \right\}_{x<0} \quad (4.14)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.528

Donde:

$$E[e_t]^2_{x \geq 0} = 2^{-2b} / 6 \quad (4.15)$$

$$E^2[e_t]_{x \geq 0} = -2^{-b} / 4 \quad (4.16)$$

$$E[e_t]^2_{x < 0} = 2^{-2b} / 6 \quad (4.17)$$

$$E^2[e_t]_{x < 0} = 2^{-b} / 4 \quad (4.18)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.529

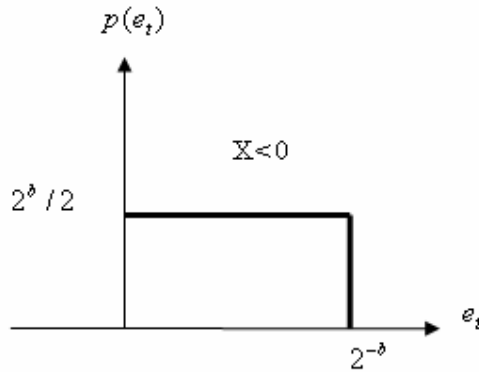


Figura 4.2 Error de probabilidad de la cuantización de truncamiento para un sistema signo-magnitud.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.529

4.1.1.2 CUANTIZACION REDONDEO

Esta Cuantización toma un valor $|x|$ que puede escribirse como una serie infinita en el sistema numérico de complemento a 2.

$$|x| = (.n_1 \ n_2 \ n_b \ n_{b+1} \ \dots)_2 \quad (4.19)$$

$$Q_r^b(x) = (s m_1 \ m_2 \ m_b \ m_{b+1} \ \dots)_2 \quad (4.20)$$

$$e_r = Q_r^b(x) - x \quad (4.21)$$

$$0 \leq (.n_1 \ n_2 \ n_b \ n_{b+1} \ \dots)_2 < 1 \quad (4.22)$$

$$0 \geq e_r - 2^{-b-1} > -2^{-b} \quad (4.23)$$

$$e_r - 2^{-b-1} = 2^{-b} [(.n_{b+1} \ n_{b+2} \ n_{b+3} \ n_{b+4} \ \dots)_2 + 2^{-1}] \quad (4.24)$$

$$\frac{-2^{-b}}{2} \geq e_r > \frac{2^{-b}}{2} \quad (4.25)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.530

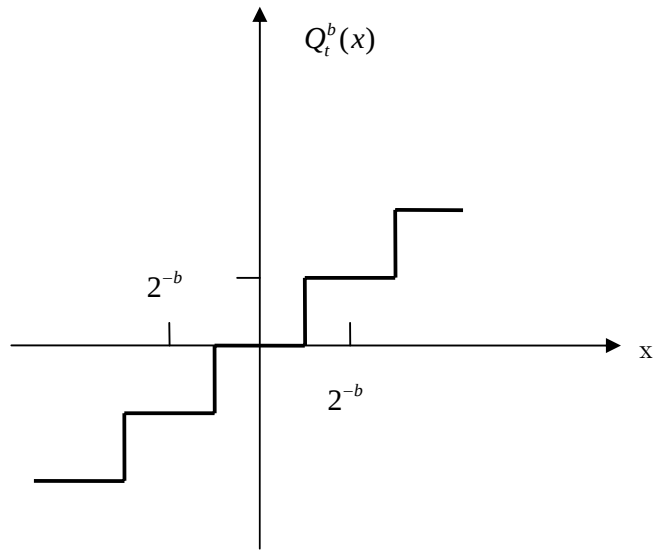


Figura 4.3 Característica de cuantización por redondeo para sistema numérico de signo y magnitud.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.528

La varianza del ruido puede aproximarse por:

$$\sigma_{er}^2 = E[e_r^2] - E^2[e_r] \quad (4.26)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.531

Donde:

$$E[e_r^2] = \frac{2^{-2b}}{12} \quad (4.27)$$

$$E^2[e_r] = 0 \quad (4.28)$$

$$\begin{aligned} \sigma_{er}^2 &= \frac{2^{-2b}}{12} - 0 \\ &= \frac{2^{-2b}}{12} \end{aligned} \quad (4.29)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.531

Por lo tanto:

$$\sigma_{et}^2 = \frac{5}{2} \sigma_{er}^2 \quad (4.30)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.531

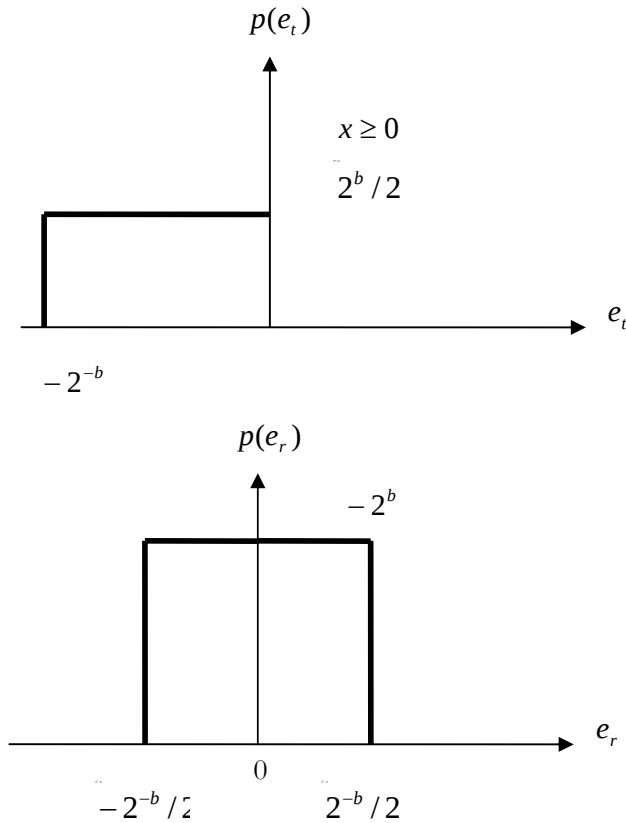


Figura 4.4 Error de probabilidad de la cuantización de redondeo para un sistema signo-magnitud.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.529

4.1.1.3 CUANTIZACION LSB-1

Esta cuantización establece el bit menos significativo en 1. Para todos los números de $Q_L^b(x)$:

$$Q_L^b(x) = (sm_1 m_2 m_b m_{b+1} \dots)_{2SMMS} \quad (4.31)$$

$$e_L = 2^{-b+1} (1/2 - (m_b m_{b+1} \dots)) \quad (4.32)$$

$$0 \leq (. m_b m_{b+1} \dots)_2 < 1 \quad (4.33)$$

$$-1/2 < 1/2 - (m_b m_{b+1} \dots)_2 \leq 1/2 \quad (4.34)$$

$$-2^{-b} < e_L \leq -2^{-b} \quad (4.35)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.531, P.532

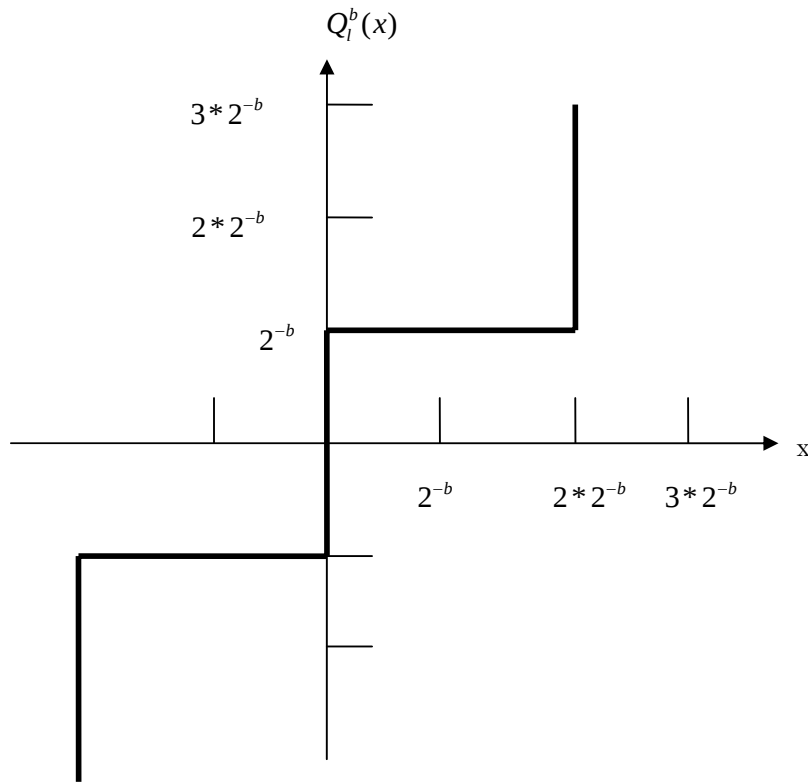


Figura 4.5 Característica de cuantización por LSB-1 para sistema numérico de signo y magnitud.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.528

La variación del ruido puede aproximarse por:

$$\sigma_{et}^2 = E[e_L^2] - E^2[e_L] \quad (4.36)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.531, P.532

Donde:

$$E[e_L^2] = 2^{-2b} / 3 \quad (4.37)$$

$$E^2[e_L] = 0 \quad (4.38)$$

$$\sigma_{et}^2 = 2^{-2b} / 3 - 0 = 2^{-2b} / 3 \quad (4.39)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.531, P.532

Por lo tanto:

$$\sigma_{eL}^2 = 4 \sigma_{er} \quad (4.40)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.531, P.532

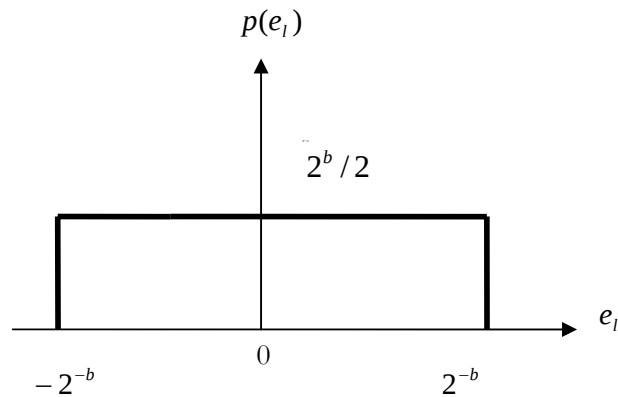


Figura 4.6 Error de probabilidad de la cuantización de LSB-1 para un sistema signo-magnitud.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.529

4.1.1.4. RELACION DE LOS TRES TIPOS DE CUANTIZACION

$$Q_r^b(x) = Q_t^b \left(x + (1-2s) \frac{2^{-b}}{2} \right) \quad (4.41)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.531, P.532

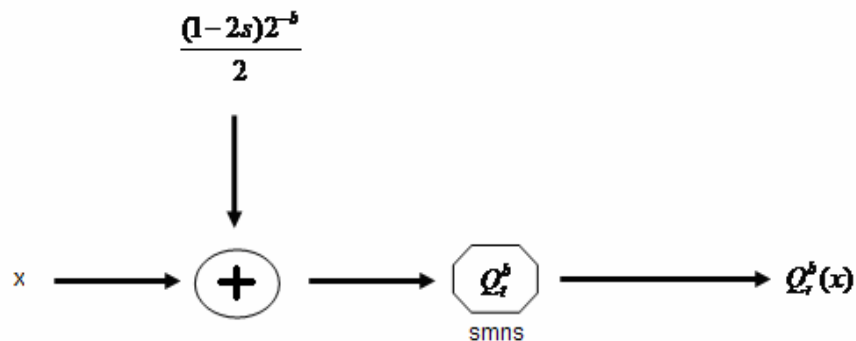


Figura 4.7 Equivalente de cuantizadores de redondeo

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.533

$$Q_b^l(x) = Q_t^{b-1}(x) + (1-2s)2^{-b} \quad (4.42)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.531, P.532

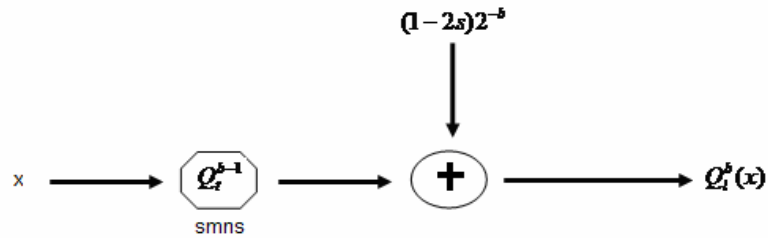


Figura 4.8 Equivalente de cuantizadores de LSB-1

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.533

4.1.1.5. RANGO DINAMICO

$$\begin{aligned} (D.R)_{2smns} &= \frac{1-2^{-b}}{2^{-b}} \\ &= -2^{-b} - 1 \end{aligned} \quad (4.43)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.531, P.533

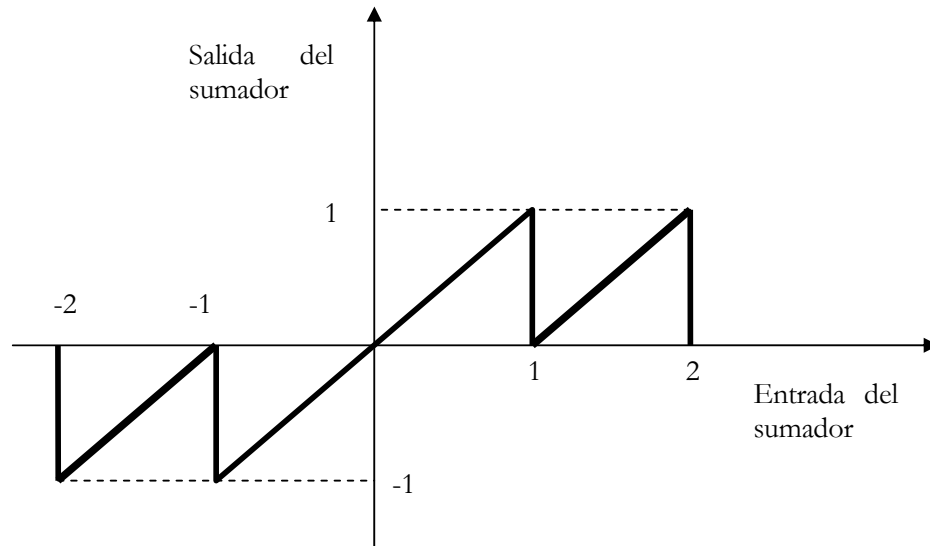


Figura 4.9 Característica de overflow para el sistema signo-magnitud

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.534

4.1.2. SISTEMA DE COMPLEMENTO A DOS

Es el sistema usado en la mayoría de las computadoras digitales y normalmente se usa para representar los filtros digitales.

$$\begin{aligned} Q^b(x) &= (0.m_1m_2\dots m_b)_{2cns}, 0 \leq x < 1 \\ Q^b(x) &= (1.n_1n_2\dots n_b)_{2cns}, -1 \leq x < 0 \end{aligned} \quad (4.44)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.531, P.534

4.1.2.1 CUANTIZACION POR TRUNCAMIENTO

$$\sigma_{et}^2 = E[e_r^2] - E^2[e_r] \quad (4.45)$$

$$E[e_t^2] = \frac{2^{-2b}}{3} \quad (4.46)$$

$$E^2[e^t] = \frac{-2^{-2b}}{2} \quad (4.47)$$

$$\sigma_{et}^2 = \frac{2^{-2b}}{3} - \left(\frac{-2^{-2b}}{2} \right) \quad (4.48)$$

$$\sigma_{et}^2 = \frac{2^{-2b}}{12} \quad (4.49)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.531, P.537

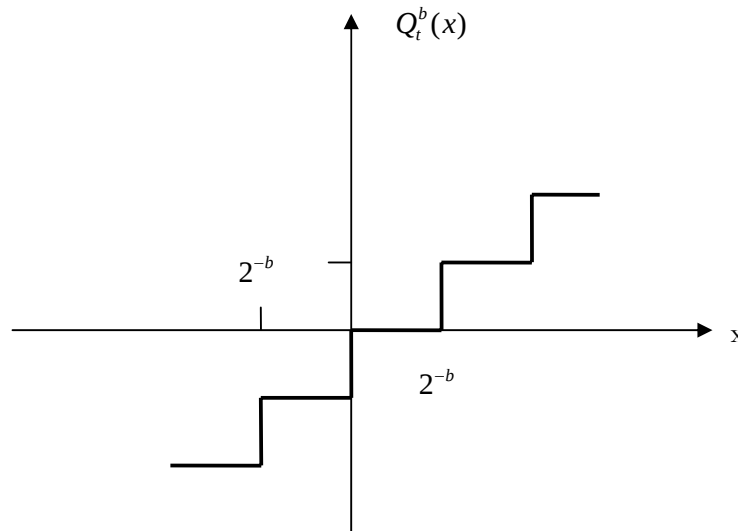


Figura 4.10 Característica de cuantización por truncamiento para el sistema de complemento a dos.

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.536

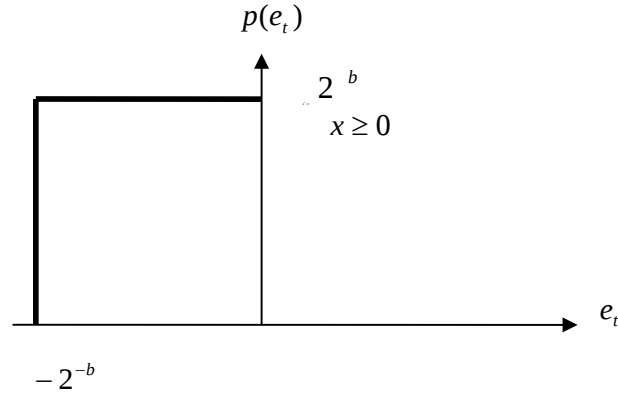


Figura 4.11 Error de probabilidad de la cuantización por truncamiento para el sistema de complemento a dos.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.537

4.1.2.2. CUANTIZACION ROUND-OFF

$$\frac{2^{-b}}{2} \geq e_r \geq \frac{-2^{-b}}{2} \quad (4.50)$$

$$\sigma_{er}^2 = \frac{2^{-2b}}{12} \quad (4.51)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.531, P.538

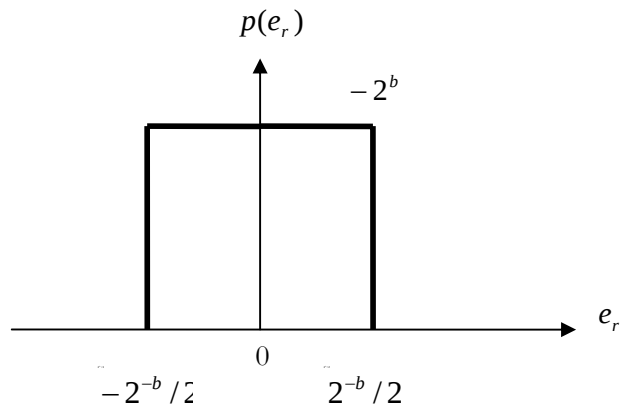


Figura 4.12 Error de probabilidad de la cuantización por redondeo para el sistema de complemento a dos.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.537

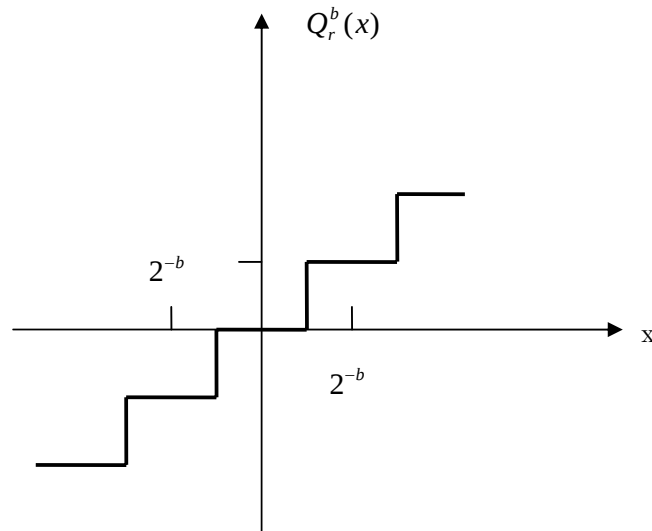


Figura 4.13 Característica de cuantización por redondeo para el sistema de complemento a dos.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.536

4.1.2.3. CUANTIZACION LSB-1

$$-2^{-b} < e_l \leq 2^{-b}, x < 0 \quad (4.52)$$

$$\sigma_{el}^2 = \frac{2^{-2b}}{3} \quad (4.53)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.531, P.539

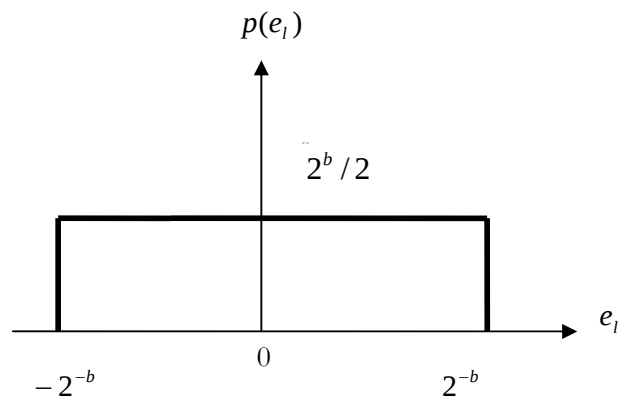


Figura 4.14 Error de probabilidad de la cuantización por lsb-1 para el sistema de complemento a dos.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.537

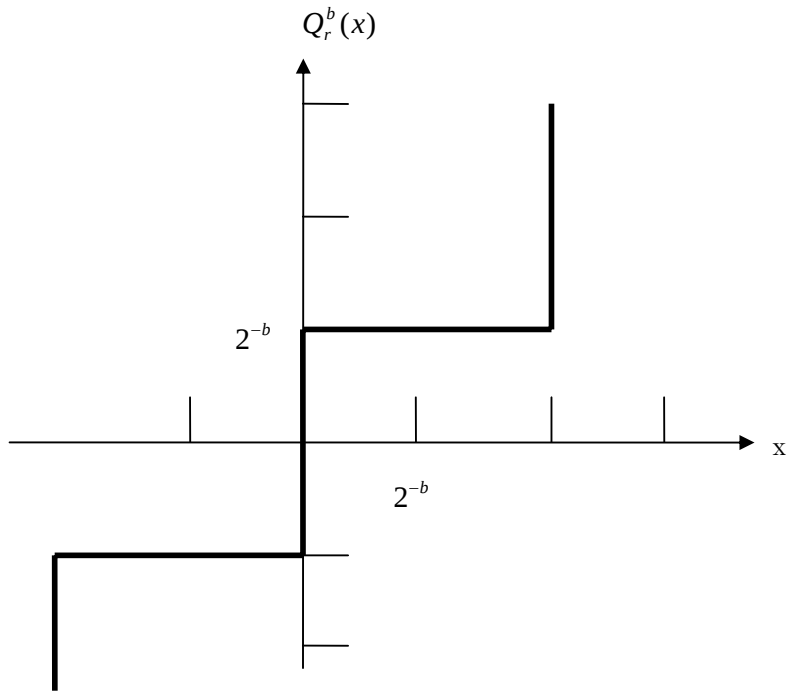


Figura 4.15 Característica de cuantización por lsb-1 para el sistema de complemento a dos.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.536

4.1.2.4 RELACION DE LOS TRES TIPOS DE CUANTIZACION

$$Q_r^b(x) = Q_t^b\left(x + \frac{2^{-b}}{2}\right) \quad (4.54)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.531, P.539

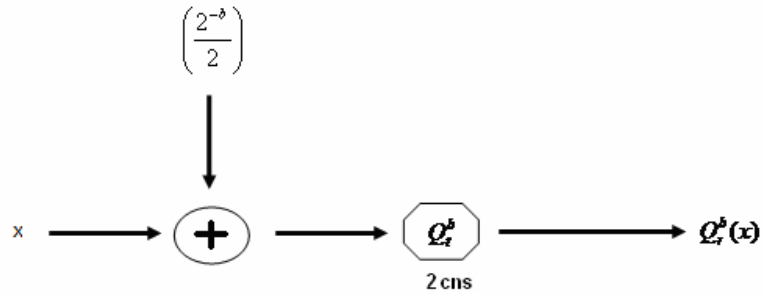


Figura 4.16 Equivalente de cuantizadores de redondeo.
 PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.539

$$Q_t^b(x) = Q_t^{b-1}(x) + 2^{-b} \quad (4.55)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.531, P.540

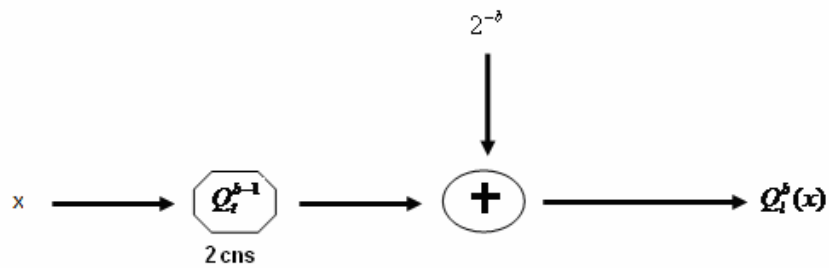


Figura 4.17 Equivalente de cuantizadores de LSB-1redondeo.
 PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.539

4.1.2.5 RANGO DINAMICO

Considerando el siguiente rango dinámico del sistema numérico en complemento a 2

$$-1 \leq Q^b(x) \leq 1 - 2^{-b} \quad (4.56)$$

$$\begin{aligned}
(D.R)_{2cns} &= \frac{|Q^b(x)|_{\max}}{|Q^b(x)|_{\min} \neq 0} \\
(D.R)_{2cns} &= \frac{|-1|}{2^{-b}} \\
(D.R)_{2cns} &= 2^{-b}
\end{aligned}
\tag{4.57}$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.531, P.540

4.2 COEFCIENTE DE CUANTIZACION

En uno de los efectos de la longitud de la palabra finita de las computadoras es que los de parámetros o coeficientes del filtro deben ser elegidos de un conjunto finito de valores

4.2.1 UBICACION DE POLOS Y CEROS

Los coeficientes de cuantización de un filtro digital restringen los polos y ceros del filtro en un número finito de punto discreto en el plano z. Considere el filtro del segundo-orden:

$$D(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \tag{4.58}$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.531, P.541

El análisis de sensibilidad puede ser empleado para determinar el efecto del coeficiente de cuantificación en el polo de migración o cambio en la característica de transferencia.

La migración de polos esta dado por:

$$\begin{aligned}
\Delta r &= \frac{\partial r}{\partial b_1} \Delta b_1 + \frac{\partial r}{\partial b_2} \Delta b_2 \\
\Delta \theta &= \frac{\partial \theta}{\partial b_1} \Delta b_1 + \frac{\partial \theta}{\partial b_2} \Delta b_2
\end{aligned}
\tag{4.59}$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.531, P.541

Usando la relación:

$$\begin{aligned}
\frac{\partial r}{\partial b_1} &= -\frac{1}{2 \cos \theta} \\
\frac{\partial r}{\partial b_2} &= \frac{1}{2r} \\
\frac{\partial \theta}{\partial b_1} &= \frac{1}{2r \sin \theta} \\
\frac{\partial \theta}{\partial b_2} &= \frac{1}{2r^2 \tan \theta}
\end{aligned}
\tag{4.60}$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.531, P.543

Por lo tanto:

$$\begin{aligned}
\Delta r &= \frac{-\Delta b_1}{2 \cos \theta} + \frac{\Delta b_2}{2r} \\
\Delta \theta &= \frac{\Delta b_1}{2r \sin \theta} + \frac{\Delta b_2}{2r^2 \tan \theta}
\end{aligned}
\tag{4.61}$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.531, P.543

Estas relaciones nos muestran que, para un r dado, cuando θ tiende a 0 , $\Delta \theta$ tiende a ∞ . Similarmente para una constante θ , cuando r tiende a 0 , $\Delta \theta$ y Δr ambos tienden a infinito.

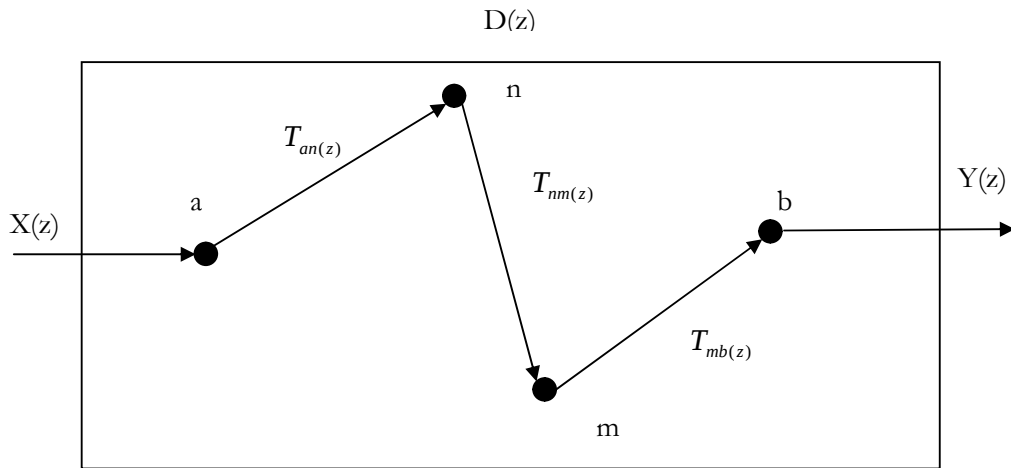


Figura 4.18 Equivalente de cuantizadores de LSB-1 redondeo.

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.543

Se considera en lo que sigue el efecto del coeficiente de cuantización en la función de transferencia del filtro digital. El punto a de la Figura 4.18 es el nodo

de entrada para el filtro; el punto b es el nodo de salida. Entonces la sensibilidad en $D(z)$ con respecto T_{an} es dada por la siguiente ecuación

$$\frac{\partial D(z)}{\partial F_{nm}(z)} = T_{an}(z)T_{mb}(z) \quad (4.62)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.544

Si la rama en la que se está trabajando es tiene un coeficiente multiplicativo

$$\frac{\partial |D(z)|}{\partial f_{cnm}} = \operatorname{Re} \left[\frac{|D(z)|}{D(z)} T_{an}(z)T_{mb}(z) \right] \quad (4.63)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.544

La rama incluye un elemento que se encuentra en retraso

$$\frac{\partial |D(z)|}{\partial f_{dnm}} = \operatorname{Re} \left[\frac{|D(z)|}{D(z)} T_{an}(z)T_{mb}(z)z^{-1} \right] \quad (4.64)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.544

La sensibilidad puede usarse para calcular el cambio en la función de traslado del filtro $D(z)$ debido a que se crea un cambio en el valor del coeficiente.

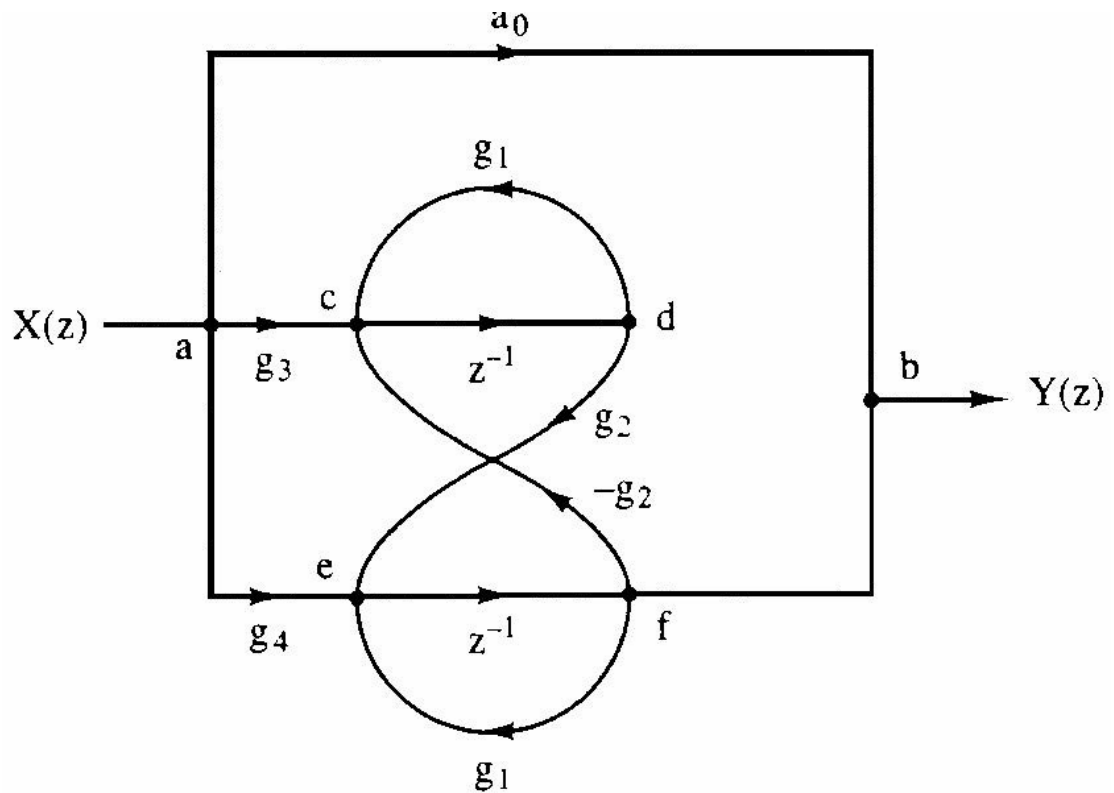


Figura 4.19 Estructura 1X.

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.545

Considerando la estructura 1x mostrada en la Figura 4.19 se puede calcular la sensibilidad de la siguiente manera

$$D(z) = \frac{a_0 + g_4 z^{-1} + g_2 g_3 z^{-2}}{1 - 2g_1 z^{-1} + (g_1 + g_2) z^{-2}} = a_0 + \frac{\frac{g_4 - jg_3}{2}}{z - g_1 + jg_2} + \frac{\frac{g_4 + jg_3}{2}}{z - g_1 - jg_2} \quad (4.65)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.544

Cuando se usan los filtros digitales en los sistemas de mando de regeneración los coeficientes de cuantización del filtro pueden afectar el comportamiento dramáticamente de un sistema de lazo cerrado en aplicaciones dónde la ubicación de los polos y ceros es crítica.

4.3 ANALISIS DE LA SEÑAL DE CUANTIZACION

Toda señal que llega a los filtros digitales después de haber pasado por los convertidores analógicos digitales se le debe cuantizar en las diferentes etapas de trabajo de la señal pues esta está expuesta a saturación por parte de la cantidad de bits usados.

4.3.1 CUANTIZACION DE LA ENTRADA DEL FILTRO

La señal de entrada de un filtro digital, $x(t)$, puede venir de un conversor A/D o de la salida de otro módulo digital; la señal $x(t)$ es muestreada y cuantizada en una secuencia en el tiempo $\{Q(x(n))\}$ que se procesa en el filtro digital.

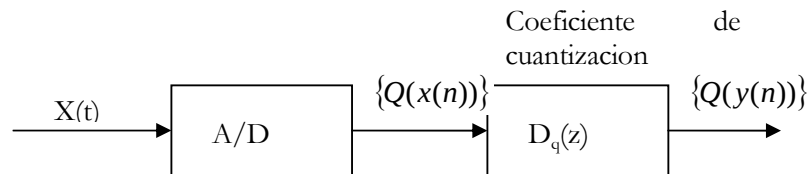


Figura 4.20 Conversor A/D.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.546

El conversor A/D más común es el conversor bipolar de aproximaciones sucesivas en el que el tiempo de conversión es proporcional al número de bits $(b+1)$; este tipo de convertidor es un caso de cuantización por truncamiento

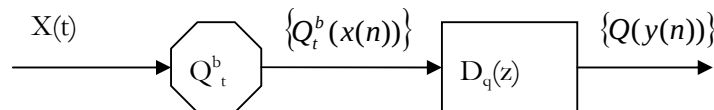


Figura 4.21 Conversor A/D de aproximaciones sucesivas.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.546

Si la señal de entrada es polarizada por una señal pequeña $2^{-b}/2$ esta señal redondea los valores de entrada

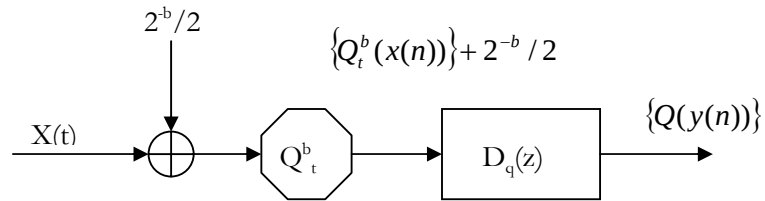


Figura 4.22 Conversor A/D de aproximaciones sucesivas bias.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.546

Este es el modelo de conversor A/D más usado y se satura generalmente cuando la señal de entrada $x(t)$ excede su gama dinámica; su característica de desbordamiento es:

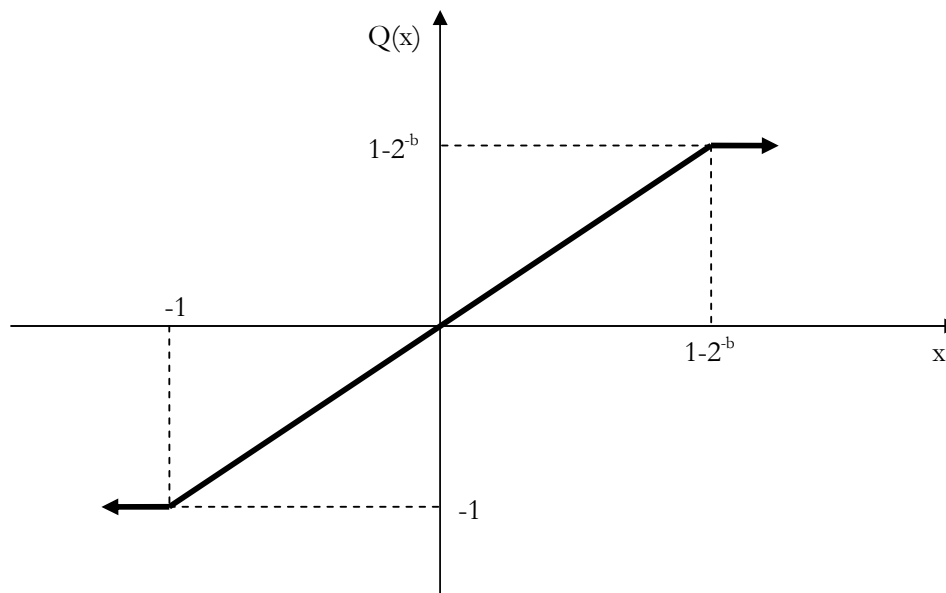


Figura 4.23 Característica de overflow para el conversor A/D

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.547

El conversor análogo digital A/D introduce un ruido de redondeo en el filtro de la siguiente manera:

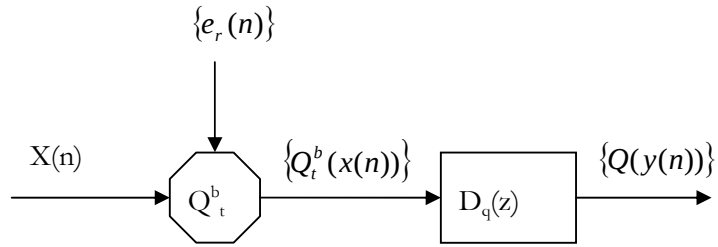


Figura 4.24 Modelo de conversor A/D con una señal de ruido.
 PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.547

En donde el error de redondeo es:

$$e_r(n) = Q_t^b(x(n)) - x(n) \quad (4.66)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.547

Este error de redondeo está para un tiempo nT . El ruido de redondeo está distribuido uniformemente con una variación.

$$\sigma_{er}^2 = \frac{2^{-b}}{12} \quad (4.67)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.547

Al igual a este ruido lo limita

$$\frac{2^{-b}}{2} \geq e_r > -\frac{2^{-b}}{2} \quad (4.68)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.547

4.3.2 CUANTIZACION DE LAS VARIABLES INTERNAS

Los nodos internos se forman sumando los términos del producto, donde cada producto se genera por un coeficiente del filtro y una variable de la señal

$$v_k = \sum_{i=1}^L c_i v_i(n) \quad (4.69)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.548

Si $v_i(n)$ representa las variables internas y c_i los coeficientes internos del filtro. Un nodo interno $v_k(n)$ es generado por la suma de varios nodos L . El caso ideal sin cuantizar es:

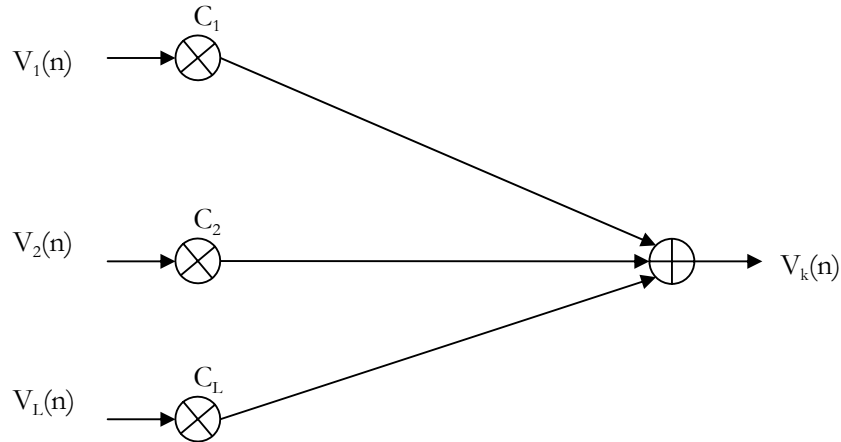


Figura 4.25 Caso ideal de la cuantización interna de variables.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.549

Los coeficientes (c_i) son cuantizados (c_i^q). Si las variables se representan en un sistema numérico de palabra finita se dice que es de complemento a dos físicamente, es:

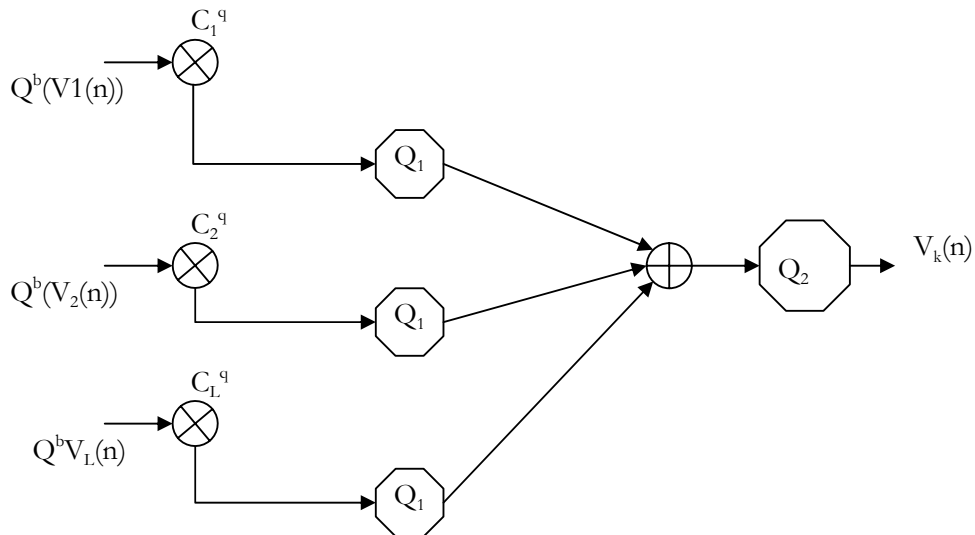


Figura 4.26 Caso físicamente realizable de la cuantización interna de variables.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.549

Si los coeficientes son cuantizados en "a" bits y las variables son cuantizadas en "b" bits el producto $c_i^q Q^b(V_i(n))$ tendrá "a+b" bits. Los términos del producto

pueden entonces ser cuantizados en “b” bits por las variables cuantizadoras Q_1 , y son sumados en su totalidad (a+b bits) y la suma que resulta cuantizada en “b” bits por el cuantizador Q_2 .

En la cuantización del punto Q_1 genera una varianza en los nodos

$$\sigma_e^1 = \sum_{i=1}^L \sigma_{e1}^2 = L \sigma_{e1}^2 = L \sigma_{e2}^2 \quad (4.70)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.548

El error de redondeo que resulta se puede expresar como

$$e_r = Q_r^b (c_i^q v_i^q) - c_i^q v_i^q \quad (4.71)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.548

Y por lo tanto

$$\begin{aligned} e_r &= 2^{-b-1} - (0.00\dots 0m_{3,b+1} \oplus 1m_{3,b+2}\dots m_{3,b+a})_{2cns} \\ e_r &= 2^{-b} (2^{-1} - (m_{3,b+1} \oplus 1m_{3,b+2}\dots m_{3,b+a})_2) \end{aligned} \quad (4.72)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.548

Por lo tanto el número binario puede ser simplificado dejando

$$\begin{aligned} m_1 &= m_{3,b} + 1 \\ m_i &= m_{3,b+i} \quad \text{para } i=2, \end{aligned} \quad (4.73)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.550

El error puede adquirir un valor discreto que esta en un rango

$$0 \leq (.m_1 m_2 \dots m_a)_2 \leq (1-2^{-a}) \quad (4.74)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.550

Entonces

$$0 \geq -(m_1 m_2 \dots m_a)_2 \geq -1-2^{-a} \quad (4.75)$$

$$1/2 \geq 1/2 - (.m_1 m_2 \dots m_a)_2 \leq (1/2 - 1 - 2^{-a})$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.550

Y por lo tanto los límites de error están definidos

$$\frac{2^{-b}}{2} \geq e_r \geq \frac{-2^{-b}}{2} + 2^{-b-a} \quad (4.76)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.550

e_r esta expresado

$$2^b e_r = (.10 \dots 0)_2 - (.m_1 m_2 \dots m_a)_2 = (n_0 n_1 n_2 \dots n_a)_{2^{cns}} \quad (4.77)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.551

La función de la probabilidad discreta es

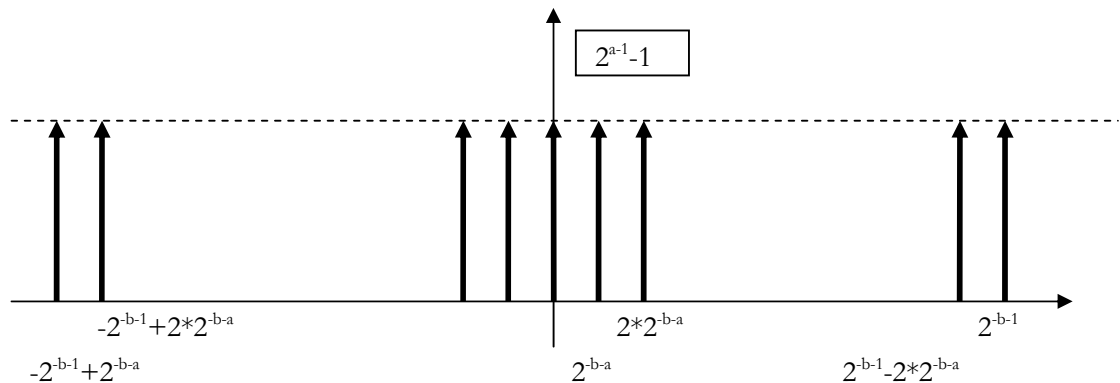


Figura 4.27 Error de la probabilidad de densidad de redondeo

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.551

La función de la densidad consiste en una serie de funciones de diversas de un impulso, cada uno valor de 2^{-a} . la función de la densidad de la probabilidad

4.3.3 RUIDO DE CUANTIZACION

El modelo del filtro digital asume Q como la suma de las uniones. La entrada cuantizada del filtro se representa por $e_o(n)$.

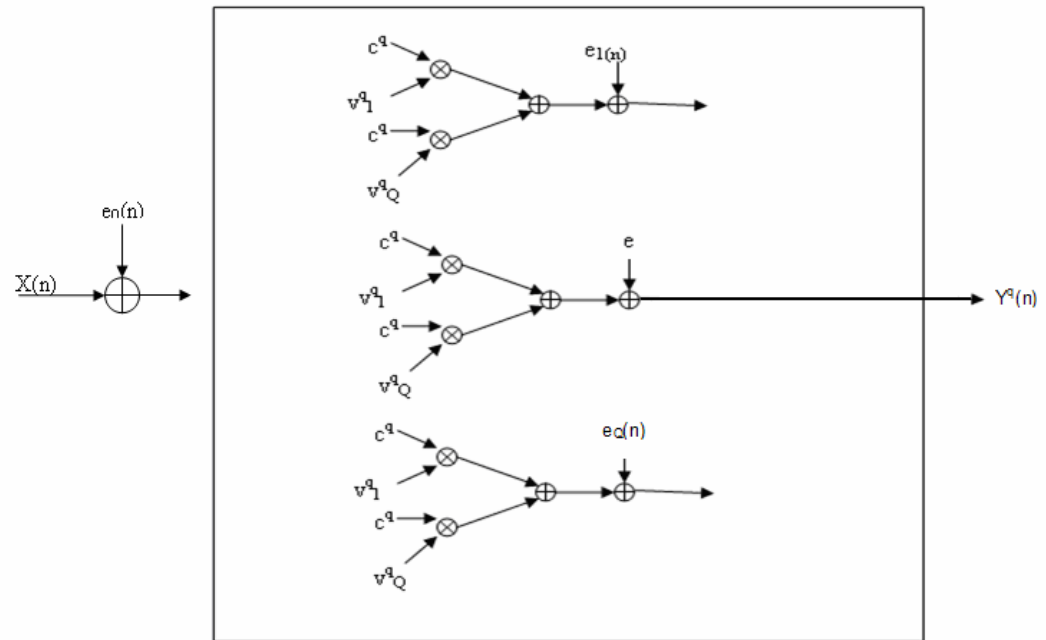


Figura 4.28 Modelo del ruido del filtro

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.555

Para cada función de transferencia de la entrada i se crea un error de la siguiente manera

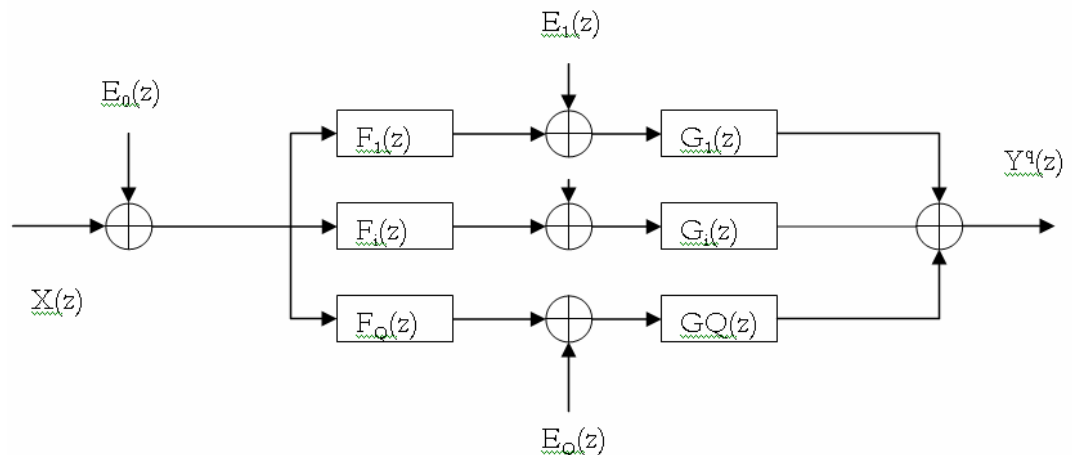


Figura 4.29 Modelo del ruido de cada termino interno del filtro

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.556

Para cada bloque se utiliza un modelo de ruido de la siguiente manera:

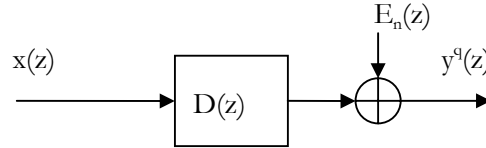


Figura 4.230 Modelo del ruido de cada termino interno del filtro

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.556

$$y^q(z) = x(z) * D(z) + E_n(z) \quad (4.78)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.554

Donde en la salida del sistema la salida seda de la siguiente manera

$$E_n(z) = E_o(z)D(z) + \sum_{i=1}^Q E_i(z)G_i(z) \quad (4.79)$$

PHILLIPS

álisis and Desing. Cuarta Edición. Prentice Hall. P.554

Por consiguiente:

$$|e_{mi}(n)| \leq \left(2^{-b} \cdot \frac{L_i}{2} \right) \sum_{j=0}^{\infty} |g_i(j)| \quad (4.80)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.555

De esta manera

$$\begin{aligned} e_n(n) &= \sum_{i=0}^Q e_{mi}(n) \\ |e_n(n)| &\leq \sum_{i=0}^Q |e_{mi}(n)| \\ &\leq \sum_{i=0}^Q \left(2^{-b} \cdot \frac{L_i}{2} \right) \sum_{j=0}^{\infty} |g_i(j)| \end{aligned} \quad (4.81)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.555

Para que

$$|e_n(n)| \leq \frac{2^{-b}}{2} \sum_{i=0}^Q \left(L_i \sum_{j=0}^{\infty} |g_i(j)| \right) \quad (4.82)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.556

Éste es un límite muy pesimista porque asume para cada caso de la muestra que el ruido de redondeo cuantizado es su valor máximo. Se puede dejar la magnitud del peor caso de cada redondeo de cuantización y se aplica una entrada de paso de magnitud es $2^{-b}/2$ el error de cada caso. Subsecuentemente el error de cuantización para el redondeo puede modelarse como un ruido aleatorio uniformemente distribuido con promedio cero y varianza

$$\sigma_{er}^2 = \frac{2^{-2b}}{12} \quad (4.83)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.558

Por tanto se abandona la derivación de los límites absoluto superior y de estado estacionario y se usa el método estadístico para analizar el error de redondeo de la salida.

$$Q_x(k+1) = E[x(k)x(l)] \quad (4.84)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.558

Para el análisis del ruido de redondeo se trabaja de la siguiente manera

$$\begin{aligned} Q_{er}(0) &= E[e_r^2] = \frac{2^{-2b}}{12} \\ Q_{er}(n) &= 0, \quad n \neq 0 \end{aligned} \quad (4.85)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.559

Por esta razón

$$S_{e_r}(z) = \sum_{n=-\infty}^{\infty} Q_x(n) z^{-n} = \frac{2^{-2b}}{12} \quad (4.86)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.559

Para la salida del filtro con el error de redondeo

$$\begin{aligned} E[e_x^2(n)] &= \frac{1}{W_s} \int_0^{2\pi} S_{e_r}(e^{j\omega T}) |D(e^{j\omega T})|^2 d\omega \\ &= \frac{2^{-2b}}{12W_s} \int_0^{2\pi} |D(e^{j\omega T})|^2 d\omega \end{aligned} \quad (4.87)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.559

Y por lo anterior

$$\sigma_{e_x}^2 = E[e_x^2(n)] = \frac{2^{-2b}}{12} \sum_{m=0}^{\infty} d^2(m) \quad (4.88)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.560

Si este resultado es aplicado al modelo ruido del filtro

$$\sigma_{ni}^2 = \sigma_i^2 \sum_{m=0}^{\infty} g_i^2(m) \quad (4.89)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.560

Como las fuentes del ruido aleatorio de entrada e tienen media cero, entonces

$$\begin{aligned} \sigma_n^2 &= \sum_{i=0}^L \sigma_{ni}^2 = \sum_{i=0}^L \sigma_i^2 \sum_{m=0}^{\infty} g_i^2(m) = \frac{2^{-2\delta}}{12} \sum_{i=0}^L \sum_{m=0}^{\infty} g_i^2(m) \\ &= \frac{2^{-2\delta}}{12} \sum_{i=0}^L \frac{1}{w_s} \int_0^{2\pi} |G_i(e^{jwT})|^2 dw = \frac{2^{-2\delta}}{12} \sum_{i=0}^L \frac{1}{2\pi j} \oint G_i(z) G_i\left(\frac{1}{z}\right) \frac{dz}{z} \end{aligned} \quad (4.90)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.560

Una medida importante del comportamiento de un filtro digital es su relación señal al ruido. Si se computa la relación de la varianza de la señal de salida a la varianza del ruido de salida se tiene:

$$\begin{aligned} \frac{\sigma_y^2}{\sigma_n^2} &= \frac{(1/w_s) \int_0^{2\pi} S_x(e^{jwT}) |G_o(e^{jwT})|^2 dw}{(2^{-2\delta}/12) \sum_{i=0}^L \frac{1}{w_s} \int_0^{2\pi} |G_i(e^{jwT})|^2 dw} \\ &= 12 * 2^{2\delta} \frac{\int_0^{2\pi} S_x(e^{jwT}) |G_o(e^{jwT})|^2 dw}{\sum_{i=0}^L \int_0^{2\pi} |G_i(e^{jwT})|^2 dw} \end{aligned} \quad (4.91)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.560

Se nota que la proporción es dependiente en la señal de entrada específica. Si se escoge una entrada de ruido blanco aleatoria limitada por ± 1.0 , cuando $S_x(e^{jwT})=1$ se tiene:

$$\frac{\sigma_y^2}{\sigma_n^2} = \frac{12 * 2^{2\delta}}{1 + \sum_{i=1}^L \frac{\int_0^{2\pi} |G_i(e^{jwT})|^2 dw}{\int_0^{2\pi} |G_o(e^{jwT})|^2 dw}} \quad (4.92)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.560

4.4 CICLO LÍMITE

4.4.1 DEFINICION DE CICLO LÍMITE

Un ciclo del límite es una condición de oscilación sostenida en un sistema de lazo cerrado, causado por no linealidades dentro del lazo.

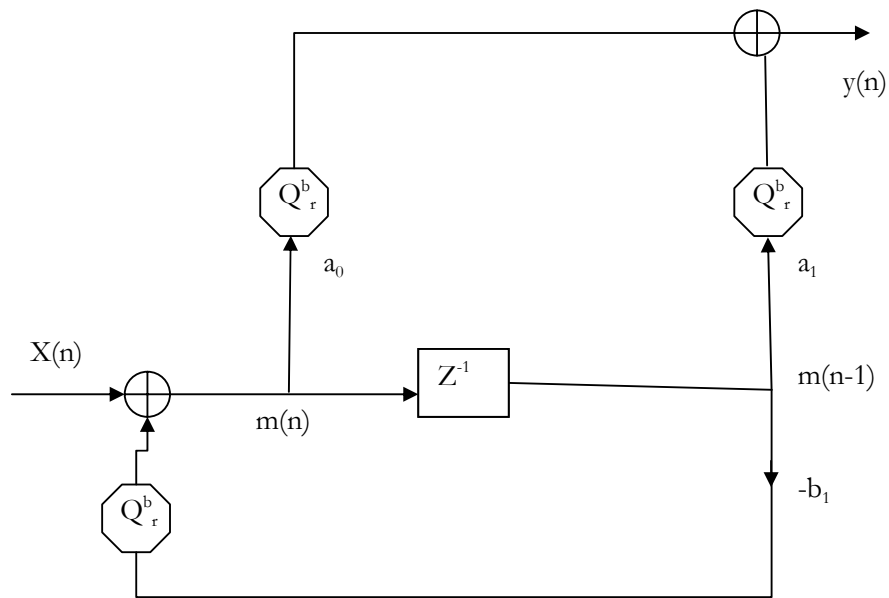


Figura 4.31 Filtro 1D de primer orden.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.561

Considerando este filtro digital de primer orden en el cual $b=3$

$$m(n) = (s.m_1 \quad m_2 \quad m_3)_{2\text{cns}} \quad (4.93)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.560

Y $b_1 = -0.6$. Asumiendo que en el tiempo cero ($n=0$) un pulso de entrada $x(n) = (0.100)_{2\text{cns}} = (0.5)_{10}$ se aplica y los valores subsecuentes de la entrada son cero. La señal $m(n)$ del filtro decaería idealmente a cero y forzaría para llevar a $y(n)$ a cero también. Sin embargo, con la cuantización de redondeo en el lazo se tiene:

N	$m(n-1)$	$-b_1 m(n-1)$	$Q_3 r(-b_1 m(n-1))$	$m(n)$
0	0	0	0	$4/8$
1	$4/8$	$2.4/8$	$2/8$	$2/8$
2	$2/8$	$1.2/8$	$1/8$	$1/8$
3	$1/8$	$0.6/8$	$1/8$	$1/8$
4	$1/8$	$0.6/8$	$1/8$	$1/8$
5	Etc.	Etc.	Etc.	Etc.

Tabla 4.1 Valores de los coeficientes del filtro 1D de primer orden.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.561

Por lo tanto la señal $m(n)$ nunca llega a cero como en el caso ideal. Un cuantizador de truncamiento produciría un valor cero para $m(3)$. Un filtro digital que trabaja como se mostró produce un deadband (banda muerta) e Ideadband cambia el valor del coeficiente de la realimentación; en este caso cambia b_1 a -1.0 . Sabemos que

$$\frac{M(z)}{X(z)} = \frac{1}{1+b_1z^{-1}} \quad (4.94)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.562

y

$$m(n) = x(n) - b_1m(n-1) \quad (4.95)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.562

Una respuesta para el impulso ideal es:

$$\begin{aligned} m(n) &= x(n) - b_1(x(n-1) - b_1m(n-2)) \\ &= 0 - 0 - b_1m(n-2) \\ &= (b_1)^n \end{aligned} \quad (4.96)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.562

Y en caso de cuantización por redondeo

$$m(n) = Q_r^{\delta}(-b_1m(n-1)) \quad (4.97)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.562

Y se tiene en cuenta que para el redondeo

$$|-b_1m(n-1)| - |Q_r^{\delta}(-b_1m(n-1))| \leq 2^{-\delta-1} \quad (4.98)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.562

Por lo tanto

$$|-b_1m(n-1)| - |m(n)| \leq 2^{-\delta-1} \quad (4.99)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.562

si $b_1 > 0$ entonces su valor en el deadband es 1.0 y $m(n) = -m(n-1)$

si $b_1 < 0$, su valor es -1.0 y $m(n) = m(n-1)$ Por lo tanto

$$|m(n)| \leq \frac{2^{-\delta-1}}{1-|b_1|} \quad (4.100)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.562

recordar que $m(n)$ es un número entero (2^{-b}) y por lo tanto un deadband no existe si $|b_1| < 0.5$

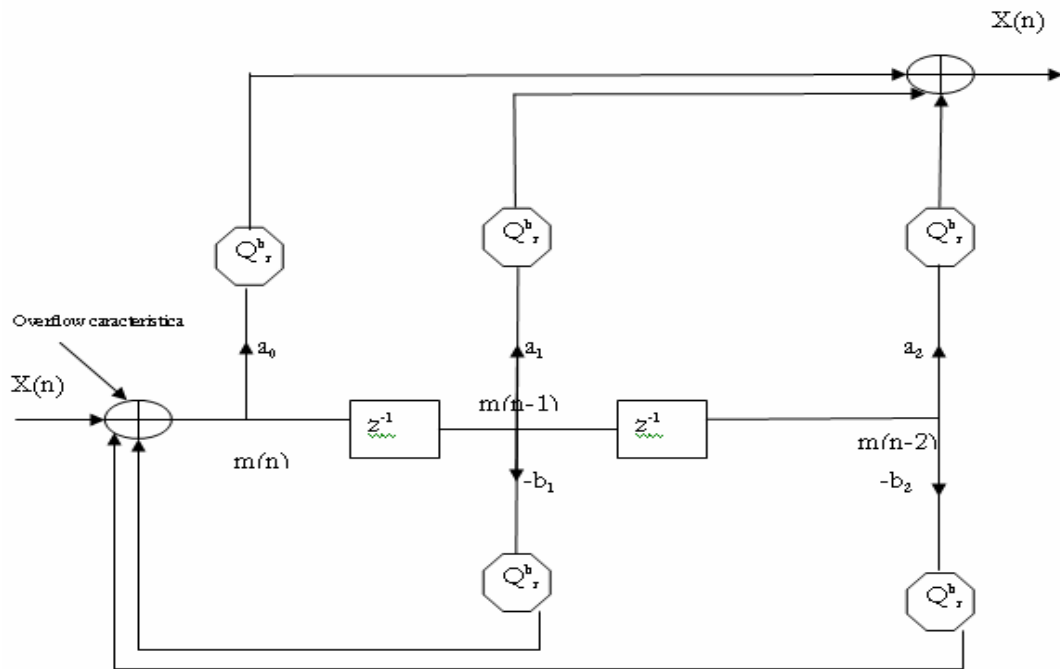


Figura 4.32 Filtro 1D con cuantización múltiple.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.563

Suponiendo un sistema de numeración de complemento a dos de 3 bits con $b_1=+1.5$, $b_2=+0.5$, y una secuencia de entrada de $+5/8.0.0.0\dots$ se aplique a la red. Los polos del filtro son:

$$(z^2 + 1.5z + 0.5) = (z + 1.0)(z + 0.5) \quad (4.101)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.562

de modo que la respuesta ideal del impulso del filtro sea estable:

n	m(n-1)	Q3r(-b1m(n-1))	m(n-2)	Q3r(-b1m(n-2))	m(n)
0	0	0	0	0	5/8
1	5/8	-7.5/8	0	0	-7.5/8
2	-7.5/8	11.25/8	5/8	-2.5/8	8.75/8
3	8.75/8	13.125/8	-7/5.8	3.75/8	-9.375/8
4	-9.375/8	14.0625/8	8.75/8	-4.375/8	9.6875/8
5	9.6875/8	-14.53125/8	-9.375/8	4.6875/8	-9.84375/8
6	-9.84375/8	14.765625/8	9.6875/8	-4.84375/8	9.921765/8
7	9.921875/8	-14.8828125/8	-9.84375/8	4.921875/8	-9.9609375/8
8	-9.9609375/8	14.941406/8	9.921875/8	-4.9608375/8	9.9805688/8

Tabla 4.2 Valores de los coeficientes del filtro 1D de segundo orden.
PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.563

Ahora suponer que aplicamos la cuantización por redondeo a los bits menos significativos y al desbordamiento característico a los bits más significativos de la salida de la serpiente. Se produce la secuencia siguiente:

n	m(n-1)	Q3r(-b1m(n-1))	m(n-2)	Q3r(-b1m(n-2))	m(n)
0	0	0	0	0	5/8
1	5/8	-8/8	0	0	-8/8
2	-8/8	12/8	5/8	-3/8	9/8 → -7/8
3	-7/8	11/8	-8/8	4/8	15/8 → -1/8
4	-1/8	2/8	-7/8	4/8	6/8
5	6/8	-9/8	-1/8	1/8	-8/8
6	-8/8	12/8	6/8	-3/8	9/8 → -7/8
7	-7/8	11/8	-8/8	4/8	15/8 → -1/8

Tabla 4.3 Tabla 4.2 Valores de los coeficientes del filtro 1D de segundo orden.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.563

Observar que la línea n=7 es idéntica a la línea n=3. por lo tanto un la longitud de ciclo limite del límite en la gama es:

$$-1 \leq m(n) \leq +0.75 \quad (4.102)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.564

Se puede producir esencialmente por la característica de desbordamiento de la suma del sistema numérico de complemento a dos según las indicaciones de la Tabla anterior. Por ejemplo, n=2:

$$\begin{array}{rcl}
 Qr3(-b1m(n-1)) & = & 12/8 = 1.100 \\
 + Qr3(-b2m(n-1)) & = & -3/8 = 1.101 \\
 \hline
 m(n) & & 9/8 \quad \underbrace{11.001}_{m(n)}
 \end{array} \quad (4.103)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.564

Por lo tanto

$$m(n) = (1.001)_{2cs} = -(0.111)_2 = -(7/8)_{10} \quad (4.104)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.564

Los ciclos del límite producidos por la característica del desbordamiento se llaman las oscilaciones del desbordamiento.

4.4.2 CLASIFICACION DE ERRORES DE CUANTIZACION

La siguiente tabla ilustra el efecto de la cuantización de la amplitud de la señal y de las características del desbordamiento en la salida de un filtro digital.

TIPO NO LINEAL		
CONDICION DE ENTRADA	CUANTIZACION	OVERFLOW
Entrada de zeros	Ciclo limite	Overflow con saturaciones
Determinar entrada Periódica No periódica	Ciclo limite Cuantización del ruido	
Stochastic entrada	Cuantización del ruido	Overflow del ruido

Tabla 4.4 Clasificación de los errores de cuantización.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.563

4.4.3 OSCILACIONES DEL DESBORDAMIENTO

Las oscilaciones de desbordamiento no se deben ocurrir en un filtro digital. Para evitarlas puede tomar tres caminos:

- Escalar la entrada al filtro de modo que solamente los niveles pequeños de la señal existan en el filtro y nunca ocurra el desbordamiento.
- Diseñar el sumador de modo que su característica de desbordamiento no produzca las oscilaciones.
- Encontrar una estructura digital del filtro que esté libre de oscilaciones de desbordamiento usando siempre aritmética de complemento a dos

4.4.4 CONDICIÓN PARA AUSENCIA DE OSCILACIONES DE DESBORDAMIENTO

El filtro digital en la notación de estado se encuentra de la siguiente forma

$$x(n+1) = Ax(n) + bu(n) \quad (4.105)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.572

Las características del desbordamiento pueden ser impuestas como sigue:

$$x(n+1) = G(Ax(n) + bu(n)) \quad (4.106)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.572

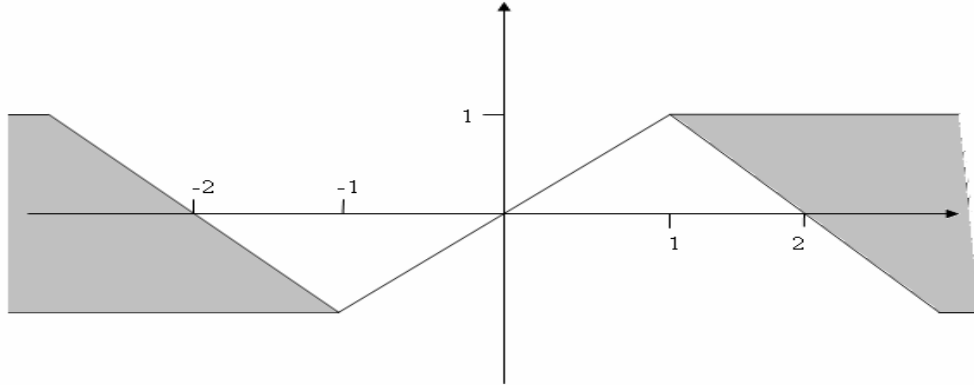


Figura 4.33 Oscilaciones de Overflow.

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.563

Teorema 1. Si hay una matriz diagonal D con elementos diagonales positivos y $D-A^TDA$ es positiva definida las oscilaciones de desbordamiento no existen

Teorema 2. Si A es una matriz (2 x 2) con valores característicos $|\lambda| < 1$ existe una matriz diagonal definida positiva D para la cual $D-A^TDA$ es positiva si y sólo si

$$a_{12}a_{21} \geq 0 \quad (4.107)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.573

O

$$a_{12}a_{21} < 0 \quad \text{and} \quad |a_{11} - a_{22}| + \det(A) < 1 \quad (4.108)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.573

4.5 FILTROS DE SEGUNDO ORDEN EN CASCADA

Un filtro de segundo orden de módulo en cascadas $D(z)$ se puede descomponer en factores de términos del numerador y el denominador: e segundo orden:

$$D(z) = \frac{\prod_{i=1}^m \alpha_i(z)}{\prod_{i=1}^m \beta_i(z)} = \prod_{i=1}^m A_i(z) \quad (4.109)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.575

De donde

$$\begin{aligned} \alpha_i(z) &= \alpha_{i0} + \alpha_{i1}z^{-1} + \alpha_{i2}z^{-2} \\ \beta_i(z) &= 1 + \alpha_{i3}z^{-1} + \alpha_{i4}z^{-2} \end{aligned} \quad (4.110)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.575

Donde m es un número entero pequeño mayor que $n/2$. Los términos del numerador y del denominador de segundo orden deben tener paridad; entonces los pares se deben ser ordenados en cascada.

Observamos que la estructura 1D tiene dos fuentes de errores por el módulo y la 3D solamente tiene una fuente de error. Sin embargo, las estructuras 1D y 4D tienen comportamiento similar de error, al igual que las 2D y 3D. Acentuamos que las oscilaciones de desbordamiento pueden producir resultados desastrosos.

En la práctica los factores de posicionamiento se introducen en cascada para limitar amplitudes de la señal de modo que no ocurra el desbordamiento. Examinemos un módulo en cascada de segundo orden 1D (o 4D) con los coeficientes del escalamiento incluidos en la cascada.

4.5.1 ESCALAMIENTO DE LA SEÑAL

En un filtro 1D de segundo orden de complemento a dos la señal interna $m(n)$ excede la gama dinámica $[-1 < m(n) \leq 1]$ del sistema de numeración de punto fijo y por lo tanto ocurre overflow e introduce oscilaciones grandes en la estructura.

Las oscilaciones de desbordamiento pueden ser eliminadas generalmente reduciendo (escalando) la señal de entrada a un filtro digital. Si se escala las señales a valores muy pequeños los errores de cuantización se convierten en una parte significativa de las señales internas y se degrada el cociente de señal a ruido.

Por lo tanto un problema importante del diseño es elegir los factores de posicionamiento que reduzcan la probabilidad del desbordamiento mientras que se mantiene la señal en los niveles significativos de la gama dinámica; los coeficientes de escalamiento se eligen tales que la magnitud de los valores internos de la señal no excede de 1, el límite del desbordamiento.

4.5.1.1. ESCALAMIENTO DEL LÍMITE SUPERIOR

Para limitar la señal V_i^q en cualquier momento a 1 podemos seleccionar el factor de escalamiento de la entrada λ_i tal que

$$V_i^q = \lambda_i X(z) F_i(z) \quad (4.111)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.579

El desbordamiento nunca puede ocurrir en la estructura del filtro. Es un caso absoluto del límite superior y es demasiado conservador en la mayoría de usos.

Los niveles de la señal son completamente restringidos, dejando gran parte de la gama dinámica del filtro sin uso.

4.5.2 OPTIMIZACION

Para reducir al mínimo el ruido de la salida de un filtro 1D en segundo orden en cascada se pueden utilizar las técnicas de optimización. El método del escalamiento primero se selecciona. Por ejemplo, si el escalamiento de L2-norm se utiliza en $q=2$. Así pues, para el módulo del i th,

$$s_0 s_1 s_2 \cdots s_{i-1} = \frac{1}{\left\| \frac{A_1 A_2 \cdots A_{i-1}}{\beta_i} \right\|_2} \quad (4.112)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.585

$$\begin{aligned} (s_0 \cdots s_{i-1}) &= \left\| \frac{A_1 \cdots A_{i-1}}{\beta_i} \right\|_2 = 1 \\ (s_0 \cdots s_{i-1})^2 &\left[\frac{1}{w_j} \int_0^{w_j} \left| \frac{A_1 \cdots A_{i-1}}{\beta_i} \right|^2 d\omega \right] = 1 \end{aligned} \quad (4.113)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.585

Consecuentemente

$$\frac{1}{2\pi j} \oint \prod_{i=1}^i s_{i-1}^2 \frac{\alpha_{i-1}(z) \alpha_{i-1}(1/z)}{\beta_i(z) \beta_i(1/z)} \frac{dz}{z} = 1 \quad (4.114)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.585

donde $\alpha_0(z) = 1$:

$$\sigma_{en}^2 = \frac{2^{-2b}}{2} \left[L_m + \sum_{i=1}^m \frac{L_{i-1}}{2\pi j} \oint \prod_{i=1}^i s_{i-1}^2 \frac{\alpha_{i-1}(z) \alpha_{i-1}(1/z)}{\beta_i(z) \beta_i(1/z)} \frac{dz}{z} \times \frac{1}{2\pi j} \oint \prod_{i=1}^m s_i^2 \frac{\alpha_i(z) \alpha_i(1/z)}{\beta_i(z) \beta_i(1/z)} \frac{dz}{z} \right] \quad (4.115)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.585

Esta relación es la que está usada en el optimization. La mayoría de los diseñadores de filtros digitales en tiempo real no tienen programas de la optimización.

Por lo tanto, las pautas del diseño que dan generalmente “buenos” resultados, pero no óptimo. Uno debe recordar siempre que la solución óptima implica la selección de un método particular del escalamiento. La selección de un método “óptimo” del escalamiento dependerá solo del uso. Por ejemplo, en sistemas de control de lazo cerrado, los niveles de la señal en un regulador digital pueden ser muy bajos. En estos casos L_∞ o el escalamiento absoluto del límite superior no es sabio.

4.5.3 PARIEDAD DE POLOS Y CEROS

La optimización del apareamiento de posibles términos del numerador y denominador requiere $(m!)^2$ evaluaciones. Jackson sugiere que se obtiene buenos resultados si se selecciona un apareamiento que minimiza $\|A_i\|_\infty$ para todo i .

$$\|A_i\|_\infty = \max_{w \in [0, \infty)} |A_i(w)| \quad (4.116)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.586

Podemos minimizar $\|A_i\|_\infty$ minimizando la respuesta en frecuencia de A_i . El procedimiento gráfico, aunque no óptimo, da buenos resultados minimizando la respuesta en frecuencia de los pares de polos que están cerca del círculo unitario:

- Graficar polos y ceros en el plano z .
- Aparear polos reales. Aparear el polo real más cercano al punto $z=+1$ con el polo más lejano del punto $z=+1$. Continuar hasta que se hayan apareado todos los polos reales.
- Apareamiento de polos y ceros. Acoplar el polo más cercano al círculo unitario con el cero más cercano a él. Si el cero es real acoplar el otro polo del par de polos con el cero real más cercano. Repetir el paso hasta que todos los polos y los ceros se hayan apareado

Ejemplo 4.1: La optimización de un filtro 1D de orden siete en conexión en cascada se realiza así:

$$\begin{aligned} \alpha_1 &= 1 + 1.12368z^{-1} + z^{-2} \\ \beta_1 &= 1 + 0.05358156z^{-1} + 0.9403549z^{-2} \\ \alpha_2 &= 1 + 0.216853z^{-1} + z^{-2} \\ \beta_2 &= 1 + 0.019248z^{-1} + 0.74403z^{-2} \\ \alpha_3 &= 1 + 0.42371z^{-1} + z^{-2} \\ \beta_3 &= 1 + 0.19405z^{-1} + 0.33908z^{-2} \\ \alpha_4 &= 1 + z^{-1} \\ \beta_4 &= 1 + 0.16535z^{-1} \end{aligned} \quad (4.117)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.586

Considerando los polos y ceros se puede seguir el algoritmo:

1. El polo real β_4 se aparea con α_4
2. Polos complejos β_1 se aparean con los ceros complejos α_2

3. Polos complejos β_2 se aparean con los ceros complejos α_3
4. Polos complejos β_3 se aparean con los ceros complejos α_1

Ejemplo de apareamiento de polos y ceros

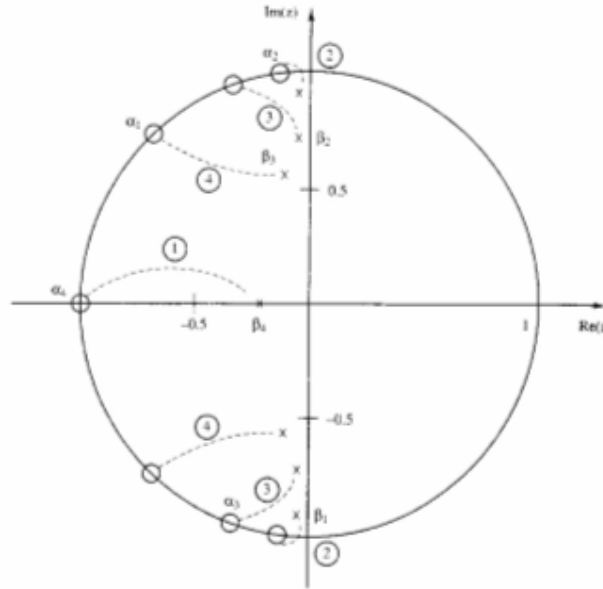


Figura 4.34 Paridad de polos y ceros

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.587

4.5.4. ORDENAMIENTO

Una vez apareados los módulos de segundo orden deben ser ordenados para minimizar el ruido de la salida y del ciclo del límite.

Los algoritmos de ordenamiento propuestos se pueden derivar en un método basado para la solución de estructuras 1D y de esta manera ser extendida hacia otras estructuras propuestas.

Como:

$$\begin{aligned} \frac{1}{2\pi j} \oint A_i(z) A_i(1/z) dz / z &= \frac{1}{W_s} \int_0^{W_s} |A_i(e^{jwT})|^2 dw \\ &= \sum_{m=0}^{\infty} \alpha_i^2(m) \end{aligned} \quad (4.118)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.587

Se puede escribir σ_{en}^2 en la forma:

$$\sigma_{en}^2 = \frac{2^{-2\delta}}{12} \left[L_m + \sum_{i=1}^m L_{i-1} \frac{1}{W_s} \int_0^{W_s} \left| \prod_{i=1}^m s_i A_i (e^{j\omega T}) \right|^2 d\omega \right] \quad (4.119)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.587

Consecuentemente

$$\sigma_{en}^2 = \frac{2^{-2\delta}}{12} \left[L_m + \sum_{i=1}^m L_{i-1} \frac{1}{W_s} \int_0^{W_s} \left| \prod_{i=1}^m s_i^2 A_i (e^{j\omega T}) \right|^2 d\omega \right] \quad (4.120)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.588

Podemos intercambiar la integral y el producto y crear una desigualdad:

$$\begin{aligned} \sigma_{en}^2 &\leq \frac{2^{-2\delta}}{12} \left[L_m + \sum_{i=1}^m L_{i-1} \frac{1}{W_s} \int_0^{W_s} \left| \prod_{i=1}^m s_i^2 A_i (e^{j\omega T}) \right|^2 d\omega \right] \\ &\leq \frac{2^{-2\delta}}{12} \left[L_m + \sum_{i=1}^m L_{i-1} \prod_{i=1}^m C_i \right] \end{aligned} \quad (4.121)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.588

Donde

$$C_i = \frac{1}{W_s} \int_0^{W_s} s_i^2 |A_i (e^{j\omega T})|^2 d\omega = \sum_{m=0}^{\infty} s_i^2 \alpha_i^2(m) \quad (4.122)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.588

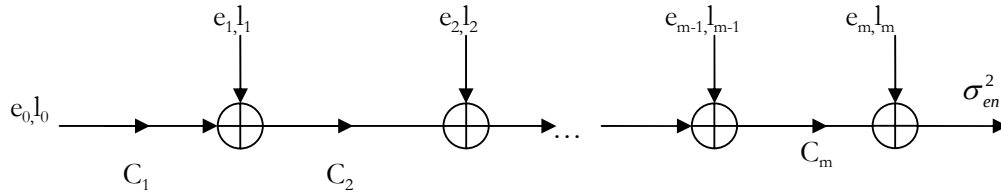


Figura 4.35 Modelo de ordenamiento para un modulo 1D

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.589

Que puede se expandida por:

$$\sigma_{en}^2 \leq \frac{2^{-2\delta}}{12} [L_m + L_{m-1}C_m + L_{m-2}C_{m-1}C_m + \dots + L_0C_1C_2 \dots C_m] \quad (4.123)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing. Cuarta Edición. Prentice Hall. P.589

Que a su vez se puede expandir de la siguiente manera

$$\sigma_{en}^2 \leq (C_m \cdots (C_2(C_1 L_0 + L_1) + L_2) \cdots + L_m) \frac{2^{-2\delta}}{12} \quad (4.124)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.589

Como $L_0=2$, $L_m=3$ y $L_1=5$, σ_{en}^2 se puede reducir exigiendo que

$$C_1 \geq C_2 \geq C_3 \cdots \geq C_m \quad (4.125)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.589

Estas constantes se pueden calcular para cada módulo una vez que se haya logrado el apareamiento y el escalamiento.

Para lograr un buen diseño del filtro se deben tener en cuenta unas pautas para su diseño:

1. Obtener la función $D(z)$.
2. Cuantificar los coeficientes para la longitud de palabra usada
3. Verificar que los coeficientes de cuantización resuelven las especificaciones de sistema
4. Aparear polos y ceros
5. Elegir una estructura para el filtro
6. Aplicar el escalamiento independientemente a cada módulo. Sugerimos el método de paso unitario. Sin embargo, uno de los otros métodos puede ser utilizado.
7. Aplicar el procedimiento ordenamiento. Este procedimiento utiliza las constantes que se pueden calcular para cada módulo independientemente. El diseño está completo ahora.
8. Simular el diseño para lazo abierto y probar la respuesta a un paso para asegurar que la gama dinámica de las variables internas está en el rango medio.
9. Simular el diseño en lazo cerrado para asegurarse de que las especificaciones de sistema son las mejores

4.6 FILTROS DE SEGUNDO ORDEN EN PARALELO

Suponiendo una conexión paralela de un filtro de módulos de segundo orden, $D(z)$ se puede expresar, usando una extensión de la fracción parcial, como:

$$D(z) = \beta_0 + \sum_{i=1}^m \beta_i(z) \quad (4.126)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.590

Donde se transforma en

$$B_i(z) = \frac{\beta_{i1}z^{-1} + \beta_{i2}z^{-2}}{1 + \beta_{i3}z^{-1} + \beta_{i4}z^{-2}} = \frac{\alpha_i(z)}{\beta_i(z)} \quad (4.127)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.590

Representación del modulo paralelo.

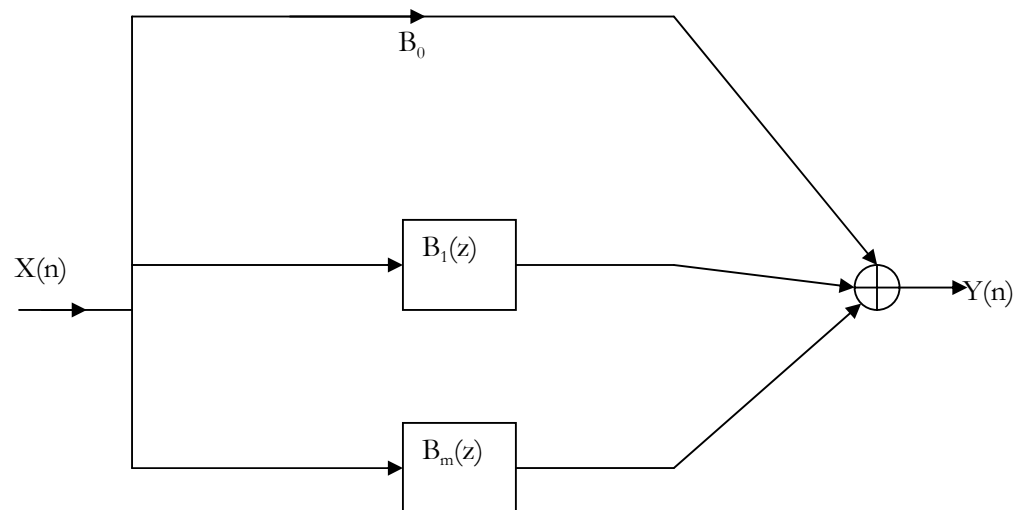


Figura 4.36 Modelo de implementación general de paralelo

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.591

El B_i de los módulos se puede poner en ejecución usando las estructuras directas.

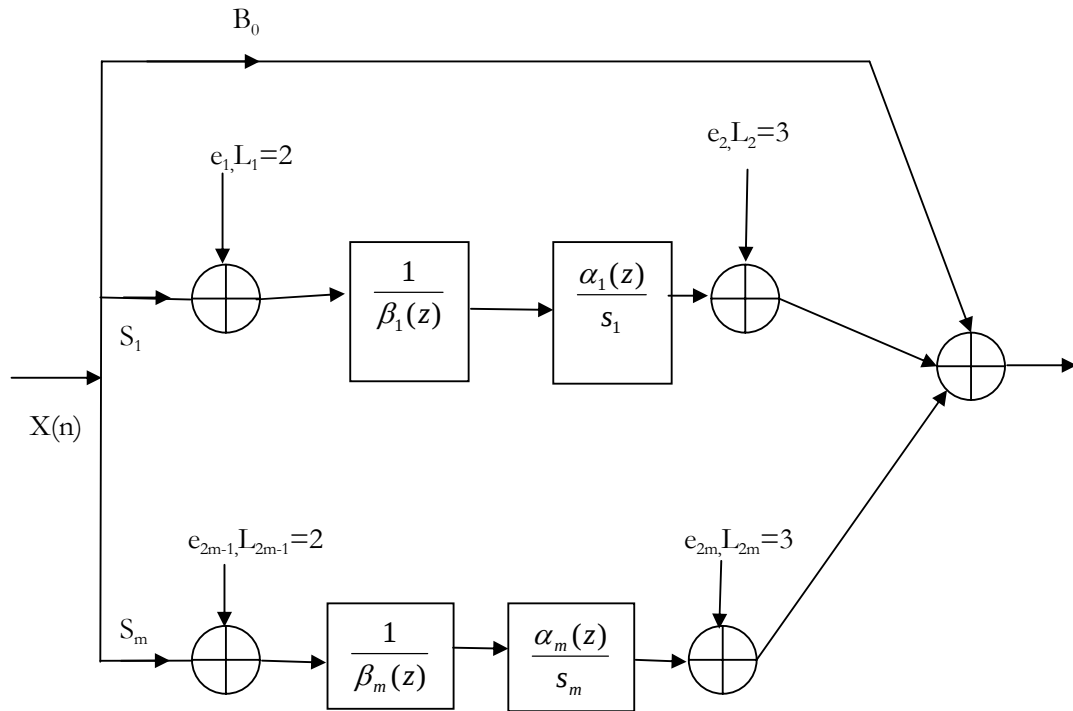


Figura 4.37 Modelo de implementación general de paralelo para módulos 1D y 4D
PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.591

Las constantes del escalamiento se han agregado para evitar desbordamiento en los módulos de segundo orden. El inverso de la constante del escalamiento se utiliza para restaurar el nivel de la señal antes de que se alcance la adición de la salida. Aquí la variación del error puede ser expresada, usando

$$\sigma_{en}^2 = \frac{2^{-2\delta}}{12} \left[3m + \sum_{i=1}^m \frac{2m}{2\pi j} \oint \frac{B_i(z) B_i(1/z) dz}{s_i^2 z} \right] \quad (4.128)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.590

Y donde

$$\sigma_{en}^2 = \frac{2^{-2\delta}}{12} \left[3m + \sum_{i=1}^m \frac{2m}{s_i^2} \sum_{l=0}^m b_i^2(l) \right] \quad (4.129)$$

PHILLIPS Charles, NAGLE H. Digital Control System Analysis and Design. Cuarta Edición. Prentice Hall. P.590

En donde $L_m=5$. La puesta en práctica de un filtro digital paralelo es un proceso algo directo.

Que debe llevar por supuesto las siguientes indicaciones

1. Realizar la expansión en fracciones parciales

2. Cuantificar los coeficientes con la palabra seleccionada.
3. Verificar que los coeficientes de cuantización cumplan las especificaciones del sistema.
4. Elegir una estructura para los módulos de segundo orden.
5. El escalamiento se puede aplicar a cada módulo. La puesta en práctica es completa ahora.
6. Simular el filtro digital en lazo abierto y probar con impulso unitario y una onda seno para asegurar que la gama dinámica de todas las variables es apropiada.
7. Completar la simulación digital del filtro con la simulación total del sistema para asegurar que las especificaciones de sistema han estado satisfechas.

Observamos que el procedimiento de diseño para el caso paralelo es similar al modulo de cascada, con excepción del hecho que el apareamiento y el ordenamiento no son necesarios.

5. ANALISIS DE UN COTROLADOR PID UTILIZANDO EL FDATAOL

Un controlador PID es aquel sistema que usa una constante proporcional (K_p), una constante integral (K_i) y una constante derivativa (K_d). La función de transferencia $D(z)$ para un controlador PID es:

$$D(z) = K_p + \frac{K_i T}{2} \frac{(z+1)}{(z-1)} + \frac{K_d}{T} \frac{(z-1)}{z} \quad (5.1)$$

PHILLIPS Charles, NAGLE H. Digital Control System Análisis and Desing.
Cuarta Edición. Prentice Hall. P.483

Al desarrollar la ecuación (5.1) teniendo en cuenta que:

$$D(z) = \frac{M(z)}{E(z)} \quad (5.2)$$

Donde $M(z)$ es la salida y $E(z)$ es la entrada respectivamente del controlador PID. Se obtiene lo siguiente:

(5.3)

$$\begin{aligned}
\frac{M(z)}{E(z)} &= \frac{Kp(z-1)z + \left(\frac{Ki}{2}\right)(z-1)z + \left(\frac{Kd}{T}\right)(z-1)^2}{z(z-1)} \\
&= \frac{(Kp * z^2) - (Kp * z) + \left(\frac{KiT}{2} * z^2\right) + \left(\frac{KiT}{2} * z\right) + \left(\frac{Kd}{T} * z\right) - \left(\frac{Kd}{T}\right)}{z(z-1)} \\
&= \frac{\left(Kp + \frac{KiT}{2}\right)z^2 + \left(\frac{KiT}{2} - Kp + \frac{Kd}{T}\right)z - \frac{Kd}{T}}{z^2 - z} = \frac{A + Bz^{-1} - Cz^{-2}}{1 - z^{-1}}
\end{aligned}$$

$$\begin{aligned}
A &= \left(Kp + \frac{KiT}{2}\right) \\
\text{donde : } B &= \left(\frac{KiT}{2} - Kp + \frac{Kd}{T}\right) \\
C &= \frac{Kd}{T}
\end{aligned}$$

Al despejar M(z), se tiene:

$$\begin{aligned}
M(z)(1 - z^{-1}) &= E(z)(A + Bz^{-1} - Cz^{-2}) \\
\text{en términos del tiempo :} & \\
m(kt) &= Ae^{kt} + Be^{(k-1)t} - Ce^{(k-2)t} + m[(k-1)t]
\end{aligned} \tag{5.4}$$

La ecuación (5.4) comprueba que el controlador PID se comporta como un filtro IIR, es decir su salida depende del valor actual de entrada (Ae^{kt}), de valores pasados de la señal de entrada ($Be^{(k-1)t}$, $Ce^{(k-2)t}$) y se observa una realimentación del valor pasado de la salida ($m[(k-1)t]$). Debido a esta característica nos centraremos en la aplicación del FDATool para filtros IIR.

5.1 EJEMPLO

Para el siguiente caso un sistema con una función de transferencia de $G(s) = \frac{10}{s^2 + 3s + 2}$ se crea un pid como lo muestra la figura:

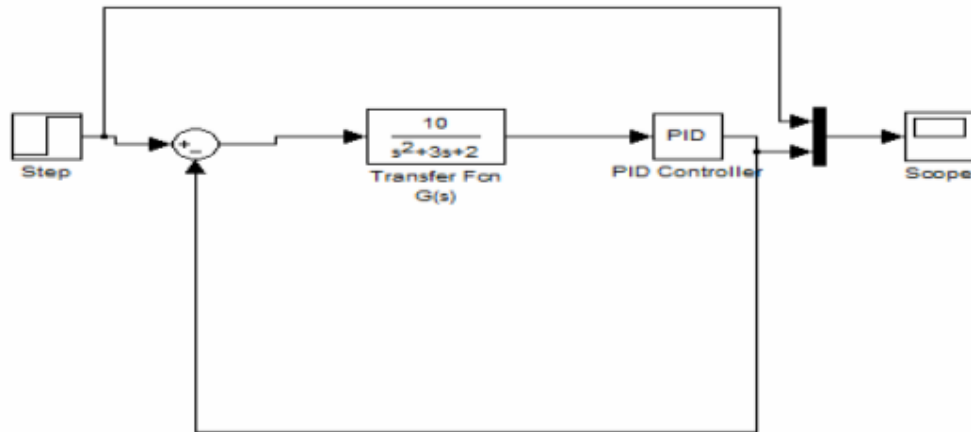


Figura 5.1 Diagrama de bloques en simulink

Esta función nos da como resultado con un tiempo muestreo de 0,1seg da los valores $k_p=1.454$, $k_i=1$ y $k_d=0.431$. Para dar una respuesta de un sistema estable como lo muestra la grafica.

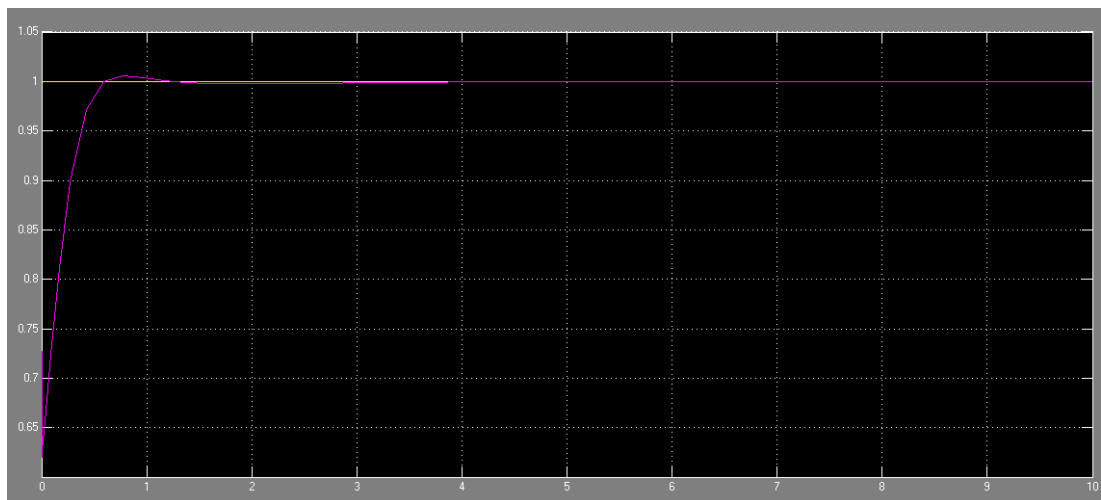


Figura 5.2 Respuesta al sistema

Llevando la función al tiempo discreto nos da como resultado la siguiente función:

$$D(Z) = 5.811 \frac{(Z - 0.905)(Z - 0.819)}{Z(Z - 1)} \quad (5.5)$$

$$D(z) = \frac{5.811z^2 - 10.017z - 4.30703}{z^2 - z} \quad (5.6)$$

Esta función se ingresa al toolbox de diseño de filtros (fdatool) donde se muestra su desarrollo para diferentes tipos de filtros digitales donde, se muestran los diferentes componentes que tiene cada filtro al mismo tiempo obtener las graficas de comportamiento, para este ejemplo se toma un filtro pasa baja para mostrar como se construye y configura el filtro digital con cada uno de sus componentes para reducir el efecto de la palabra finita.

5.2 VENTANA DE DISEÑO DE FILTRO Y GRAFICA DE ESPECIFICACIONES DEL FILTRO

En esta ventana muestra el grado en el que se trabaja el filtro, el tipo de respuesta, las frecuencias de corte y de trabajo al mismo tiempo se designa el metodo del filtro a trabajar ya sea FIR o IIR para nuestro caso por las características y por las ayudas prestadas por el Toolbox se cogio un FIR que tiene un grado de orden 2.

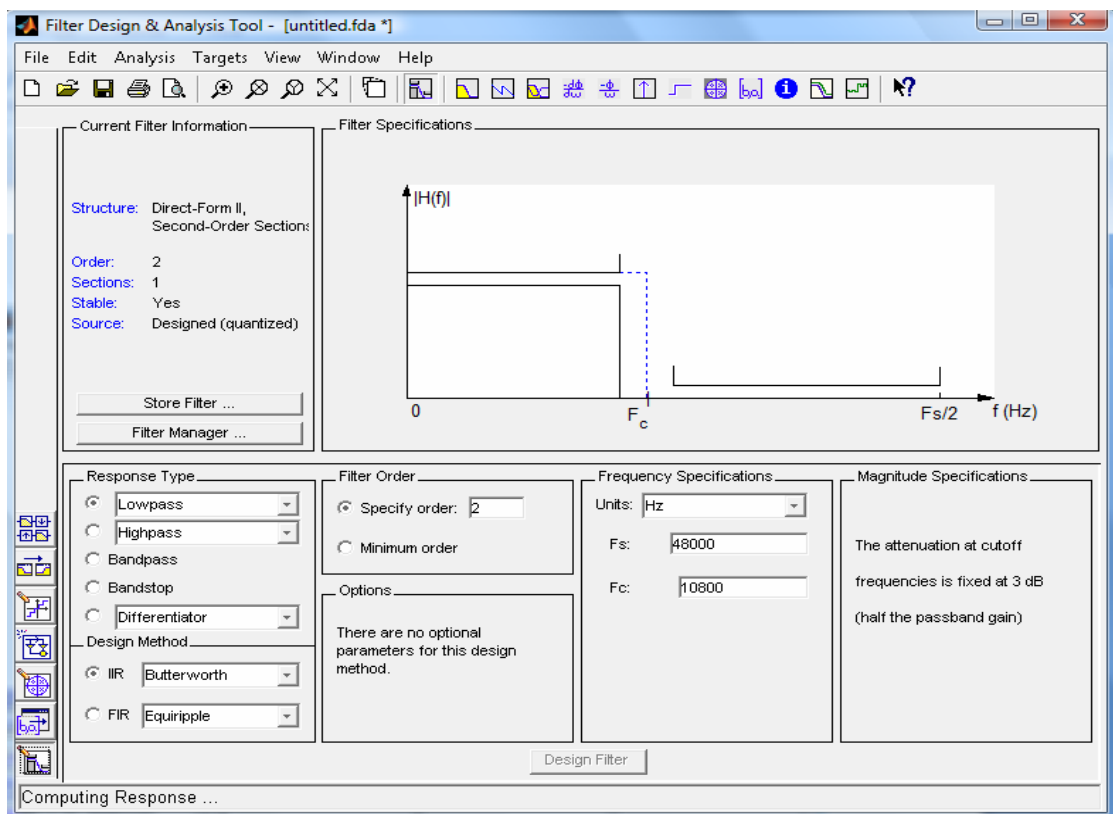


Figura 5.3 FDATOOL. Especificaciones del filtro.

5.3. VENTANA DE IMPORTACION DESDE EL WORKSPACE Y GRAFICA DE RESPUESTA EN MAGNITUD

En la siguiente ventana se observan los valores ingresados de polos y ceros de nuestra función como muestra la figura, se escoge la estructura del filtro en donde su frecuencia de muestreo que para nuestro ejemplo esta normalizada. En la parte superior se observa en la grafica de la respuesta del filtro en magnitud la que muestra los valores en magnitud de la grafica

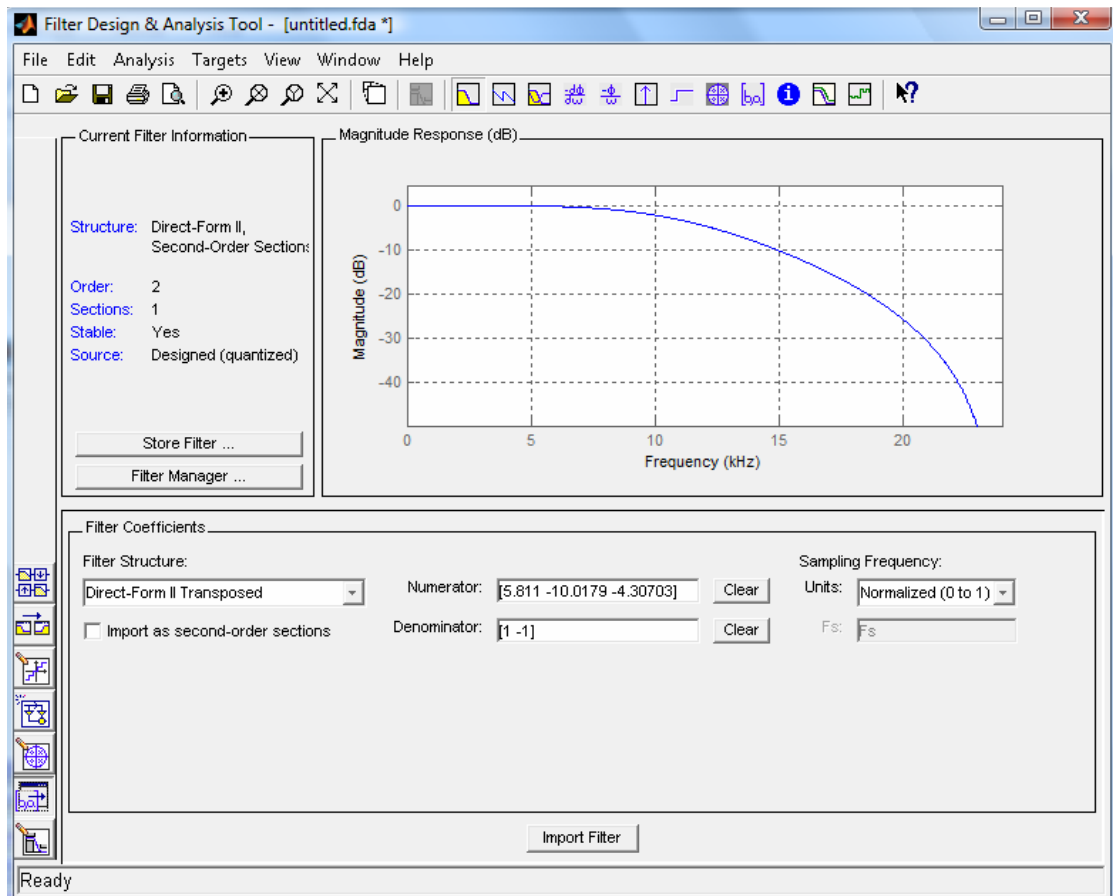


Figura 5.4 FDATool. Respuesta en magnitud del filtro.

5.4. VENTANA DE MODELACION Y OPTIMISACION DEL FILTRO CON GRAFICA DE FASE Y MAGNITUD

En esta ventana se muestra la grafica de fase y magnitud de la señal, en la cual se hace la comparación del comportamiento de ambas señales al mismo tiempo, en esta ventana se puede observar las opciones de generación de bloques para simulink y la construcción del filtro con elementos básicos que están en la ventana en la opción de optimización del filtro.

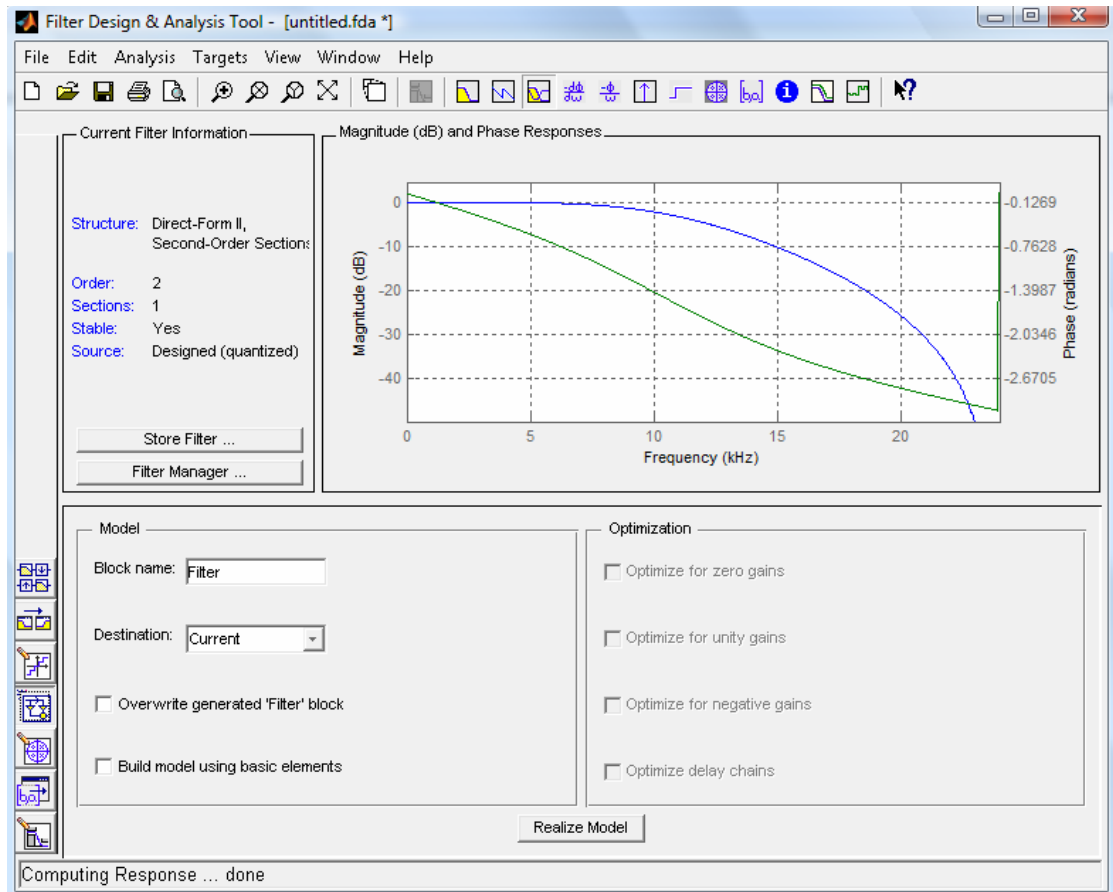


Figura 5.5 FDATool. Respuesta en magnitud y respuesta en fase del filtro.

5.5 VENTANAS DE KIT DE CUANTIZACION

Es la ventana en la cual contiene las opciones de cuantización para de esta manera llevar al filtro a una respuesta optima , esta ventana muestra la opción de escoger el tamaño de la palabra a trabajar como también la longitud de la fracción que se escoge para realizar la cuantización de esta manera se puede escoger longitud de la palabra de salida, esta ventana nos muestra la opciones de las funciones internas del filtro como lo muestra el submenú filter internals y el manejo de los coeficientes .

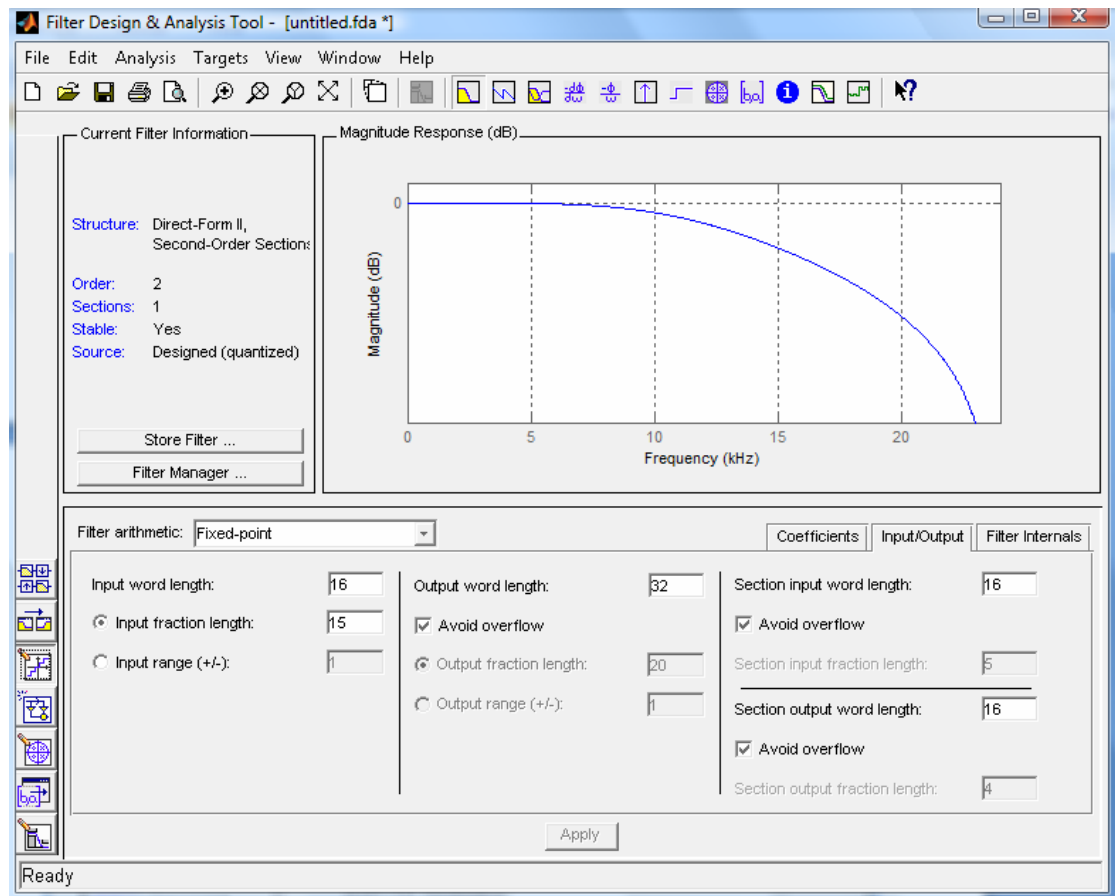


Figura 5.6 FDATool. Parámetros de cuantización

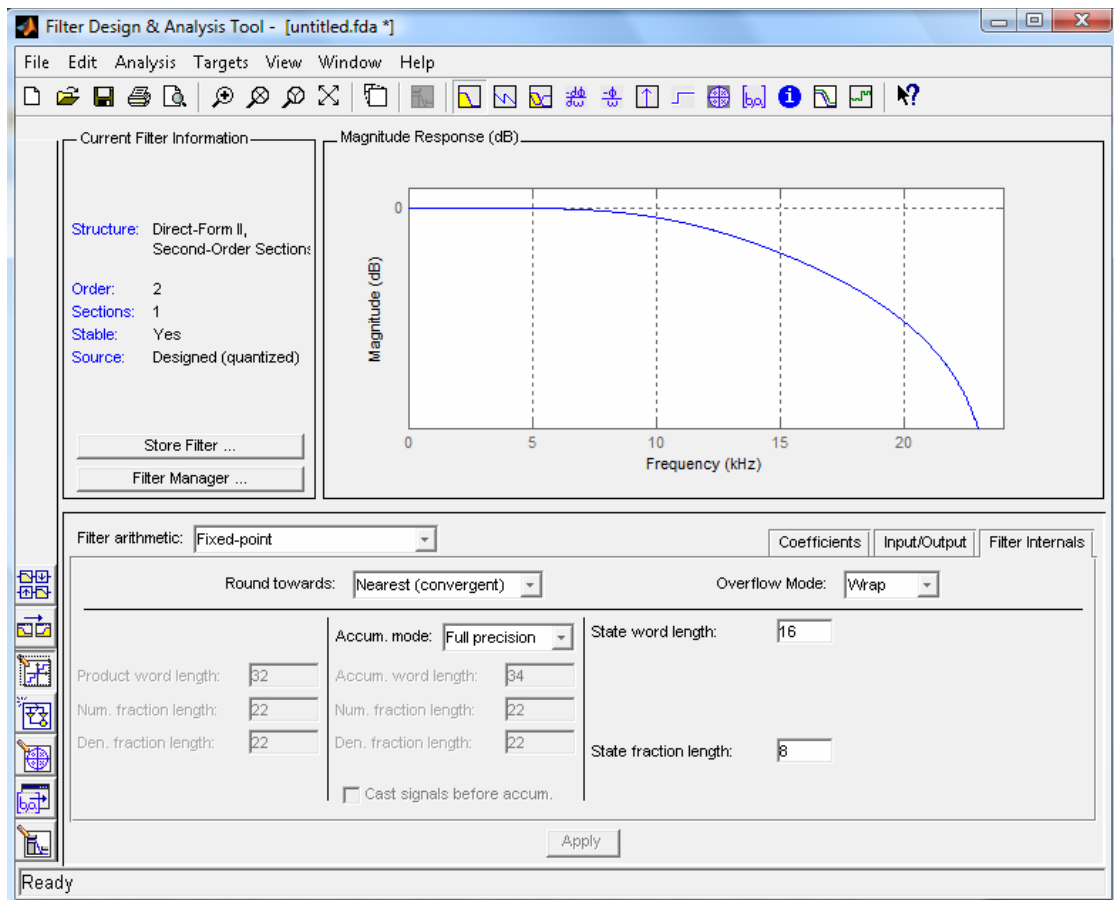


Figura 5.7 FDATool. Parámetros de cuantización

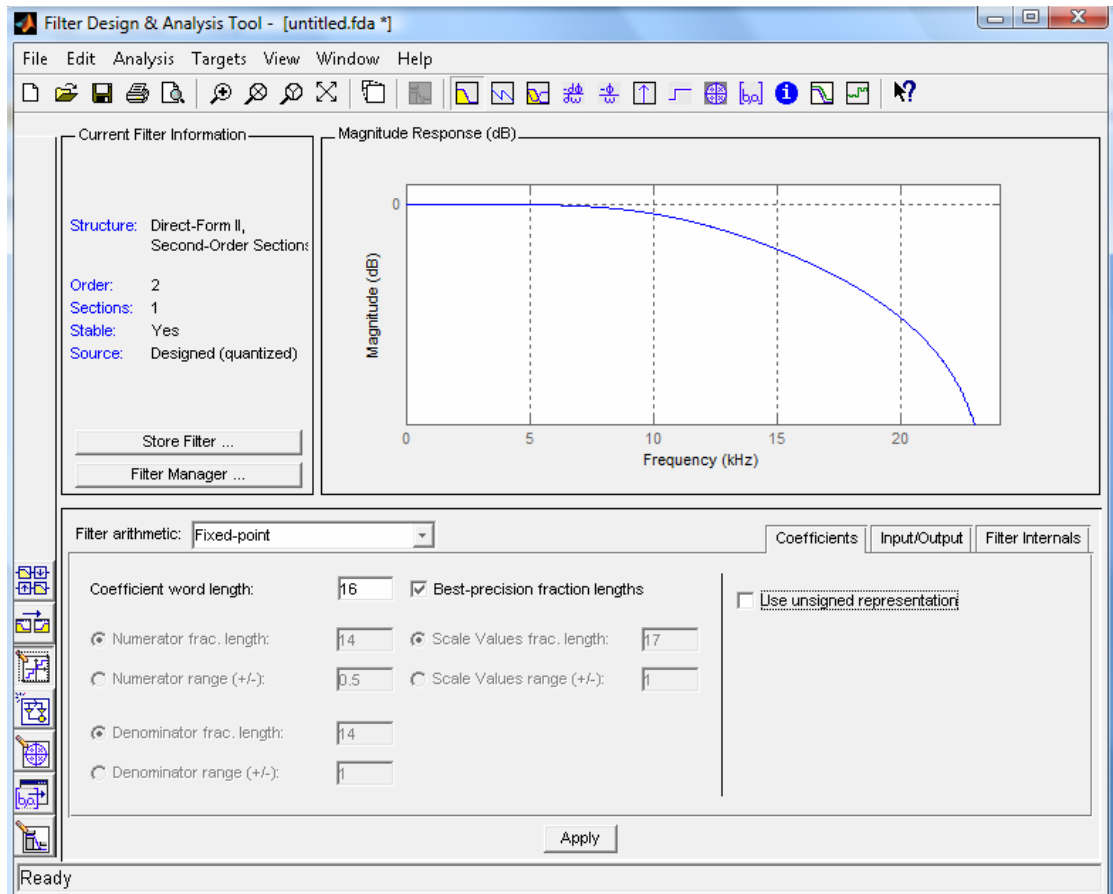


Figura 5.8 FDATool. Parámetros de cuantización

En la ventana anterior muestra una opción de escoger punto fijo y punto flotante dependiendo del filtro en el cual se este trabajando

5.6. VENTANA DE TRANSFORMACION DE FILTRO Y GRAFICA GRUPO DE ATRASO

Esta ventana contiene la grafica de retraso de la señal del filtro como se muestra en la parte superior de la ventana. En la parte inferior se encuentra ventanas de dialogo que permiten transformar nuestro filtro a cualquier otro filtro que se desee.

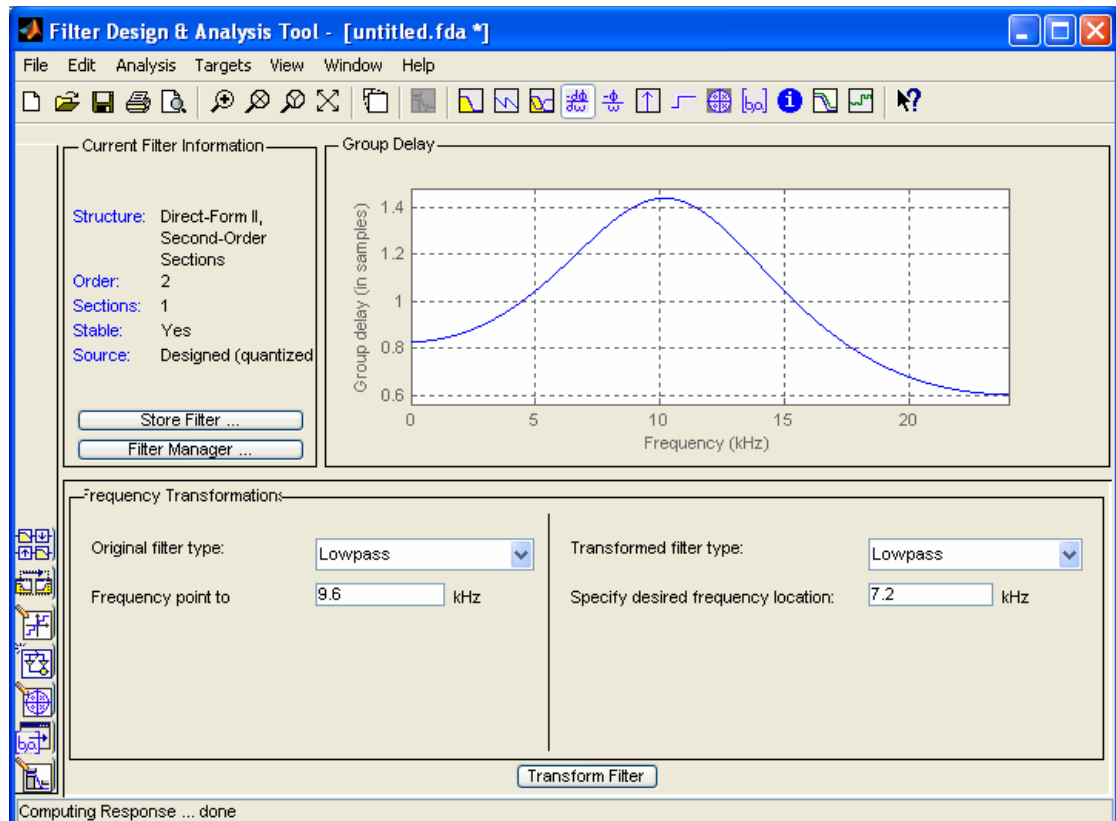


Figura 5.9 FDATool. Grupo de retardo

5.7 GRAFICA DE ATRASO DE FASE

Esta ventana da la información grafica de cómo esta la señal en atraso con respecto a su fase.

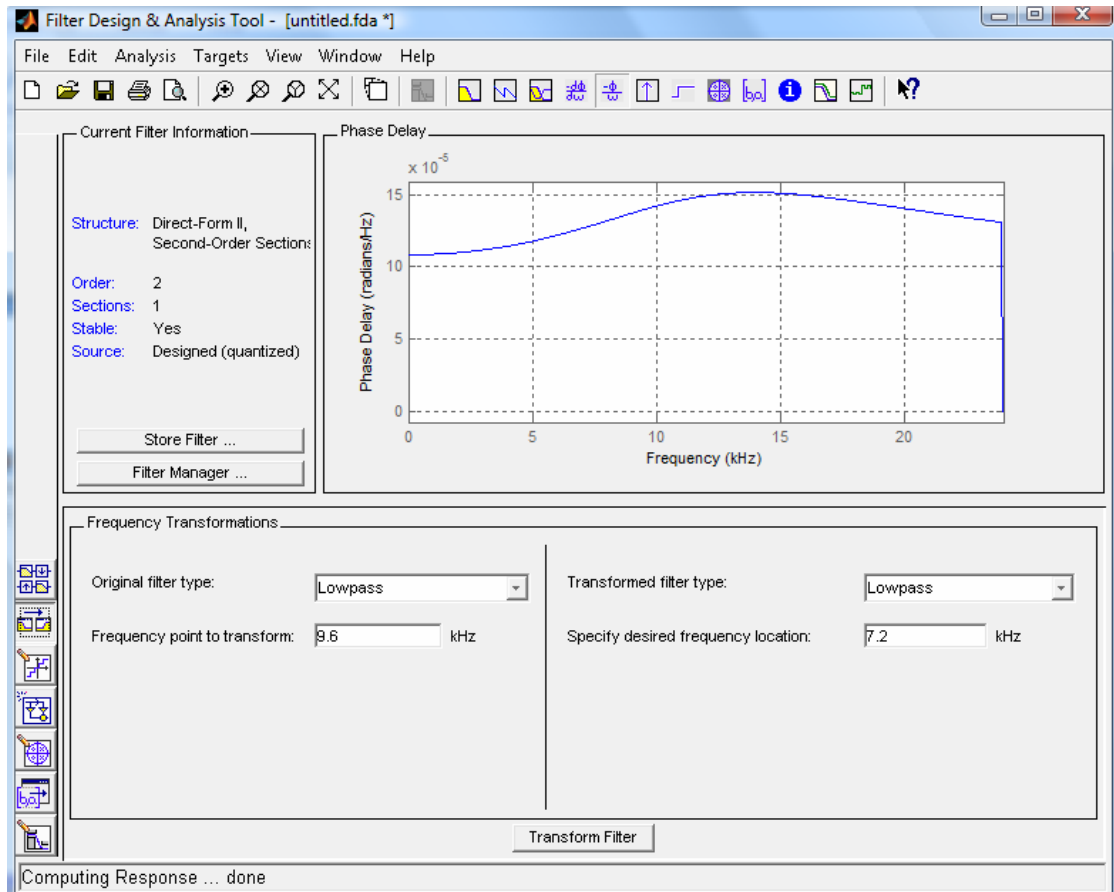


Figura 5.10 FDATool. Retardo de fase

5.8. VENTA DE CREACION DE UN FILTRO MULTIRATE Y GRAFICA DE RESPUESTA A UN IMPULSO

En la ventana se muestra la opción de crear un filtro multirate con sus diferentes características que están en el cuadro de diálogo de la ventana. La grafica muestra una respuesta a un pulso y sus valores en tiempo discreto.

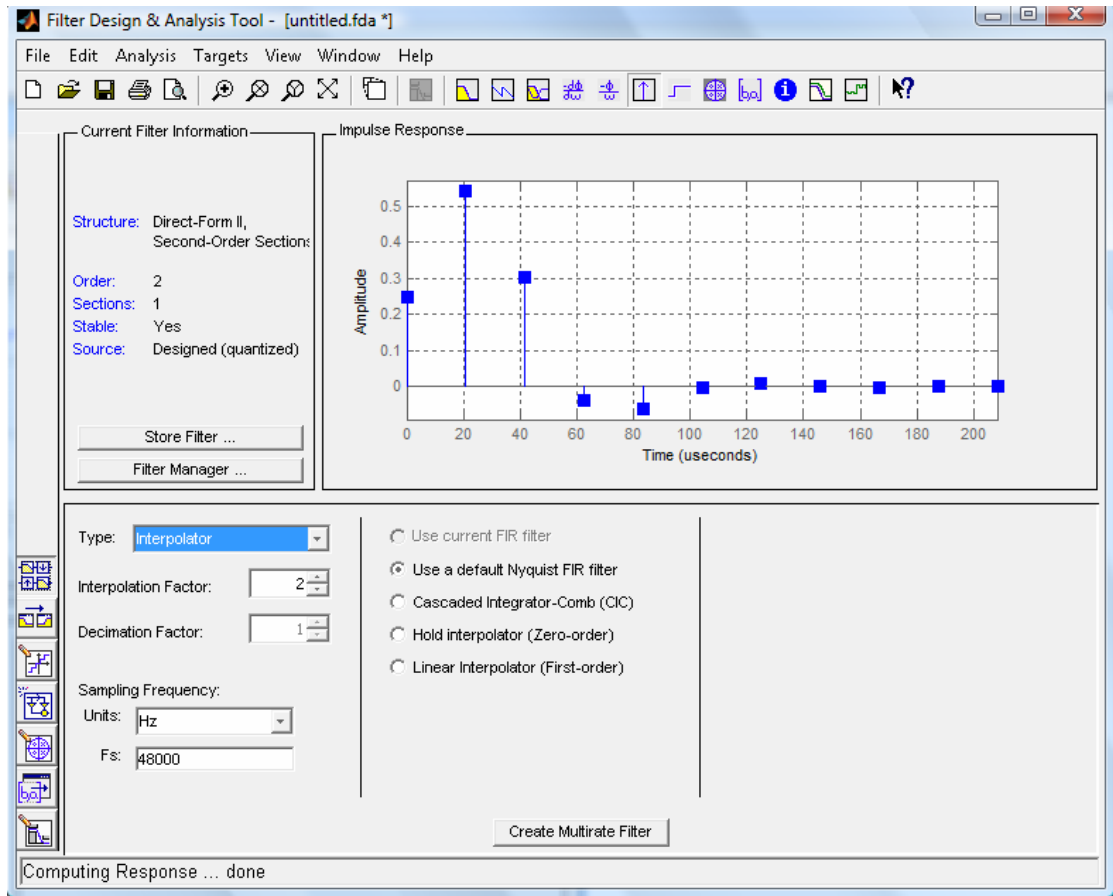


Figura 5.11 FDATool. Respuesta al impulso

5.9 GRAFICA DE RESPUESTA A UNA SEÑAL ESCALON

El grafico muestra el resultado con una señal de entrada escalón. El cuadro de dialogo está en cada una de las ventanas mostrada en el lado derecho superior es donde se encuentra la información del filtro creado. En el se encuentra el orden, su estabilidad, su estructura utilizada y su fuente en donde indica en que forma fue diseñado para nuestro caso e aparéese cuantizado.

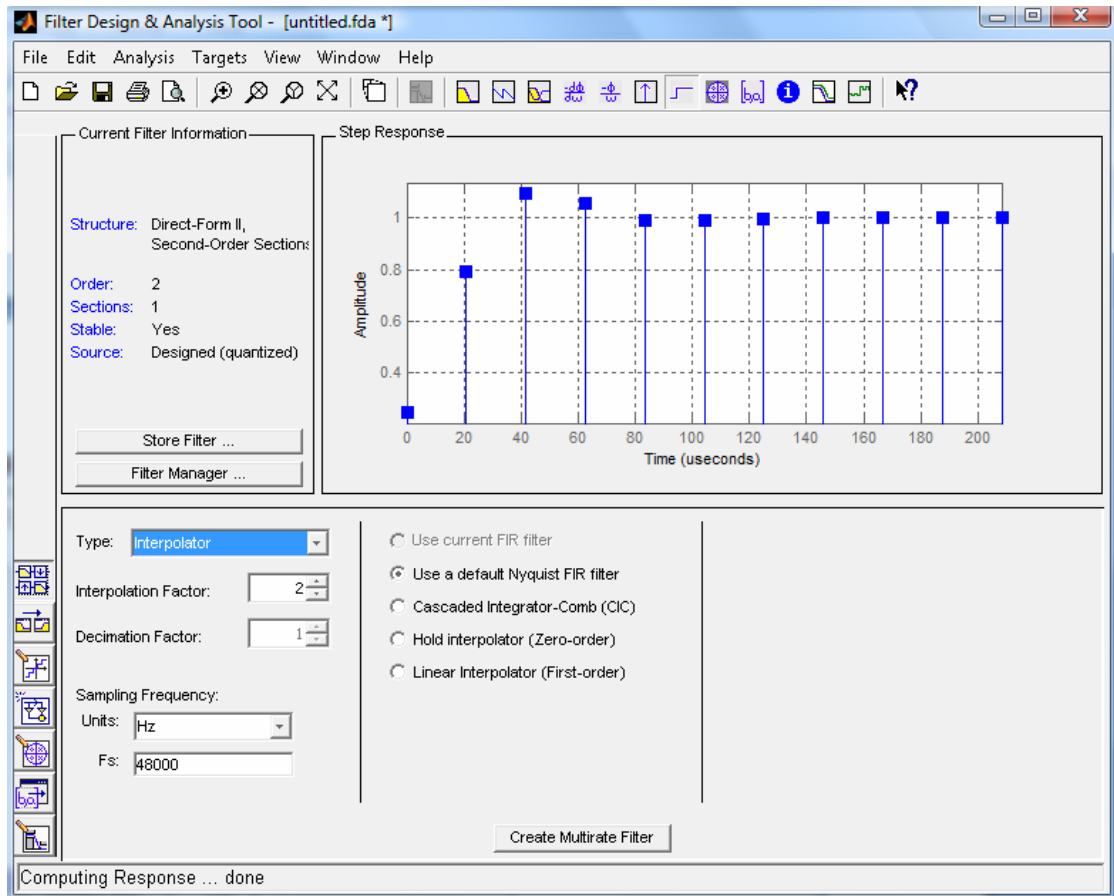


Figura 5.12 FDATool. Respuesta a la señal escalón

5.10. GRAFICA DE COMPORTAMIENTO DE POLOS Y CEROS

Como ya se mostró en un grafico anterior esta es otra opción en la cual también se muestra el comportamiento de los polos y los ceros al igual que su ubicación en el plano

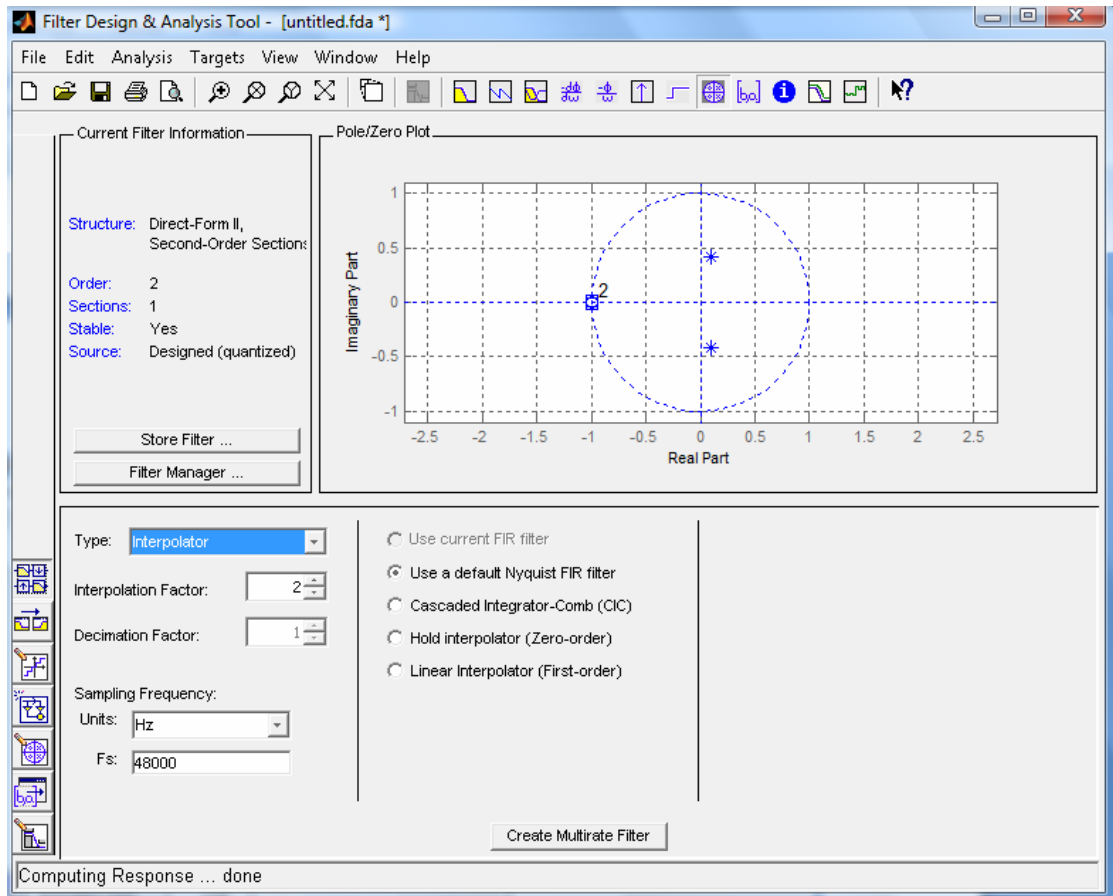


Figura 5.13 FDATool. Ubicación de polos y ceros.

5.11 COEFICIENTES DEL FILTRO

Este es el reporte que el Fdatool crea para que el diseñador tenga la información del filtro con sus diferentes secciones en las cuales se especifican las operaciones realizadas

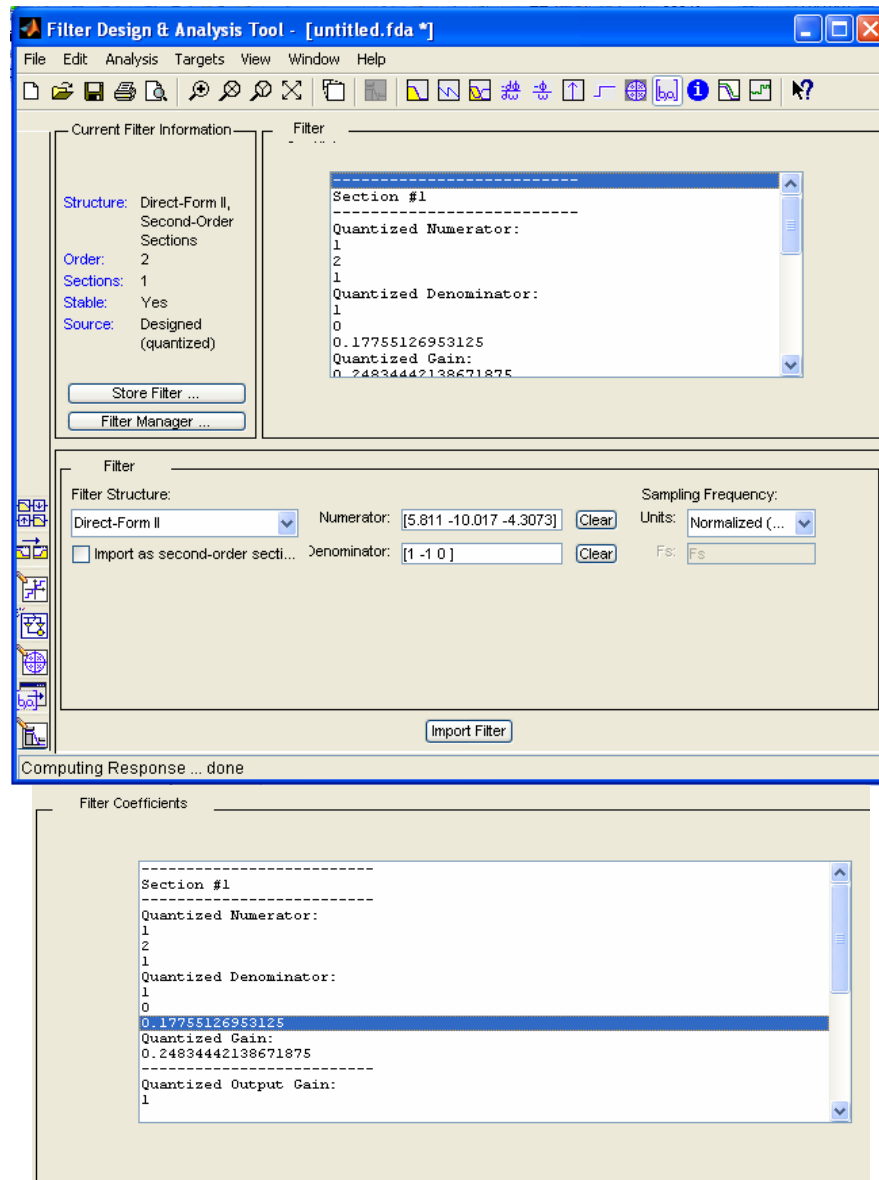


Figura 5.14 FDATool. Coeficientes del filtro. Ubicación de polos y ceros.

5.12. REPORTE DEL FILTRO

Es te es otro reporte que el fdatoool proporción en el que aparecen los datos del filtro con sus componentes activos

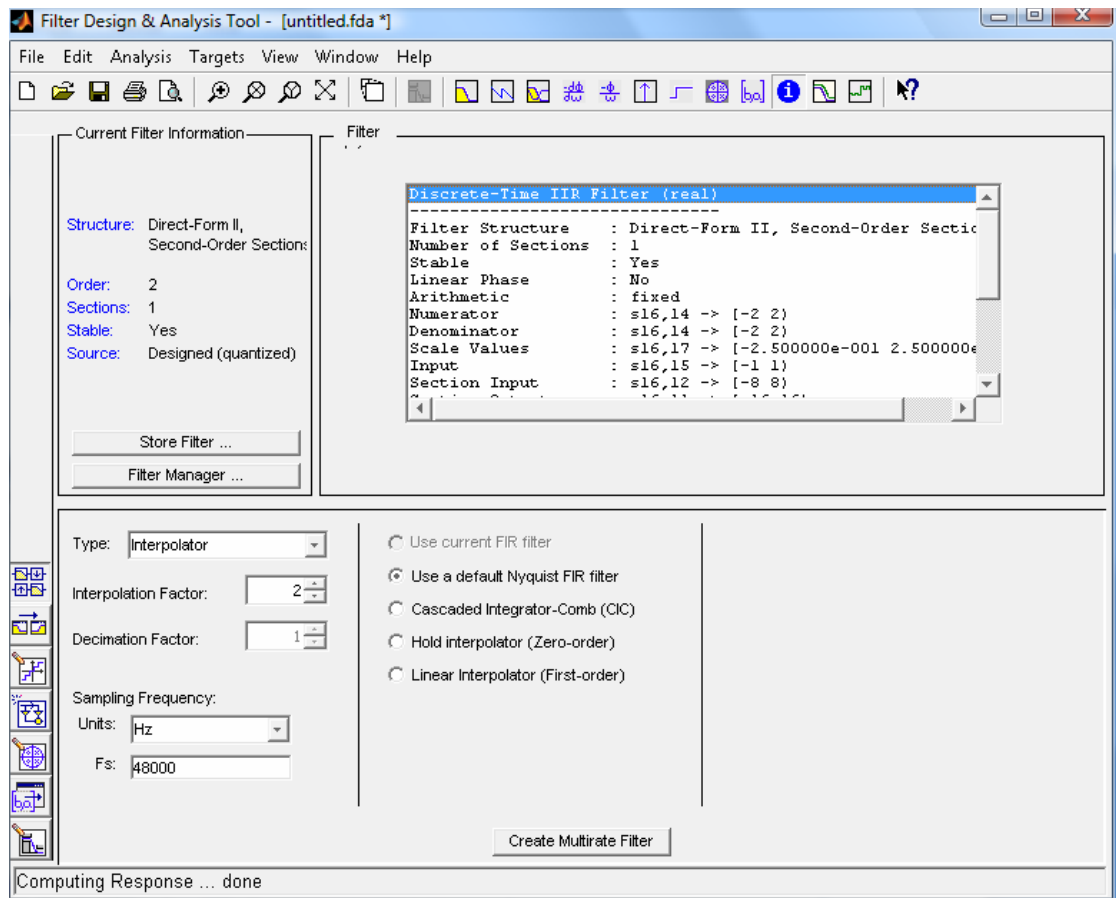


Figura 5.15 FDATool. Reporte del filtro.

5.13. GRAFICA DE RESPUESTA ESPERADA

Esta es la grafica que se da para el rendimiento del filtro en relación con su magnitud

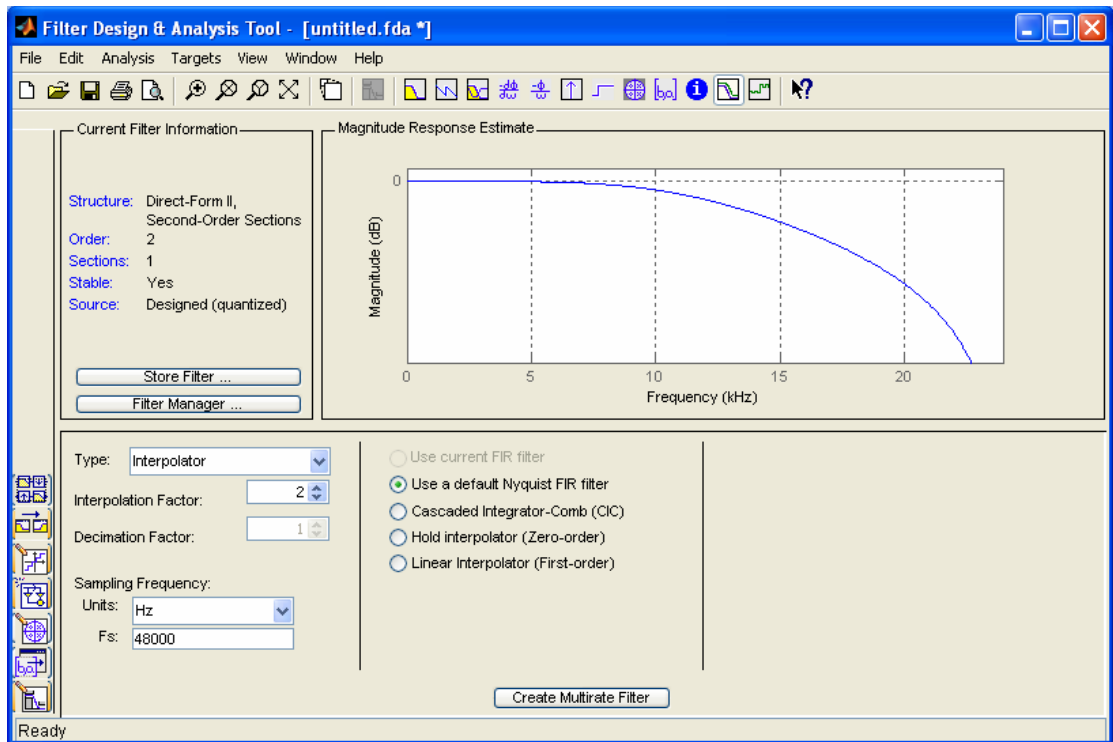


Figura 5.16 FDATool. Respuesta en magnitud estimada.

5.14. GRAFICA DE POTENCIA DEL ESPECTRO DEL RUIDO

En esta grafica que nos proporciona la ventana de potencia del espectro del ruido se puede analizar en un estudio mas profundo .

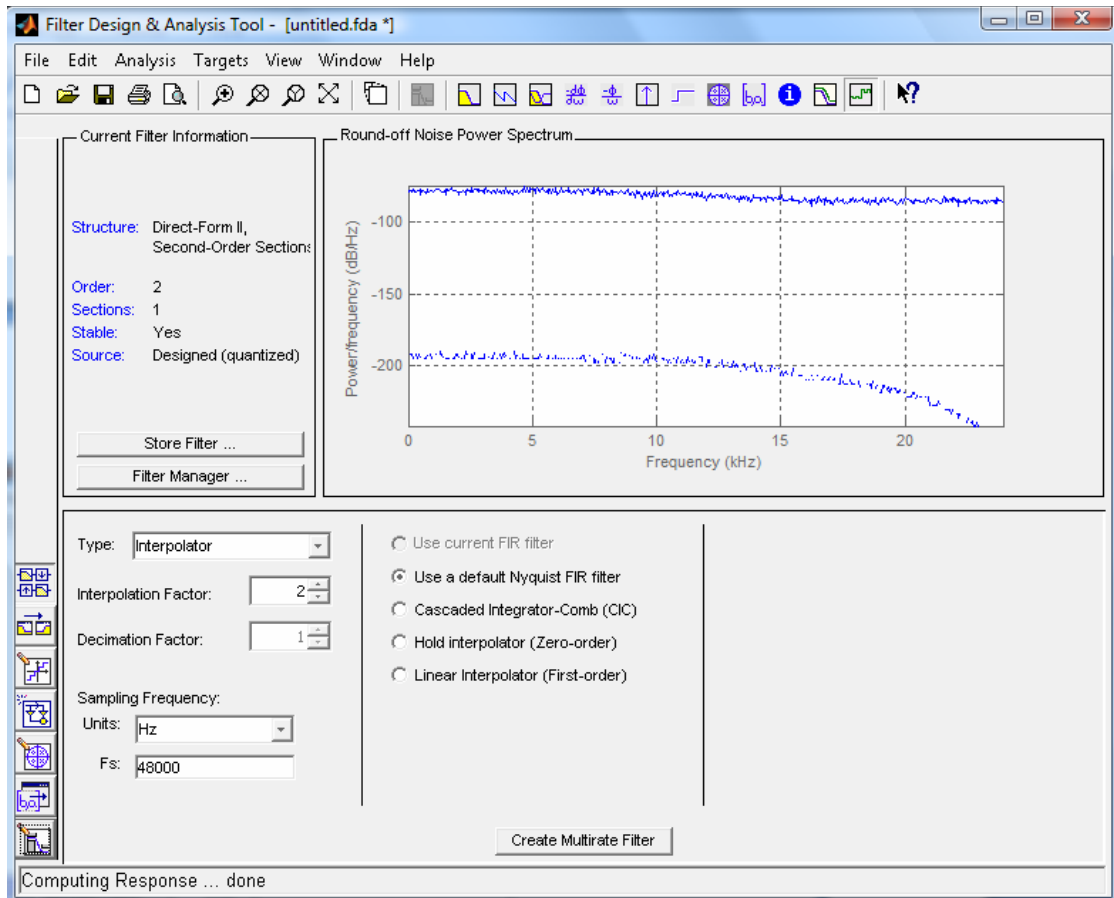


Figura 5.17 FDATool. Espectro de potencia del ruido redondeado

5.15 ANALISIS

Para cada elemento de ventanas y de graficas utilizadas en el desarrollo del filtro se muestra como cada parámetro utilizado, representan los datos en los cuadros de reporte como gráficamente. El comportamiento del pid a si como su estabilidad esta comprobado por los datos suministrados por el las diferentes características mostradas por el pid trabajándolo en un sistema discreto, nos da la capacidad de trabajar con una herramienta con fiable y segura que a su vez proporciona ayuda en un desarrollo total del filtro al igual que muestra los problemas que puedan ocurrir, los diferentes parámetros que al utilizar el filtro nos dicen si son viables o no.

6. CONCLUSIONES

- Los microprocesadores contienen sistemas numéricos con palabra finita en cada uno de sus componentes; al aplicarles los métodos para atenuar los efectos de palabra finita, estudiados en esta investigación, se consigue reducir los errores que aparecerían normalmente; se tendrá un comportamiento mejor del sistema en el cual está instalado el microprocesador.
- Cuando se diseñan filtros digitales se debe tener en cuenta la cuantización. El estudio realizado mostró que cada filtro digital, ya sea FIR o IIR, tiene características que deben tenerse en consideración durante la implementación.
- Los Toolbox de Matlab son una herramienta que minimiza los efectos de la palabra finita con poco esfuerzo por parte del diseñador de filtros; presentan una ayuda gráfica en la cual el usuario observa el comportamiento de cada componente del filtro.
- En los sistemas de control implementados en sistemas numéricos de punto fijo en los que se tenga una longitud de palabra finita para representar la magnitud de la señal o una ganancia es necesario que el diseñador investigue el desempeño del sistema cuando sus parámetros solo pueden realizarse con un conjunto finito de números; de esta manera el diseñador pueda controlar satisfactoriamente el efecto de la palabra finita.
- Del estudio realizado sobre los controladores PID se puede concluir que cada controlador tiene una forma de comportamiento diferente para cada caso; para estos casos el diseñador debe tener en cuenta los parámetros de la estructura en la que se está llevando a cabo para utilizar una estructura adecuada y una mejor respuesta en cada caso.
- El estudio de los efectos de la palabra finita es importante porque permite analizar, entender y reducir los efectos que se presentan en la aplicación de filtros digitales.
- Se recomienda la continuación de esta tesis, para la aplicación en tiempo real a un sistema físico específicamente a cualquiera de los módulos de control existente en la facultad de electrónica.
- Uno de los problemas encontrados al realizar el estudio es la falta de bibliografía existente en la universidad acerca de este tema.
- Es recomendable al utilizar los toolbox de diseños de filtros la realización de una interfaz gráfica donde se deben presentar el análisis de la respuesta al sistema en simulink y la utilización de los toolbox de diseño de filtros y de punto fijo.

REFERENCIA BIBLIOGRAFICAS

Constantine H. Houppis, Ph. D, Gary B. Lamont, Ph. D. "DIGITAL CONTROL SYSTEMS THEORY, HADWARE, SOFTWARE". Second Edition. Professors of Electrical Engineering. School of Engineering. Air Force Institute of Technology. Wriyth-Patterson Air Force Base, Ohio.

Sanyit K. Mitra. "DIGITAL SIGNAL PROCESSING. A COMPUTER-BASED APPROACH". Second Edition. Departament of Electrical and Computer Engineering. University of California. Santa Barbara.

Charles L. Phillips. H. Troy Nagle. "DIGITAL CONTROL SYSTEM. ANALYSIS AND DESIGN". Third Edition. Prentice Hall. 1995.

Vijay K. Madisetti, Douglas B. Williams. "DIGITAL SIGNAL PROCESSING. HANDBOOK". Center for Signal and Image Processing. School of Electrical on Computer Engineering. Georgia Institute of Techonology. Atlanta, Georgia.

REFERENCIAS BIBLIOGRAFICAS DE INTERNET

Ing. Lino Evgueni Coria Mendoza. Instituto tecnológico y de Estudios Superiores de Occidente. México.

<http://www.mathworks.com/>

ANEXO A

“MANUAL TOOLBOX DE PUNTO FIJO”

A.1 ¿QUÉ ES EL TOOLBOX DE PUNTO FIJO?

El Toolbox de punto fijo proporciona datos de tipo “punto fijo” en MATLAB y permite el desarrollo de algoritmos mediante el uso de la aritmética de punto fijo. El Toolbox de punto fijo permite la creación de los siguientes tipos de objetos.

- FI: Define un objeto numérico de punto fijo en el Work – Space de Matlab. Cada objeto “FI” está compuesto por un valor de datos, por un objeto fimath y un objeto de tipo numérico.
- FIMATH: Rige como sobrecarga cuando se realizan operaciones aritméticas con los objetos FI.
- FIPREF: Define el despliegue, la explotación y las preferencias sobre los tipos de datos de los objetos FI.
- NUMERICTYPE (TIPO NUMERICO): Define el tipo de datos y la escala de atributos de los objetos FI.
- CUANTIZAR: Cuantiza conjunto de datos.

El toolbox de punto fijo proporciona:

- La capacidad para definir los datos de tipo “punto fijo”, el escalamiento, el redondeo y los métodos de overflow en el workspace de matlab.
- BIT: Simulación real y compleja.
- Aritmética básica de punto fijo: Operadores aritméticos de (+, -, *, /) solo para puntos binarios y señales reales (signed). La función de división es solo para señales de punto binario.
- Longitud de palabra arbitraria hasta intmaxbits.
- Registro de mínimos, máximos, overflow, underflows.
- Tipo de datos con sobrecarga sencillo, sobrecarga doble, o escalas dobles.
- Conversión entre binarios, hexadecimal, conversiones dobles y conversiones entre enteros.
- Relacional, lógico, y operaciones de modo BIT.
- Matriz de funciones tales como ctranspose y horzcat.
- Funciones estadísticas como máximas y mínimas.
- Inter-operabilidad con simulink, proceso Blockset de señales, Embedded Matlab, y Toolbox de diseño de filtros.
- Compatibilidad con el workspace de simulink y la forma de bloques del workspace.

A.2 LICENCIAS

Se puede utilizar el objeto FI con la propiedad DataType y un conjunto de punto fijo y la licencia del Toolbox de punto fijo esta apagada. La licencia del Toolbox de punto fijo se obtiene de la siguiente forma:

- Utilice el objeto FI solo con otros DataType dobles.
- Crea el objeto FI cuando la prioridad FIPREF Logging Made esta en ON, incluyendo los objetos FI con DataType doble.
- Cargue un Mat-File que contenga un objeto FI con la propiedad DataType a único, booleano, doble escrito o fijo.

Se puede evitar la compra de la licencia del Toolbox de punto fijo cuando se trabaja con el código del Toolbox de punto fijo mediante el establecimiento de la propiedad Fipref DataType Override a Truedobles.

A. 3 OBTENCION DE AYUDA EN LA LINEA DE COMANDOS DE MATLAB

Para obtener la ayuda en la línea de comando se hace:

```
>> help objecthame
```

Por ejemplo:

```
>>help fi  
>>help fimath  
>>help fipref  
>>help numerictype  
>>help quantizer
```

Para obtener la ayuda del Toolbox de punto fijo en el comando linea de todas las funciones:

```
>> help embedded.fi\functionname
```

Por ejemplo:

```
>>help embedded.fi/abs  
>>help embedded.fi/bitset  
>>help embedded.fi/sqrt
```

Para llamar el archivo de ayuda de la documentación para el Toolbox de punto fijo de funciones en la línea de comandos de matlab:

```
>>doc.fixedpoint\functionname
```

Por ejemplo :

```
>>doc fixedpoint/int
>>doc fixedpoint/add
>>doc fixedpoint/savefipref
>>doc fixedpoint/quantize
```

A.4 AJUSTE DE PANTALLA

En el Toolbox de punto fijo la visualización de los objetos FI esta determinada por el objeto Fipref, el ejemplo general del código objeto Fi es:

- NumberDisplay: RealWorldValue
- Numeric Type Display: full
- FimathDisplay: none

Por ejemplo:

```
>> p = fipref('NumberDisplay', 'RealWorldValue','NumericTypeDisplay', 'full',
'FimathDisplay', 'none')
p =
    NumberDisplay: 'RealWorldValue'
    NumericTypeDisplay: 'full'
    FimathDisplay: 'none'
    LoggingMode: 'Off'
    DataTypeOverride: 'ForceOff'
>>a = fi(pi)
a =
    3.1416
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 16
    FractionLength: 13
```

En otros casos, se muestra en la pantalla el objeto fimath

- NumberDisplay: RealWorldValue
- Numeric Type Display “full”
- Firmath Display: “full”

Por ejemplo:

```
>>p = fipref('NumberDisplay', 'RealWorldValue','NumericTypeDisplay', 'full',
'FimathDisplay', 'full')
p =
```

```

    NumberDisplay: 'RealWorldValue'
    NumericTypeDisplay: 'full'
    FimathDisplay: 'full'
    LoggingMode: 'Off'
    DataTypeOverride: 'ForceOff'
>>a = fi(pi)
a =
    3.1416
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 16
    FractionLength: 13
    RoundMode: nearest
    OverflowMode: saturate
    ProductMode: FullPrecision
    MaxProductWordLength: 128
    SumMode: FullPrecision
    MaxSumWordLength: 128
    CastBeforeSum: true

```

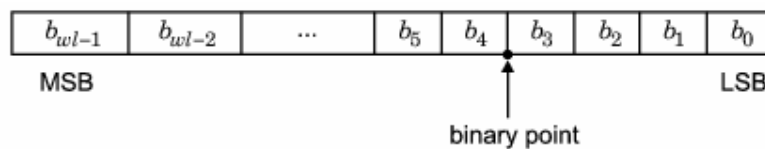
A.5 TIPOS DE DATOS PUNTO FIJO

En el hardware digital, los números se almacenan en palabras binarias. Una palabra binaria es una secuencia de longitud fija de bits (1 y 0).

Como componentes de hardware o software las funciones interpretan la secuencia de unos y ceros es definido por el tipo de datos. Los números binarios son representados ya sea como tipo de datos, punto fijo o punto flotante.

El tipo de dato punto fijo se caracteriza por la longitud de palabra en bits, la posición del punto binario y si tiene signo o no. La posición del punto binario es el medio por el cual los valores de punto fijo se escala e interpreta.

Por ejemplo, una representación binaria de un numero de punto fijo generalizada ya sea con signo o sin signo es:



Donde:

- b_i : dígito i -ésimo binario.
- wl : longitud de la palabra en bits.
- b_{wl-1} : bit más significativo (MSB)
- b_0 : bit menos significativo (LSB)

Los tipos de datos de punto fijo pueden tener o no tener signos. El signo binario del punto fijo suelen ser representados de tres maneras.

- Signo Magnitud
- Complemento a uno
- Complemento a dos

El complemento a dos es la representación del signo de los números de punto fijo más común y es la única representación utilizada en el Toolbox de punto fijo.

A. 6 ESCALAMIENTO

Los números de punto fijo pueden ser codificados de acuerdo con el siguiente régimen.

$$\text{Valor de la población real} = (\text{pendiente} \times \text{integrados}) + \text{escalón}$$

Donde la pendiente puede ser expresada por:

$$\text{Pendiente} = \text{Fracción de la pendiente} \times 2^{\text{exponente fijo}}$$

El entero a veces se llama entero almacenado. Este es el número binario, en el cual el punto binario supone como el extremo derecho de la palabra.

En el Toolbox punto fijo el valor negativo del exponente fijo se le conoce como la FRACCION DE LONGITUD.

La pendiente y el escalón en conjunto, representan la ampliación del número de punto fijo. En un número con escalón cero, solo la pendiente afecta la escalización.

Un número punto fijo que es solo escalado por la posición del punto binario es equivalente a un número (pendiente sesgada) con representación que tiene un escalón igual a cero y una fracción de pendiente igual a uno. Esto se conoce como escalización de sólo punto binario ó escalización power of two.

$$\text{Valor real de la palabra} = 2^{\text{exponente fijo}} \times \text{entero}$$

Ó

$$\text{Valor real de la palabra} = 2^{-\text{fracción de la longitud}} \times \text{entero}$$

El Toolbox de punto fijo soporta tanto escalamiento de punto único binario como el escalamiento de pendiente sesgada.

Nota: para ejemplos de escalamiento de punto único binario se puede ver el demo del Toolbox de punto fijo “Escalización del punto binario fi”-

A.7 RANGO Y PRECISIÓN

Nota: Se debe prestar atención al rango y a la precisión del punto fijo y a la escala de datos que se elijan, ya que con esto se sabe si el método provoca redondeo o puede ocurrir overflow o underflow.

RANGO: El rango es el lapso de números que existe entre los tipos de datos del punto fijo y la escalización que se pueda representar.

El rango de los números representados por el punto fijo de complemento a dos es:

number of word length wl , scaling S and bias B is illustrated below:



Donde:

- WL : Longitud de palabra numerica.
- S : Escalamiento.
- B : Escalón

Para los números de punto fijo con signo o sin signo, el número fundamental de bits es: 2^{wl} .

Por ejemplo, en el complemento a dos, los números negativos y el cero deben ser representados, de modo que el valor máximo es: $2^{wl-1} - 1$. Porque sólo hay una representación de cero. Hay una desigualdad entre los números positivos y negativos, esto significa que hay una representación de -2^{wl-1} pero no por 2^{wl-1}

Para pendiente= 1, Escalón= 0

A.7.1 MANIPULACION DEL OVERFLOW

Debido a que el tipo de datos punto fijo representa a los números dentro de un rango, el Overflow y el underflow puede ocurrir si el resultado de una operación es mayor o menor que el número en ese rango.

El Toolbox de punto fijo, permite la recapitulación del overflow y el underflow. La saturación representa el overflow positivo como el mayor número positivo del rango a utilizar y el overflow negativo como el mayor número negativo del rango que se use cuando se crea el objeto Fi, en el Toolbox de punto fijo el overflow es saturado.

La propiedad Overflow Made por defecto sature al objeto Fimath.

Se puede acceder al Overflow y al Underflow mediante la propiedad Logging Mode de los objetos Fipref.

A.7.2 PRESICION

La precisión de un número de punto fijo es la diferencia entre los valores sucesivos representables por su tipo de dato y la ampliación que es el valor del bit menos significativo. El valor del bit menos significativo y el número de precisión, es determinado por el número fraccional de bits. Un valor de punto fijo puede ser representado dentro de la mitad de la precisión de sus datos y el tipo de escala.

A.8 METODO DE REDONDEO

Una de las limitaciones de la representación de números finitos con precisión que no todos los números en el rango disponible se puede representar con exactitud.

Cuando el resultado de un cálculo de punto fijo es un número que no puede ser representado exactamente por el tipo de datos y la escalización que se está utilizando, se pierde precisión.

El método de redondeo debe ser utilizado para emitir el resultado de un número de manera representable. El Toolbox de punto fijo admite los siguientes métodos de redondeo:

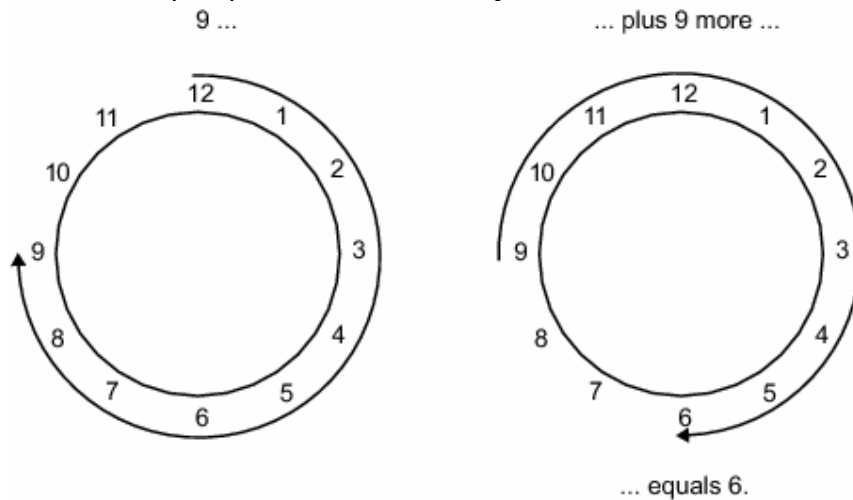
- Ceil: Redondea el valor más cercano del positivo infinito.
- Convergent: Redondea al entero más cercano. En caso de igualdad se redondea al entero más cercano aun almacenado. Este es el método menos sesgado del redondeo de las cifras proporcionadas por el Toolbox de punto fijo.
- Fix: Redondea al entero más cercano en dirección de cero.
- Floor: Es equivalente al complemento a dos del truncamiento, redondea al entero más cercano del negativo infinito.
- Nearest: Redondea al entero más cercano representable en caso de igualdad redondea al número entero más cercano al positivo infinito. Este es el método de redondeo por defecto para la creación de objetos Fi y para la aritmética de Fi.

- Round: Redondea al entero mas cercano. Si existe igualdad con un número positivo redondea al entero más cercano al positivo infinito.

Si existe igualdad con un número negativo redondea al entero más cercano al negativo infinito.

A.9 MODULO ARIMETICO

Las matemáticas binarias se buscan en el modulo aritmético. El modulo aritmético solo un conjunto de los números y envuelven todos los resultados de los cálculos que quedan fuera del conjunto dado.



Del mismo modo la matemática binaria sólo puede utilizar los números del 0 al 1 y todos los resultados que caen fuera de este caen dentro del círculo de 0 a 1.

A.10 COMPLEMENTO A DOS

El complemento a dos es una manera de representar un número binario. En el complemento a dos, los números positivos siempre empiezan con un cero y los números negativos empiezan con un uno. Si el primer bit de un número es un complemento a dos es cero, el valor se obtiene mediante el cálculo del valor estándar de dicho número.

Si el primer bit de un número de complemento a dos es 1, el valor se obtiene así:

$$01 = (0 + 2^0) = 1$$

$$11 = \left((-2^1) + (2^0) \right) = (-2 + 1) = -1$$

Para calcular el negativo de un número binario utilizando el complemento a dos:

- Tome el complemento a uno.
- Sume un 1 binario.
- Deseche el carry mas alla de la longitud de palabra a usar

A.11 ADICION Y SUSTRACCION

La suma de números de punto fijo requiere que el punto binario de los sumandos estén alineados. Por ejemplo:

$$\begin{array}{r} 010010.1 \quad (18.5) \\ +0110.110 \quad (6.75) \\ \hline 011001.010 \quad (25.25) \end{array}$$

La resta de punto fijo es equivalente a la suma, mientras que utilizamos el complemento a dos para el valor negativo. Por ejemplo, considere la resta de 0110.110 (6.75) y 010.010,1 (18.5)

$$\begin{array}{r} 010010.100 \quad (18.5) \\ -0110.110 \quad (6.75) \\ \hline \end{array} \xrightarrow[\text{and sign extension}]{\text{two's complement}} \begin{array}{r} 010010.100 \quad (18.5) \\ +111001.010 \quad (-6.75) \\ \hline 1001011.110 \quad (11.75) \end{array}$$

Carry bit
is discarded

El objeto Fimath tiene un valor de 1(verdadero), para la propiedad CastBeforeSum. Esto arroja la suma de los sumandos antes de la adición. Por lo tanto, no es necesario seguir cambiando el punto binario en la línea de la suma.

Si CastBeforeSum tiene un valor de 0(falso), se añaden los sumandos manteniendo una total precisión. Después de la suma se cuantiza.

A.12 MULTIPLICACION

La multiplicación en el complemento a dos de números de punto fijo es análoga a la multiplicación decimal común con la excepción de que los resultados deben tener en el lado izquierdo el signo – extendido.

Por ejemplo, considere la multiplicación de 10.11 (-1.25) con 011 (3):

$$\begin{array}{r}
 10.11 \text{ } (-1.25) \\
 \times 011 \text{ } (3) \\
 \hline
 11011 \\
 1011 \\
 \hline
 1100.01 \text{ } (-3.75)
 \end{array}$$

The extra 1 is the result of necessary sign extension.

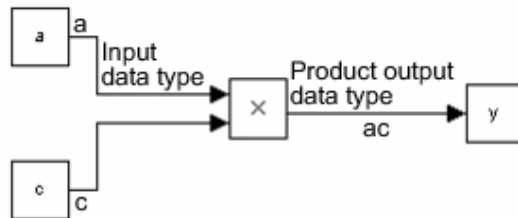
The number of fractional bits of the result is the sum of the number of fractional bits of the factors.

A.12.1 MULTIPLICACION DE LOS TIPOS DE DATOS

Los siguientes diagramas muestran los tipos de datos utilizados para la multiplicación de punto fijo. Los diagramas ilustran las diferencias entre los tipos de datos utilizados por: (real-real, complejo real, complejo – complejo) multiplicaciones.

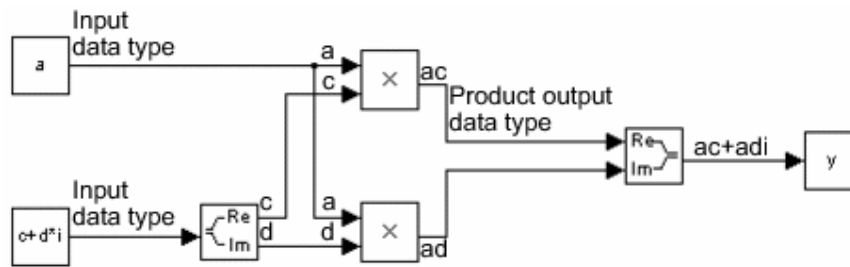
A.12.1.1MULTIPLICACION REAL – REAL

El siguiente diagrama muestra los tipos de datos utilizados en la multiplicación de dos números reales del Toolbox de punto fijo. El resultado de esta multiplicación es el tipo de datos del producto y es regido por la propiedad FimathProductMode.



A.12.1.2 MULTIPLICACION REAL – COMPLEJO

El siguiente diagrama muestra los tipos de datos utilizados en la multiplicación de un numero real y uno complejo del Toolbox de punto fijo.



A.12.1.3 MULTIPLICACION CON FIMATH

- F: Fimath ("ProductMode", "FullPresicion", "SumMode", "FullPrecision").
- T1= NumericType ("wordlength, 24, fructionlength, 20).
- T2= numerictype (2worldlength, 16, fractionlength,10).
- Real – Real: Observe que la longitud de palabra y la duración de la fracción, su resultado Z son iguales a la suma de las longitudes de palabra y fracción de la longitud respectivamente de la multiplicada, esto se debe a que la Fimat Summode y Producmode se establecen a Fullprecision.

```

>>P = fipref;
>>P.FimathDisplay = 'none';
>>x = fi(5, T1, F)
x =
    5
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 24
    FractionLength: 20
>> y = fi(10, T2, F)
y =
    10
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 16
    FractionLength: 10
>> z = x*y
z =
    50
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 40
    FractionLength: 30
  
```

- Real – Complejo: Observe que la longitud de palabra y la duraxion de Z son iguales a la suma de las longitudes de palabra y fracción de la

longitud de palabra y fracción de la longitud respectivamente. Esto se debe a que las propiedades SumMode y ProductMode se establecen a Fullprecision.

```
>>x = fi(5,T1,F)
x =
    5
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 24
    FractionLength: 20
>>y = fi(10+2i,T2,F)
y =
10.0000 + 2.0000i
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 16
    FractionLength: 10
>>z = x*y
z =
50.0000 +10.0000i
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 40
    FractionLength: 30
```

- Complejo – Complejo: La multiplicación Complejo – Complejo implica un agregado.

```
>>x = fi(5+6i,T1,F)
x =
    5.0000 + 6.0000i
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 24
    FractionLength: 20
>>y = fi(10+2i,T2,F)
y =
10.0000 + 2.0000i
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 16
    FractionLength: 10
>>z = x*y
z =
```

38.0000 +70.0000i

DataTypeMode: Fixed-point: binary point scaling

Signed: true

WordLength: 41

FractionLength: 30

A.13 CASTS

El objeto fimath le permite especificar el tipo de datos y la Scalizacion de sumas intermedias y productos con la propiedad Product Mode.

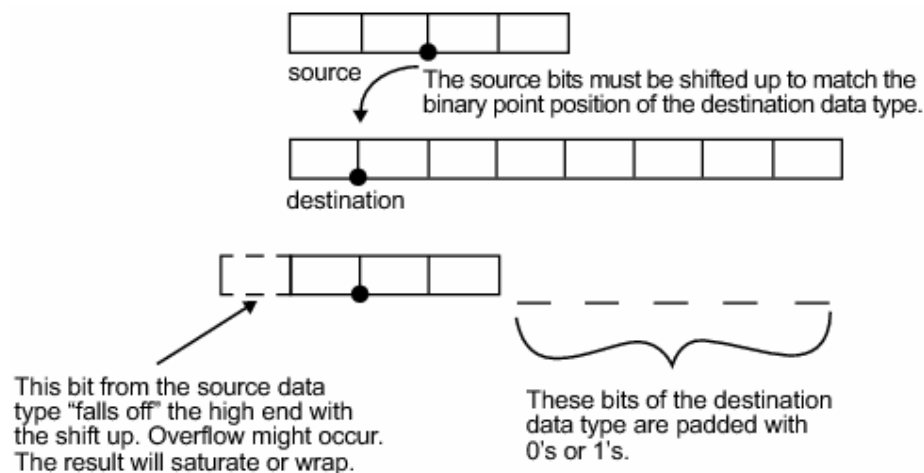
Es importante tener en cuenta las ramificaciones de cada casts al configurar las propiedades SumMode y ProductMode dependiendo de los diferentes tipos de datos que seleccione el overflow y el redondeo pueda ocurrir.

A.13.1 CASTING DE UN TIPO DE DATO MENOR A UN TIPO DE DATO MAS LARGO

Considere el cast de un numero distinto de cero, representado por un tipo de datos fraccionales de 4 bits, a un tipo de datos fraccional de 8 bits con una fracción de 7 bits.

El siguiente diagrama muestra que la fuente bits se desplaza hasta el binario para que se coincida con el punto de posición del destino binario.

La fuente mayor del bit no cabe, por lo que podrían producirse overflow, y el resultado puede saltarse o resaltar.



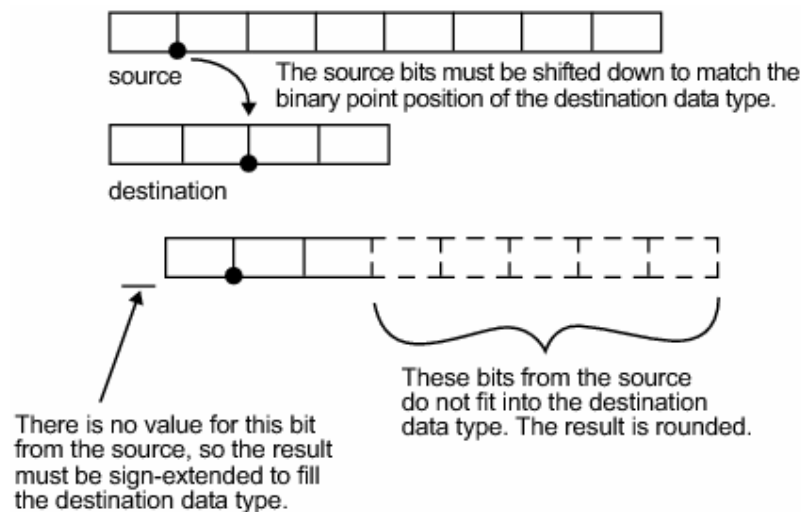
Los bits vacios en el extremo derecho son llenos con 0 y con 1

- Si no se produce overflow, se llena con ceros
- Si se produce la recapitulación, se llena con ceros
- Si se produce la saturación y el numero de bits es positivo se llena con 1

- Si se produce la saturación y el número de bits es negativo se llena con cero.

Del mismo modo, el redondeo de las cifras podría ser necesario incluso cuando se emitan de un dato tipos menor a un tipo de dato mas largo.

Considere el reparto de un número distinto de cero, representado por un tipo de datos fraccional de 8 bits con una fracción de 7 bits, a un tipo de datos de 4 bits con una fracción de 2 bits



El pero de los casos puede producirse cuando el entero de la longitud y la fracción de la longitud de los tipos de datos son más cortos que con los de la fuente de datos y puede ocurrir una pérdida de precisión

A.14 ENTEROS DE TIPOS DE DATOS

A.14.1 TIPO DE DATOS ANSI C ENTEROS

El siguiente cuadro muestra los rangos mínimos del tipo de datos enteros ANSI C. los rangos enteros pueden ser mayor o iguales a la que se muestra, pero no pueden ser menor.

Integer Type	Minimum	Maximum
signed char	−127	127
unsigned char	0	255
short int	−32,767	32,767
unsigned short	0	65,535
int	−32,767	32,767
unsigned int	0	65,535
long int	−2,147,483,647	2,147,483,647
unsigned long	0	4,294,967,295

El rango del largo debe ser mayor o igual al rango de un int, que debe ser igual o superior al rango de un corto.

Tenga en cuenta que el rango mínimo del ANSI C son lo suficientemente grande para dar cabida al componente a uno, o a la representación signo – magnitud, pero no a la representación del complemento a dos.

En el complemento a uno y en el signo – magnitud, un signo entero con n bits tiene un rango de a. en estas dos representaciones por un numero igual de números positivos y negativos están representados y el cero esta representado dos veces.

A.14.2 TIPOS DE DATOS ENTEROS FI

La siguiente tabla muestra los límites numéricos de los tipos de datos enteros del objeto fi, en particular los equivalentes de tipo de datos c.

Constructor	Signed	Word Length	Fraction Length	Minimum	Maximum	Closest ANSI C Equivalent
$fi(x, 1, n, 0)$	Yes	n (2 to 65,535)	0			N/A
$fi(x, 0, n, 0)$	No	n (2 to 65,535)	0	0		N/A
$fi(x, 1, 8, 0)$	Yes	8	0	-128	127	signed char
$fi(x, 0, 8, 0)$	No	8	0	0	255	unsigned char
$fi(x, 1, 16, 0)$	Yes	16	0	-32,768	32,767	short int
$fi(x, 0, 16, 0)$	No	16	0	0	65,535	unsigned short
$fi(x, 1, 32, 0)$	Yes	32	0	-2,147,483,648	2,147,483,647	long int
$fi(x, 0, 32, 0)$	No	32	0	0	4,294,967,295	unsigned long

Los rangos son lo suficientemente grandes como para dar cabida a la representación del complemento a dos que es la única técnica de codificación binario firmado al apoyo del Toolbox de punto fijo.

En la representación del complemento a dos, los bits de signo tienen un rango de -2^{n-1} a $2^{n-1}-1$, un entero sin signo con n bits tiene un rango desde 0 a 2^n-1 .

A.15 CONVERSION UNITARIA

La conversión unitaria, es la conversión que se produce antes de una operación a realizar.

A.15.1 CONVERSION UNITARIA USUAL ANSI C

Solo las conversiones unitarias en ANSI C, se aplican automáticamente en operadores unitarios1, -, n, y, * y de los operadores binarios <<and>> de acuerdo a la siguiente tabla.

Original Operand Type	ANSI C Conversion
char OR short	int

Original Operand Type	ANSI C Conversion
unsigned char OR unsigned short	int OR unsigned int ¹
float	float
Array of T	Pointer to T
Function returning T	Pointer to function returning T

Si el tipo int no es representa todos los valores del tipo de datos original sin overflow.

A.16 CONVERSION BINARIA

A.16.1 CONVERSION BINARIO USUAL ANSI C

En ANSI C, los operandos de un operador binario deben ser del mismo tipo. Si son diferentes uno se convierte en el tipo de la otra de acuerdo a la primera conversión aplicables en la siguiente tabla.

Type of One Operand	Type of Other Operand	ANSI C Conversion
long double	Any	long double
double	Any	double
float	Any	float
unsigned long	Any	unsigned long
long	unsigned	long OR unsigned long ¹
long	int	long
unsigned	int OR unsigned	unsigned
int	int	int

El tipo long es usado si se puede representar todos los valores sin signo .

A16.2 CONVERSION BINARIO USUAL PARA FI

Cuando uno de los operandos de un operador binario (-, *, *) es un objeto FI y el otro es una estructura numerica de matlab, entonces el operador no – fi se convierta en un objeto fi antes de realizar la operación, de acuerdo a la siguiente tabla.

Type of One Operand	Type of Other Operand	Properties of Other Operand After Conversion to a fi Object
fi	double or single	- Signed = same as the original fi operand - WordLength = same as the original fi operand - FractionLength = set to best precision possible
fi	int8	- Signed = 1 - WordLength = 8 - FractionLength = 0
fi	uint8	- Signed = 0 - WordLength = 8 - FractionLength = 0
fi	int16	- Signed = 1 - WordLength = 16 - FractionLength = 0
fi	uint16	- Signed = 0 - WordLength = 16 - FractionLength = 0
fi	int32	- Signed = 1 - WordLength = 32 - FractionLength = 0
fi	uint32	- Signed = 0 - WordLength = 32 - FractionLength = 0

A17. MANEJO DEL OVERFLOW

A17.1 MANEJO DEL OVERFLOW EN ANSI C

En ANSI C, el resultado de las operaciones con signo es cualquier valor y es introducido por la maquina utilizada para la ejecución de dicha accion, por lo tanto ANSI c no tiene reglas para tratar el overflow de enteros con signos.

A17.2 MANEJO DEL OVERFLOW DEL OBJETO FI

La suma y la multiplicación con objetos fi, son como resultados objetos que pueden ser representados en objetos fi, hasta longitudes de palabra de 65.535 bits de la memoria disponible en su maquina.

Esto no es cierto en el caso de la division, sin embargo, debido a que muchos resultados son relacionados a expresiones binarias infinitas.

Se puede realizar la división con objetos fi utilizando la brecha de la función, que requiere especificar explícitamente el tipo numérico de los resultados.

Las condiciones de overflow y los resultados producidos de un objeto fi se determina por la sociedad con el objeto fimath se puede especificar ciertas características del overflow de las sumas, las restas y los productos.

fimath Object Properties Related to Overflow Handling	Property Value	Description
OverflowMode	'saturate'	Overflows are saturated to the maximum or minimum value in the range.
	'wrap'	Overflows wrap using modulo arithmetic if unsigned, two's complement wrap if signed.
ProductMode	'FullPrecision'	Full-precision results are kept. Overflow does not occur. An error is thrown if the resulting word length is greater than MaxProductWordLength. The rules for computing the resulting product word and fraction lengths are given in ProductMode in the Property Reference.
	'KeepLSB'	The least significant bits of the product are kept. Full precision is kept, but overflow is possible. This behavior models the C language integer operations. The resulting word length is determined by the ProductWordLength property. If ProductWordLength is greater than is necessary for the full-precision product, then the result is stored in the least

fimath Object Properties Related to Overflow Handling	Property Value	Description
		significant bits. If ProductWordLength is less than is necessary for the full-precision product, then overflow occurs. The rule for computing the resulting product fraction length is given in ProductMode in the Property Reference.
	'KeepMSB'	The most significant bits of the product are kept. Overflow is prevented, but precision may be lost. The resulting word length is determined by the ProductWordLength property. If ProductWordLength is greater than is necessary for the full-precision product, then the result is stored in the most significant bits. If ProductWordLength is less than is necessary for the full-precision product, then rounding occurs. The rule for computing the resulting product fraction length is given in ProductMode in the Property Reference.
	'SpecifyPrecision'	You can specify both the word length and the fraction length of the resulting product.
ProductWordLength	Positive integer	The word length of product results when ProductMode is 'KeepLSB', 'KeepMSB', or 'SpecifyPrecision'.
MaxProductWordLength	Positive integer	The maximum product word length allowed when ProductMode is 'FullPrecision'. The default is 128 bits. The maximum is 65,535 bits. This property can help ensure that your simulation does not exceed your hardware requirements.
ProductFractionLength	Integer	The fraction length of product results when ProductMode is 'Specify Precision'.
SumMode	'FullPrecision'	Full-precision results are kept. Overflow does not occur. An error is thrown if the resulting word length is greater than MaxSumWordLength. The rules for computing the resulting

fimath Object Properties Related to Overflow Handling	Property Value	Description
		sum word and fraction lengths are given in SumMode in the Property Reference.
	'KeepLSB'	The least significant bits of the sum are kept. Full precision is kept, but overflow is possible. This behavior models the C language integer operations. The resulting word length is determined by the SumWordLength property. If SumWordLength is greater than is necessary for the full-precision sum, then the result is stored in the least significant bits. If SumWordLength is less than is necessary for the full-precision sum, then overflow occurs. The rule for computing the resulting sum fraction length is given in SumMode in the Property Reference.
	'KeepMSB'	The most significant bits of the sum are kept. Overflow is prevented, but precision may be lost. The resulting word length is determined by the SumWordLength property. If SumWordLength is greater than is necessary for the full-precision sum, then the result is stored in the most significant bits. If SumWordLength is less than is necessary for the full-precision sum, then rounding occurs. The rule for computing the resulting sum fraction length is given in SumMode in the Property Reference.
	'SpecifyPrecision'	You can specify both the word length and the fraction length of the resulting sum.
SumWordLength	Positive integer	The word length of sum results when SumMode is 'KeepLSB', 'KeepMSB', or 'SpecifyPrecision'.
MaxSumWordLength	Positive integer	The maximum sum word length allowed when SumMode is 'FullPrecision'. The

fimath Object Properties Related to Overflow Handling	Property Value	Description
		default is 128 bits. The maximum is 65,535 bits. This property can help ensure that your simulation does not exceed your hardware requirements.
SumFractionLength	Integer	The fraction length of sum results when SumMode is 'SpecifyPrecision'

A.18 CONSTRUCCION DEL OBJETO FI

A.18.1 SINTEXIS DEL OBJETO FI

Se puede crear un objeto fi en el Toolbox de punto fijo de dos formas:

- Se puede utilizar la función constructor de un objeto Fi para crear un nuevo objeto.
- Se puede utilizar la función constructor Fi, para copiar un objeto Fi existente

Para empezar, el tipo:

```
>>a = fi(0)
```

Para crear un objeto fi, se crea con el tipo de datos por defecto y un valor de 0.

```
>>a =
    0
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 16
    FractionLength: 15
```

El objeto Fi, es creado con un valor de 0, una longitud de palabra de 16 y una fracción de longitud de 15.

Se puede usar la función de constructor Fi de la siguiente manera:

- `a= Fi:` es el constructor por defecto y devuelve un objeto Fi con ningun valor, la longitud de palabra de 16 bits y la fracción de longitud de 15 bits.

- $a=Fi(v)$: retorna la señal del objeto punto fijo con valor V , longitud de palabra de 16 bits y mejor precisión de la fracción de longitud.
- $a=Fi(v,s)$: retorna un objeto de punto fijo con valor V , signedness S , longitud de palabra 16 mejor precisión de la fracción de longitud. Se puede ser 0 (falso) o 1 (verdadero).
- $a=Fi(v,s,w)$: retorna un objeto de punto fijo con valor v , signodness S , longitud de palabra W , mejor precisión de la fracción de longitud.
- $a=fi(v,s,w,f)$: retorna un objeto punto fijo con valor V , Signedess S , logitud de palabra W y longitud de fracción F .
- $a=FI(V,S,W, Slope,bias)$: retorna un objeto punto fijo con valo V , signedness S , longitud de palabra W , pendiente y escalonada.
- $a=FI(V,S,W,Slopeadjustmentfactor, fixedexponet, bias)$: retoma un objeto punto fijo con valor v , signedness S , longitud de palabra W , ajustamiento del factor de pendiente, punto exponente, escalonada.
- $a=FI(V,T)$: retorna un objeto punto fijo con un valor V , encaja el tipo numérico a T .
- $a=FI(V,F)$: retorna un objeto punto fijo con valor V , encaja F fimath, longitud de palabra de 16 bits y mejor presicion de la fracción de longitud.
- $b=FI(a,f)$: permite mantener el valor de a y el tipo numérico del objeto Fi , y cambia el objeto a F fimath.
- $a=FI(V,T,F)$: retorna un valor V de un objeto punto fijo, encaja un tipo numérico T , encaja fimath F .
- $a=FI(V,S,F)$: retorna un objeto punto fijo con valor V , signedness S , longitud de palabra de 16 bits, mejor precision de la fracción de longitud y encaja imath F .
- $a=FI(V, S,, W, F)$: retorna un objeto punto fijo con valor V , signedness S , longitud de palabra W , mejor presicion de la fracción de longitud, encaja F fimath.
- $a=FI(V, S,W,F,f)$: retorna un objeto punto fijo con valor V , signedness S , longitud de palabra W , fracción de longitud f , encaja fitmath F .
- $a=FI(V,S,W, slope, bias, F)$: retorna un objeto punto fijo con valor V , signedness S , pendiente, escalonada, encaja F fimath.
- $a=FI(V,S,W,slopeadjuntment factor, fixedexponet, bias, F)$: retorna un objeto punto fijo con valor V , Signedness S , longitud de palabra W , ajusta el factor de pendiente, punto exponente, escalonada, encaja F fimath.
- $a=FI(... "PROPERTYNAME", PROPERTY VALUE...)$.
- $a=FI(... "PROPERTYNAME", PROPERTY VALUE...)$: permiten configurar 8 tsblo objetos punto fijo para un objeto FI con propiedades de nombre y valores de propiedad.

A.18.2 EJEMPLOS DE LA CONSTRUCCION DE OBJETOS FI

El siguiente ejemplo crea un objeto con un valor π y una longitud de palabra de 8 bits y una longitud de fracción de 3 bits.

```
>>a = fi(pi, 1, 8, 3)
a =
    3.1250
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 8
    FractionLength: 3
```

El valor de v puede tener un array de:

```
>>a = fi((magic(3)/10), 1, 16, 12)
a =
    0.8000    0.1001    0.6001
    0.3000    0.5000    0.7000
    0.3999    0.8999    0.2000
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 16
    FractionLength: 12
```

Si se omite el argumento f, automáticamente se hace la precisión más posible:

```
>>a = fi(pi, 1, 8)
a =
    3.1563
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 8
    FractionLength: 5
```

Si se omite w y f, se ajusta automáticamente a 16 bits y a la mejor precisión posible:

```
>>a = fi(pi, 1)
a =
    3.1416
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 16
    FractionLength: 13
```

A.18.3 CONSTRUCCION DE UN OBJETO FI CON PROPIEDAD DE NOMBRES Y PROPIEDAD DE VALORES

Puede utiliza la propiedad nombre/valor para establecer los pares al crear un objeto fi:

```
>>a = fi(pi, 'roundmode', 'floor', 'overflowmode', 'wrap')
a =
    3.1415
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 16
    FractionLength: 13
    RoundMode: floor
    OverflowMode: wrap
    ProductMode: FullPrecision
    MaxProductWordLength: 128
    SumMode: FullPrecision
    MaxSumWordLength: 128
    CastBeforeSum: true
```

A.18.4 CONSTRUCCION DE UN OBJETO FI, USANDO EL OBJETO NUMERICTYPE

Se puede usar el objeto `numerictype` para definir un objeto `fi`:

```
>>T = numerictype
T =
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 16
    FractionLength: 15
>>a = fi(pi, T)
a =
    1.0000
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 16
    FractionLength: 15
    RoundMode: nearest
    OverflowMode: saturate
    ProductMode: FullPrecision
    MaxProductWordLength: 128
    SumMode: FullPrecision
    MaxSumWordLength: 128
    CastBeforeSum: true
```

Se puede usar un objeto `fimath` con un objeto de tipo `numerictype` para definir un objeto `fi`:

```
>>F = fimath
```

```

F =
    RoundMode: nearest
    OverflowMode: saturate
    ProductMode: FullPrecision
    MaxProductWordLength: 128
    SumMode: FullPrecision
    MaxSumWordLength: 128
    CastBeforeSum: true
>>a = fi(pi, T, F)
a =
    1.0000
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 16
    FractionLength: 15
    RoundMode: nearest
    OverflowMode: saturate
    ProductMode: FullPrecision
    MaxProductWordLength: 128
    SumMode: FullPrecision
    MaxSumWordLength: 128
    CastBeforeSum: true

```

A.18.5 CONSTRUCCION DE UN OBJETO FI USANDO UN OBJETO FIMATH

```

>>F = fimath
F =
    RoundMode: nearest
    OverflowMode: saturate
    ProductMode: FullPrecision
    MaxProductWordLength: 128
    SumMode: FullPrecision
    MaxSumWordLength: 128
    CastBeforeSum: true
>>F.OverflowMode = 'wrap'
F =
    RoundMode: nearest
    OverflowMode: wrap
    ProductMode: FullPrecision
    MaxProductWordLength: 128
    SumMode: FullPrecision
    MaxSumWordLength: 128
    CastBeforeSum: true
>>a = fi(pi, F)
a =

```

```

3.1416
  DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 16
  FractionLength: 13
    RoundMode: nearest
  OverflowMode: wrap
  ProductMode: FullPrecision
MaxProductWordLength: 128
  SumMode: FullPrecision
MaxSumWordLength: 128
  CastBeforeSum: true

```

Se puede crear un objeto fi usando un objeto fimath siempre que se especifiquen varias propiedades de numuerictype en el momento de crear dicho objeto:

```

>>b = fi(pi, 0, F)
b =
  3.1416
    DataTypeMode: Fixed-point: binary point scaling
      Signed: false
      WordLength: 16
    FractionLength: 14
      RoundMode: nearest
    OverflowMode: wrap
    ProductMode: FullPrecision
MaxProductWordLength: 128
  SumMode: FullPrecision
MaxSumWordLength: 128
  CastBeforeSum: true
>>c = fi(pi, 0, 8, F)
c =
  3.1406
    DataTypeMode: Fixed-point: binary point scaling
      Signed: false
      WordLength: 8
    FractionLength: 6
      RoundMode: nearest
    OverflowMode: wrap
    ProductMode: FullPrecision
MaxProductWordLength: 128
  SumMode: FullPrecision
MaxSumWordLength: 128
  CastBeforeSum: true
>>d = fi(pi, 0, 8, 6, F)
d =

```

3.1406

```
    DataTypeMode: Fixed-point: binary point scaling
      Signed: false
      WordLength: 8
    FractionLength: 6
      RoundMode: nearest
    OverflowMode: wrap
    ProductMode: FullPrecision
MaxProductWordLength: 128
    SumMode: FullPrecision
MaxSumWordLength: 128
    CastBeforeSum: true
```

A.18.6 DETERMINACION DE LA PROPIEDAD PRECEDENTE

Tenga en cuenta que el valor de una propiedad se ha tomado desde la última vez que se ha establecido. Por ejemplo, crear un objeto `numerictype` con un valor de propiedad `'signed'`:

```
>>T = numerictype('signed', true)
T =
    DataTypeMode: Fixed-point: binary point scaling
      Signed: true
      WordLength: 16
    FractionLength: 15
```

Crear un objeto `fi` donde la propiedad `numerictype` es especificado después de la propiedad `signed`, de modo que el resultado sea un objeto `fi signed`.

```
>>a = fi(pi,'signed',false,'numerictype',T)
a =
    1.0000
    DataTypeMode: Fixed-point: binary point scaling
      Signed: true
      WordLength: 16
    FractionLength: 15
      RoundMode: nearest
    OverflowMode: saturate
    ProductMode: FullPrecision
MaxProductWordLength: 128
    SumMode: FullPrecision
MaxSumWordLength: 128
    CastBeforeSum: true
```

A.18.7 COPIA DE UN OBJETO FI

La copia de un objeto fi es muy sencilla, como se observa en el siguiente ejemplo:

```
>>a = fi(pi)
a =
    3.1416
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 16
    FractionLength: 13
>>b = a
b =
    3.1416
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 16
    FractionLength: 13
```

A.19 PROPIEDADES DEL OBJETO FI

A.19.1 PROPIEDAD DE DATOS

Las propiedades de los datos de un objeto siempre son de escritura fi:

- Bin: Almacena valores de un objeto fi en binario.
- Data: Valor numérico real la palabra de un objeto fi.
- Dec: Almacena valores de un objeto fi en decimal.
- Double: Valor de la palabra real de un objeto fi, almacenado en matlab como doble.
- Hex: Almacena valores de un objeto fi en hexagecimal.
- Int: Almacena valores enteros de un objeto fi, almacena tipo de datos enteros incorporados en matlab. También se puede almacenar un objeto fi en formatos de Int, Int 10, Int 32, Unit 8, Unit 16, Unit 32.
- Oct: Almacena valores de un objeto fi en octal

A.19.2 PROPIEDAD DE FIMATH

Cuando se crea un objeto fi, se crea automáticamente un objeto fimath como una propiedad del objeto fi:

- Fimath: Objeto fimath asociado con un objeto fi

Las siguientes propiedades fimath son por transitividad de las propiedades de un objeto fi:

- Cast Before Sum: Cuando ambos operandos son cast al tipo de datos antes de la adición.
- Max Product Word Length: Valor máximo de la longitud de palabra para el producto de tipos de datos.
- Max Sum Word Length: Valor máximo de la longitud de palabra para la suma de tipos de datos.
- Overflow mode: modo overflow.
- Product Bias: Escalonamiento de los productos de tipos de datos.
- Product Fixed Point: Punto exponente del producto de tipo de datos.
- Product Fraction Length: Longitud de fracción en bits, de el producto de tipo de datos.
- Product Mode: Define como el producto de tipos de datos es determinado.
- Product Slope: Pendiente del producto de tipo de datos.
- Product Slope Adjustment Factor: Factor de ajuste del producto de la pendiente de tipos de datos.
- Product Word Length: Longitud de palabra en bits del producto de tipo de datos.
- Round Mode: Modo redondeo.
- Sum Bias: Escalonamiento de la suma de tipo de datos.
- Sum Fixed Exponet: Punto exponente de la suma de datos tipo.
- Sum Fraction Length: Fracción de longitud en bits, de la suma de datos tipo.
- Sum Mode: Define como se determina la suma de tipos de datos.
- Sum Slope: Pendiente de la suma de tipo de datos.
- Sum Slope Adjustment Factor: Factor de ajuste de la pendiente de la suma de los tipos de datos.
- Sum Word Length: Longitud de la palabra en bits de la pendiente de la suma de los tipos de datos.

A.19.3 PROPIEDAD NUMERIC TYPE

Al crear un objeto fi, un objeto numerictype es creado automáticamente con las características de las propiedades del objeto fi.

- NumericType: Objeto que contiene todos los atributos de tipo numérico de un objeto fi.
- Bias: Escalonamiento de un objeto fi.
- Data Type: Categoría de tipos de datos asociado con un objeto fi.
- Data Type Mode: Tipo de dato y modo de escalizacion de un objeto fi.
- Fixed Exponet: Exponente punto fijo asociado de un objeto fi.
- Fraction Length: fracción de longitud de el valor entero almacenado de un objeto fi en bits.
- Scaling: Modo de escalizacion de punto fijo de un objeto fi.
- Signed: Hecho de que un objeto fi tenga o no signo.

- Slope: Pendiente asociado con un objeto fi.
- Slope Adjustment Factor: Ajuste de pendiente de un objeto fi.
- Word Length: Valor de la longitud de pendiente de un objeto fi almacenado en bits.

A.19.4 CONFIGURACION DE PROPIEDADES DE PUNTO FIJO EN LA CREACION DE UN OBJETO

Se puede configurar las propiedades de los objetos fi en el momento de la creación de objetos mediante la inclusión de propiedades después de los argumentos de la construcción de un objeto fi.

```
>>a = fi(pi, 'OverflowMode', 'wrap', 'RoundMode', 'convergent')
a =
    3.1416
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 16
    FractionLength: 13
    RoundMode: convergent
    OverflowMode: wrap
    ProductMode: FullPrecision
    MaxProductWordLength: 128
    SumMode: FullPrecision
    MaxSumWordLength: 128
    CastBeforeSum: true
```

A.19.5 USO DIRECTO DE LA PROPIEDAD DE REFERENCIA CON FI

Puede una propiedad reverenciarse directamente para establecer o recuperar el valor de la propiedad de un objeto fi, usando la referencia de estructura de matlab.

```
>>a.DataTypeMode
ans =
Fixed-point: binary point scaling
```

Para obtener el overflowmode de a:

```
>>a.OverflowMode = 'wrap'
a =
    3.1416
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 16
```

FractionLength: 13
 RoundMode: convergent
 OverflowMode: wrap
 ProductMode: FullPrecision
 MaxProductWordLength: 128
 SumMode: FullPrecision
 MaxSumWordLength: 128
 CastBeforeSum: true

A.20 FUNCIONES DE UN OBJETO FI

Las siguientes funciones son operadores directos de un objeto fi.

Abs	Counter 3	Fimath	Ipermute
All	Counter C	Flipdim	Isodumn
And	Counter F	Flipzg	Isempy
Any	Ctronpose	Flipud	Isequal
Area	Dey	Fplot	Isfi
Bar	Diag	Ge	Isfinite
Boch	Disp	Get	Isinf
Binary	Double	Gplast	Isnan
Bitand	End	Gt	Isnorent
Bitemp	Eps	Hanler	Isubyect
Bitget	Eq	Hex	Ispropequiel
Bitar	Errorloar	Hexznum	Isred
Bitshift	Etreeplst	Hist	Isrow
Bitxor	Ezccantar	Histe	Isscalar
Buffe	Ezcontourf	Horzcst	Iassigned
Clabel	Ezmesh	Imaq	Isvector
Canet	Ezplut	Innerprodntbits	Le
Carnet 3	Ezplut 3	Int	Length
Compás	Ezpolar	Int 8	Line
Complex	Ezsuf	Int 16	Logical
Coneplut	Ezsurfc	Int 32	Lowerboard
Conj	Feather	Intmax	Lsb
Contour	Fi	Intmin	Lt
Max	Plot 3	Rose	Subsiref
Mesh	Plotmatrix	Scotter	Sum
Meshc	Plotxy	Scatter 3	Surf
Mezhz	Plus	Ssec	Surf
Min	Polar	Shiftdim	Sufl
Minus	Pow2	Sign	Surfnom
Mtimes	Quantizer	Single	Text
Ndimes	Quiver	Size	Tines

Ne	Quiver 3	Slice	Topplitz
Not	Range	Spy	Transpose
Number of element	Real	Staire	Treeplut
Numerictype	Relamas	Stem	Tril
Oct	Realmin	Stem3	Tristem
Or	Repmat	Streamribbon	Triplot
Match	Rescale	Streamslice	Tndurf
Pcolor	Reshafa	Streamtube	Triu
Permute	Rgbplut	Stripsaline	Unit 8
Plot	Ribbon	Subsasgn	Unit 16
Unit 32	Uminus	Uplus	Upperbound
Vertcat	Vacune	Voronoin	Waterfull
Xlim	Ylim	Zlim	

Las siguientes funciones de datos de acceso, tambien pueden ser usadas para obtener los datos de un objeto fi usando la notacion punto

- Bin
- Data
- Dec
- Double
- Hex
- Int
- Oct

A.21 CONSTRUCCION DE OBJETOS FIMATH

Los objetos fimath definen la aritmetica de atributos de los objetos fi, se pueden crear objetos fimath en el toolbox de punto fijo de 2 formas

- Usando la funcion fimath para crear una nueva funcion.
- Utilizando la funcion fimath para copiar un objeto fimath ya existente.

Por ejemplo:

```
>>a = fi(pi);
>>n = int(a)
n = 25736
>>a.int
ans = 25736
>>h = hex(a)
h = 6488
>>a.hex
ans =6488
```

A.22 PORPIEDADES DE UN OBJETO FIMATH

A.22.1 PROPIEDADES MATH, ROUNDING Y OVERFLOW

Las siguientes propiedades de los objetos fimath siempre son escribibles.

- Cast Before Sum: Si ambos operandos son echadas al tipo de datos antes de la suma.
- MaxProductWordLength: Maxima longitud de palabra admisible para el producto de datos.
- MaxSumWordLength: Maxima longitud de palabras admisible para la suma de datos.
- Overflow Mode: Modo de desbordamiento.
- ProductBias: Producto escalonado de datos.
- Product Fixed Exponent: Punto exponente del tipo de datos.
- Product Fraction Length: fracción de palabras en bits del producto de datos.
- Productmode: Define la forma como el producto de datos es determinado.
- Productslope: Pendiente del producto de datos.
- Productslope Adjustment Factor: Factor de ajuste de la pendiente del producto de datos.
- Product Word Length: Longitud de palabra a bits del producto de datos.
- Round Mode: Modo de redondeo.
- Sumbias: Escalonamiento de la suma de datos.
- SumFixedExponet: Punto exponente de la suma de datos.
- SumFractionLength: fracción de longitud en bit de la suma de datos.
- Sum Mode: Define la forma de suma de datos es predeterminada.
- Sumslope: Pendiente de la suma de datos.
- SumSlopeAdjustmentfactor: Factor de ajuste de la pendiente en la suma de datos.
- Sumwordlength: Longitud de palabras a bit de la suma de datos.

A.22.2 CONFIGURACION DE LAS PROPIEDADES FIMATH EN LA CREACION DE OBJETOS.

Se puede configurar las propiedades de objetos fimath en el momento de creación de objetos mediante la inclusión de propiedades después de los argumentos de la funcion creada fimath, por ejemplo para establecer desaturacion y el modo de redondeo convergente

```
>> F= fimath ("overflow", "saturate", "roundmode", 2convergent")
```

```
F=
```

```
Roundmode: Convergent
```

```
Overflowmode: Saturate
Product mode: Fullprecision
Maxproductwordlength=128
Summode=fullprecision
Maxsumwordlength=128
CastBeforeSum= true
```

A.22.3 USO DE LA PROPIEDAD DE REFERENCIA DIRECTA DEL OBJETO FIMATH

Se puede referenciar directamente en una propiedad para establecer o recuperar el valor de una propiedad objeto FIMATH utilizando la referencia de estructura de MATLAB. Para ello se utiliza un periodo para el índice introduciendo una propiedad por su nombre.

Por ejemplo, para obtener el redondeo de F:

```
>>F.RoundMode
    ans = convergent
```

Para establecer el tipo de redondeo de F:

```
>>F.OverflowMode = 'wrap'
    F =
        RoundMode: convergent
        OverflowMode: wrap
        ProductMode: FullPrecision
        MaxProductWordLength: 128
        SumMode: FullPrecision
        MaxSumWordLength: 128
        CastBeforeSum: true
```

A.22.4 CONFIGURACION DE LAS PROPIEDADES EN EL MODELO EXPLORADOR

Se puede ver y cambiar las propiedades de cualquier objeto FIMATH definidos en el workspace de Matlab. Abra el modelo explorador seleccionando VIEW> MODEL EXPLORER, en cualquier modelo de Simulink.

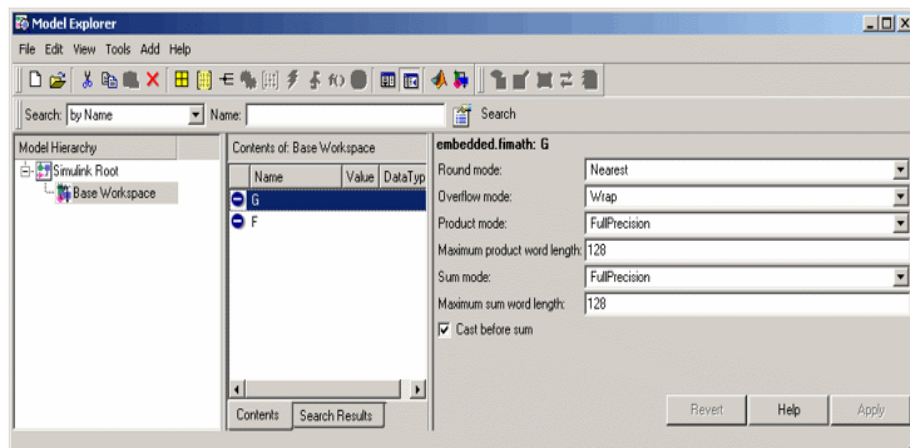
La siguiente figura muestra el modelo explorador cuando se definen los siguientes objetos FIMATH en el workspace de Matlab:

```
>>F = fimath
    F =
        RoundMode: nearest
        OverflowMode: saturate
        ProductMode: FullPrecision
```

```

MaxProductWordLength: 128
    SumMode: FullPrecision
    MaxSumWordLength: 128
    CastBeforeSum: true
>>G = fimath('OverflowMode','wrap')
    G =
        RoundMode: nearest
        OverflowMode: wrap
        ProductMode: FullPrecision
MaxProductWordLength: 128
    SumMode: FullPrecision
MaxSumWordLength: 128
    CastBeforeSum: true

```



Seleccione el nodo base del workspace en el panel del modelo jerargico, para mirar el objeto actual en el panel de contenidos. Cuando se selecciona un objeto FIMATH en el panel de contenidos, se puede ver y cambiar las propiedades ene el panel de dialogo.

A.23 USO DE OBJETOS FIMATH EN LAS OPERACIONES ARITMETICA DE PUNTO FIJO.

A.23.1ARITMETICA DE PUNTO BINARIO

El objeto FIMATH encapsula propiedades matematicas del toolbox de punto fijo, y a su vez es una propiedad del objeto FI. Cada objeto FI tiene un objeto FIMATH como propiedad.

```

>>a = fi(pi)
    a =  3.1416
    DataTypeMode: Fixed-point: binary point scaling

```

```

        Signed: true
        WordLength: 16
        FractionLength: 13
        RoundMode: nearest
        OverflowMode: saturate
        ProductMode: FullPrecision
        MaxProductWordLength: 128
        SumMode: FullPrecision
        MaxSumWordLength: 128
        CastBeforeSum: true
>>a.fimath
ans =
        RoundMode: nearest
        OverflowMode: saturate
        ProductMode: FullPrecision
        MaxProductWordLength: 128
        SumMode: FullPrecision
        MaxSumWordLength: 128
        CastBeforeSum: true

```

Para realizar operaciones aritmeticas cono +, -, .*, o *, los dos operandos fi deben tener las misma propiedades FIMATH.

```

>>a = fi(pi);
>>b = fi(8);
>>isequal(a.fimath, b.fimath)
ans =
    1
>>a + b
ans =
    11.1416
        DataTypeMode: Fixed-point: binary point scaling
        Signed: true
        WordLength: 19
        FractionLength: 13
        RoundMode: nearest
        OverflowMode: saturate
        ProductMode: FullPrecision
        MaxProductWordLength: 128
        SumMode: FullPrecision
        MaxSumWordLength: 128
        CastBeforeSum: true

```

Para realizar operaciones aritmeticas con +, -, .*, o *, dos operandos FI tambien deben tener el mismo tipo de dato. Por ejemplo, se puede realizar operaciones

con dos objetos FI con tipos de datos doble, pero no con un onjeto de datos dobles y un dato unico:

```
>>a = fi(3, 'DataType', 'double')
a =
    3
    DataTypeMode: double
>>b = fi(27, 'DataType', 'double')
b =
    27
    DataTypeMode: double
>>a + b
ans =
    30
    DataTypeMode: double
>>c = fi(12, 'DataType', 'single')
c =
    12
    DataTypeMode: single
>>a + c
>>> Math operations are not allowed on FI objects with
        different data types.
```

Los operandos del objeto FI punto fijo no tienen el mismo escalamientos. Las matematicas estan permitidas entre los objetos FI punto fijo y los objetos FI de doble escalamiento. En este sentido el tipo de datos del doble escalamiento actua como tipo de datos punto fijos:

```
>>a = fi(pi)
a =
    3.1416
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 16
    FractionLength: 13
    RoundMode: nearest
    OverflowMode: saturate
    ProductMode: FullPrecision
    MaxProductWordLength: 128
    SumMode: FullPrecision
    MaxSumWordLength: 128
    CastBeforeSum: true
>>b = fi(magic(2),'DataTypeMode', 'Scaled double: binary point scaling')
b =
     1     3
     4     2
```

```

        DataTypeMode: Scaled double: binary point scaling
        Signed: true
        WordLength: 16
        FractionLength: 12
        RoundMode: nearest
        OverflowMode: saturate
        ProductMode: FullPrecision
        MaxProductWordLength: 128
        SumMode: FullPrecision
        MaxSumWordLength: 128
        CastBeforeSum: true
>>a + b
ans =
    4.1416    6.1416
    7.1416    5.1416
        DataTypeMode: Scaled double: binary point scaling
        Signed: true
        WordLength: 18
        FractionLength: 13
        RoundMode: nearest
        OverflowMode: saturate
        ProductMode: FullPrecision
        MaxProductWordLength: 128
        SumMode: FullPrecision
        MaxSumWordLength: 128
        CastBeforeSum: true

```

El uso de la funcion de division para llevar a cabo la division de datos dobles, unicos, objetos FI binario de punto unico de escalamiento.

A.23.2 [PENDIENTE INCLINACION] ARITMETICA

La aritmetica de punto fijo utilizando un objeto FIMATH cuenta con el apoyo de todos los puntos binarios de la señal.la aritmetica tambien cuenta con el apoyo de la pendiente segada de las señales con las siguientes restricciones:

- [pendiente inclinacion] las señales deben ser reales.
- Las propiedades SUMMODE y PRODUCTMODE del objeto FIMATH debe ser configurado de la forma “specifyprecision” para las operaciones de la suma y de la multiplicación respectivamente.
- La propiedad CASTBEFORESUM del objeto FIMATH debe ser establecido en TRUE.
- La funcion de division no es compatible para las señales con pendiente inclinacion.

```

>>f = fimath('SumMode', 'SpecifyPrecision','SumFractionLength', 16)
f =
    RoundMode: nearest
    OverflowMode: saturate
    ProductMode: FullPrecision
    MaxProductWordLength: 128
    SumMode: SpecifyPrecision
    SumWordLength: 32
    SumFractionLength: 16
    CastBeforeSum: true
>>a = fi(pi, 'fimath', f)
a =
    3.1416
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 16
    FractionLength: 13
    RoundMode: nearest
    OverflowMode: saturate
    ProductMode: FullPrecision
    MaxProductWordLength: 128
    SumMode: SpecifyPrecision
    SumWordLength: 32
    SumFractionLength: 16
    CastBeforeSum: true
>>b = fi(22, true, 16, 2^-8, 3, 'fimath', f)
b =
    22
    DataTypeMode: Fixed-point: slope and bias scaling
    Signed: true
    WordLength: 16
    Slope: 0.00390625
    Bias: 3
    RoundMode: nearest
    OverflowMode: saturate
    ProductMode: FullPrecision
    MaxProductWordLength: 128
    SumMode: SpecifyPrecision
    SumWordLength: 32
    SumFractionLength: 16
    CastBeforeSum: true
>>a + b
ans =
    25.1416
    DataTypeMode: Fixed-point: binary point scaling

```

```

Signed: true
WordLength: 32
FractionLength: 16
RoundMode: nearest
OverflowMode: saturate
ProductMode: FullPrecision
MaxProductWordLength: 128
SumMode: SpecifyPrecision
SumWordLength: 32
SumFractionLength: 16
CastBeforeSum: true

```

Al configurar las propiedades SUMMODE y PRODUCTMODE de la forma SEPECIFYPRECISION se excluyen mutuamente, salvo cuando se realiza la operacion * entre matrices. En ese caso, tanto las propiedades SUMMODE Y PRODUCTMODE deben ser establecidas de la forma SPECIFYPRECISION para señales con pendiente inclinacion, ya que ambas operaciones (suma y multiplicacion) calculan el resultado.

A.24 USO DE LAS ESPECIFICACIONES DE REDONDEO Y EL MODO DE OVERFLOW EN UN OBJETO

Solo los modos de redondeo y de overflow establecidos antes de una operacion de un objeto FI afectan el resultados de esas operaciones. Al cambiar el moro de redondeo y del overflow de un objeto FI después de que se ha creado no afecta su valor. Por ejemplo cosiderando los objetos a y b:

```

>> p = fipref('NumberDisplay', 'RealWorldValue','NumericTypeDisplay',
    'none', 'FimathDisplay', 'none');
>>T = numerictype('WordLength',8,'FractionLength',7);
>>F = fimath('RoundMode','floor','OverflowMode','wrap');
>>a = fi(1,T,F)
a =
    -1
>>b = fi(1,T)
b =
    0.9922

```

Porque a es creado con un objeto F FIMATH que ha establecido un OVERFLOWMODE como WRAP de -1. Por otro lado b es creado con un valro de OVERFLOWMODE como SATURATE de 0.9922.

Ahora se asigna el objeto FIMATH b:

```

>>b.fimath = F
b =

```

0.9922

Debido a que la operaciones asignadas y correspondientes del overflow y saturación ocurrio cuando se creo b, no ocurren cambios en b cuando se asignan los nuevos valores del objeto FIMATH, a pesar de que el modo de redondeo se ha cambiado de SATURATE a WRAP.

A.25 USAR FIMATH PARA COMPATIR REGLAS ARITMETICAS

Se puede utilizar un objeto FIMATH para definir normas comunes de la aritmetica que se usa en muchos objeto FI. A continuacion, se pueden crear varios objetos FI utilizando el mismo objeto FIMATH, para ello se necesita crear un tipo de datos comunes para la aplicación.

El siguiente ejemplo muestra la creación de un objeto NUMERICTYPE y un objeto FIMATH, que luego se utiliza para crear dos objetos FI con los mismos NUMERICTYPE y los mismo atributos FIMATH:

```
>>T = numerictype('WordLength', 32, 'FractionLength', 30)
```

```
T =
```

```
    DataTypeMode: Fixed-point: binary point scaling
```

```
        Signed: true
```

```
    WordLength: 32
```

```
    FractionLength: 30
```

```
>>F = fimath('RoundMode', 'floor', 'OverflowMode', 'wrap')
```

```
F =
```

```
    RoundMode: floor
```

```
    OverflowMode: wrap
```

```
    ProductMode: FullPrecision
```

```
    MaxProductWordLength: 128
```

```
        SumMode: FullPrecision
```

```
    MaxSumWordLength: 128
```

```
    CastBeforeSum: true
```

```
>>a = fi(pi, T, F)
```

```
a =
```

```
-0.8584
```

```
    DataTypeMode: Fixed-point: binary point scaling
```

```
        Signed: true
```

```
    WordLength: 32
```

```
    FractionLength: 30
```

```
    RoundMode: floor
```

```
    OverflowMode: wrap
```

```
    ProductMode: FullPrecision
```

```
    MaxProductWordLength: 128
```

```
        SumMode: FullPrecision
```

```
    MaxSumWordLength: 128
```

```

        CastBeforeSum: true
>>b = fi(pi/2, T, F)
b =
    1.5708
        DataTypeMode: Fixed-point: binary point scaling
           Signed: true
        WordLength: 32
    FractionLength: 30
           RoundMode: floor
        OverflowMode: wrap
        ProductMode: FullPrecision
MaxProductWordLength: 128
           SumMode: FullPrecision
MaxSumWordLength: 128
        CastBeforeSum: true

```

A.26 USO DEL PRODUCTMODE Y SUMMODE DEL OBJETO FIMATH

A.26.1 EJEMPLO DE CONFIGURACION:

Los siguientes ejemplos muestran las diferencias entre los cuatro ajustes de las propiedades PRODUCTMODE y del SUMMODE, como lo son:

- Fullprecision
- keepLSB
- keepMSB
- specifyprecision

El siguiente ejemplo muestra como se establecen las propiedades:

```

>>p = fipref;
>>p.NumericTypeDisplay = 'short';
>>p.FimathDisplay = 'none';
>>p.LoggingMode = 'on';
>>F = fimath('OverflowMode','wrap','RoundMode','floor',...
            'CastBeforeSum',false);
warning off
format compact

```

A continuacion se definen los objetos FI a y b. Ambos tienen bit de signo y un tipo de datos de 8 bits. La longitud de fraccion es elegida automáticamente para cada objeto FI, para el mejor rendimiento posible de la precision:

```

>>a = fi(pi, true, 8)
a =

```

```

3.1563
s8,5
>>b = fi(exp(1), true, 8)
b =
2.7188
s8,5

```

A.26.2 FULLPRECISION

Ahora para las propiedades de PRODUCEMODE y SUMMODE de a y b, se especifican como FULLPRECISION y se ven los siguientes resultados:

```

>>F.ProductMode = 'FullPrecision';
>>F.SumMode = 'FullPrecision';
>>a.fimath = F;
>>b.fimath = F;
>>a
a =
3.1563                                %011.00101
s8,5
>>b
b =
2.7188                                %010.10111
s8,5
>>a*b
ans =
8.5811                                %001000.1001010011
s16,10
>>a+b
ans =
5.8750                                %0101.11100
s9,5

```

En el modo de FULLPRECISION, la longitud de palabra del producto crece con respecto a la longitud de palabra de los operandos de la suma. En este caso, cada uno de los operandos tiene 8 bits, de modo que la longitud de palabra del producto es de 16 bits.

La longitud de fraccion del producto es la suma de las longitudes de fraccion de los operandos, en este caso es $5 + 5 = 10$ bits. La suma de la longitud de palabra crece un poco mas de forma importante para dar cabida a la posibilidad de una transferencia de bits.

La longitud de fraccion de la suma esta alineado con la longitud de fraccion de los sumandos y todos los bits se mantienen para la plena precision. En este caso ambos sumandos tienen una longitud de fraccion de 5 bits, de modo que la suma de contiene una longitud de fraccion tambien de 5 bits.

A.26.3 KeepLSB

Ahora estableciendo el PRODUCTMODE y SUMMODE para a y b del modo KEEPLSB:

```
>>F.ProductMode = 'KeepLSB';
>>F.ProductWordLength = 12;
>>F.SumMode = 'KeepLSB';
>>F.SumWordLength = 12;
>>a.fimath = F;
>>b.fimath = F;
>>a
a =
    3.1563                %011.00101
    s8,5
>>b
b =
    2.7188                %010.10111
    s8,5
>>a*b
ans =
    0.5811                %00.1001010011
    s12,10
>>a+b
ans =
    5.8750                %0000101.11100
    s12,5
```

En el modo KEEPLSB se puede especificar la longitud de palabra y el resultado se da con los bits menos significativos de manera automatica. El producto de la longitud de fraccion es la suma de las longitudes de fraccion de cada uno de los operandos. En este caso cada operando tiene una longitud de fraccion de 5 bits, de modo que la longitud de fraccion del producto es de 10 bits y los bits se guardan de manera fraccional.

El overflow ocurre porque el resultado para fullprecision requiere numeros enteros de 6 bits, y solo permanecen en el producto numero enteros de 2 bits.

La longitud de fraccion de la suma esta alineada con la longitud de fraccion de los sumandos y en este caso, todos los bits menos significativos se mantienen; cada sumando tiene una longitud de fraccion de 5 bits, de modo que la longitud de las fraccion de la suma es de 5 bits. El resultado a fullprecision requiere de 4 bits y 7 bits permanecen en la suma, por lo que no se produce overflow en la suma.

A.26.4 KeepMSB

Ahora se establece el modo KEEPMSB, para las propiedades PRODUCTMODE y SUMMODE:

```
>>F.ProductMode = 'KeepMSB';
>>F.ProductWordLength = 12;
>>F.SumMode = 'KeepMSB';
>>F.SumWordLength = 12;
>>a.fimath = F;
>>b.fimath = F;
>>a
a =
    3.1563                %011.00101
    s8,5
>>b
b =
    2.7188                %010.10111
    s8,5
>>a*b
ans =
    8.5781                %001000.100101
    s12,6
>>a+b
ans =
    5.8750                %0101.11100000
    s12,8
```

En el modo KEEPMSB, se puede especificar la longitud de palabra y el resultado del producto mantiene los bits mas significativos de forma automatica. Este modo es empleado en muchos dispositivos DSP en el que el producto y la suma se mantiene en los registros doble-wide, donde el programador elige para transferir los bits mas significativos de los registros después de cada operación.

26.5 PRECISION ESPECIFICA (SpecifyPrecision)

Ahora se establece el modo SPECIFYPRECISION, para las propiedades PRODUCTMODE y SUMMODE:

```
>>F.ProductMode = 'SpecifyPrecision';
>>F.ProductWordLength = 8;
>>F.ProductFractionLength = 7;
>>F.SumMode = 'SpecifyPrecision';
>>F.SumWordLength = 8;
>>F.SumFractionLength = 7;
>>a.fimath = F;
>>b.fimath = F;
```

```

>>a
a =
    3.1563                                %011.00101
    s8,5
>>b
b =
    2.7188                                %010.10111
    s8,5
>>a*b
ans =
    0.5781                                %0.1001010
    s8,7
>>a+b
ans =
   -0.1250                                %1.1110000
    s8,7

```

En el modo SPECIFYPRECISION se especifica tanto como la longitud de palabra, como la longitud de fraccion para la suma y el producto. en este ejemplo se utiliza el formato fraccionario para los productos y las sumas, con una longitud de palabra de 8 bits y una longitud de fraccion de 7 bits.

A.27 FUNCIONES DEL OBJETO FIMATH

Las siguientes funciones operan directamente sobre los objetos FIMATH:

- add
- disp
- fimath
- get
- isequal
- isfimath
- mpy
- sub

A.28 CONSTRUCCION DE OBJETOS FIPREF

El objeto FIPREF define todos los atributos del objeto FI. Se puede utilizar la funcion FIPREF para crear un nuevo objeto. Para empezar, el tipo:

```
>>P = fipref
```

Para crear un objeto predeterminado FIPREF:

```

>>P =
    NumberDisplay: 'RealWorldValue'

```

```
NumericTypeDisplay: 'full'  
FimathDisplay: 'full'  
LoggingMode: 'Off'  
DataTypeOverride: 'ForceOff'
```

La sintaxis `P = fipref(...'PropertyName','PropertyValue'...)` permite configurar las propiedades de un objeto FIPREF para crear objetos con propiedades nombre/propiedad. Use `reset (fipref)` para volver a la configuración por defecto durante las secciones. Se usa `savefipref` para guardar las preferencias en Matlab para posteriores usos.

A.29 PROPIEDADES DEL OBJETO FIPREF

A.29.1 PROPIEDADES DE TYPE OVERRIDE Y LOGGING

Las siguientes propiedades de los objetos FIPREF son siempre de escritura:

- `FimathDisplay`: muestra opciones para los atributos FIMATH de un objeto FI.
- `DataTypeOverride`: opciones de tipos de datos.
- `LoggingMode`: opciones de registro de las operaciones realizadas sobre los objetos FI.
- `NumericTypeDisplay`: muestra opciones para el tipo numérico de los atributos de un objeto FI.
- `NumberDisplay`: opciones de visualización del valor de un objeto FI.

Hay dos maneras de especificar las propiedades de los objetos FIPREF en el toolbox de punto fijo.

A.29.2 CONFIGURACION DE LAS PROPIEDADES FIPREF EN LA CREACION DE OBJETOS

Se puede configurar las propiedades de los objetos FIPREF en el momento de la creación de objetos mediante la inclusión de las propiedades después de los argumentos de la función creadora de FIPREF. Por ejemplo, para establecer `NumberDisplay` en bin y `NumericTypeDisplay` en short:

```
>>P = fipref('NumberDisplay', 'bin','NumericTypeDisplay', 'short')  
P =  
    NumberDisplay: 'bin'  
    NumericTypeDisplay: 'short'  
    FimathDisplay: 'full'  
    LoggingMode: 'Off'  
    DataTypeOverride: 'ForceOff'
```

A.29.3 USO DE LA PROPIEDAD DE REFERENCIA DIRECTA CON FIPREF

Se puede referenciar directamente en una propiedad para establecer o recuperar el valor de la propiedad de un objeto FIPREF utilizando Matlab como estructura de referencia. Para ello se puede utilizar un periodo de indice en una propiedad por su nombre. Por ejemplo, para obtener la NumberDisplay de P:

```
>>P.NumberDisplay
ans =
    bin
```

Para establecer el NumericTypeDisplay de P:

```
>>P.NumericTypeDisplay = 'full'
P =
    NumberDisplay: 'bin'
  NumericTypeDisplay: 'full'
    FimathDisplay: 'full'
    LoggingMode: 'Off'
  DataTypeOverride: 'ForceOff'
```

A.30 USO DE UN OBJETO FIPREF PARA ESTABLECER LAS PREFERENCIAS DE VISUALIZACION

Se puede utilizar el objeto FIPREF para determinar tres aspectos de la visualizacion de un objeto FI:

- Como el valor de un objeto FI es mostrado.
- Como las propiedades del objeto FIMATH es mostrado.
- Como las propiedades NUMERCITYPE es mostrado.

Por ejemplo, el siguiente ejemplo, muestra por defecto el FIPREF de un objeto FI:

```
>>a = fi(pi)
a =
    3.1416
  DataTypeMode: Fixed-point: binary point scaling
        Signed: true
    WordLength: 16
  FractionLength: 13
        RoundMode: nearest
    OverflowMode: saturate
    ProductMode: FullPrecision
MaxProductWordLength: 128
        SumMode: FullPrecision
MaxSumWordLength: 128
    CastBeforeSum: true
```

Ahora, al cambiar las propiedades FIPREF:

```
>>P = fipref;
>>P.NumberDisplay = 'bin';
>>P.NumericTypeDisplay = 'short';
>>P.FimathDisplay = 'none'
P =
    NumberDisplay: 'bin'
    NumericTypeDisplay: 'short'
    FimathDisplay: 'none'
    LoggingMode: 'Off'
    DataTypeOverride: 'ForceOff'
>>a
a =
0110010010001000
    s16,13
```

A.31 USO DE OBJETOS FIPREF PARA ESTABLECER LAS PROPIEDADES DE LOGGING

A.31.1 ADVERTENCIAS DE LOGGING OVERFLOW Y UNDERFLOWS

el overflow y el underflows que se registran como advertencias para todas asignaciones, signo mas, menos, multiplicación y todas operaciones cuando la propiedad LOGGINGMODE del FIPREF esta establecida. Por ejemplo:

1. Crear un objeto FI con signo, que es un vector de valores de 1 a 5, con una longitud de palabra de 8 bits y longitud de fraccion de 6 bits.
2. `a = fi(1:5,1,8,6);`
3. Define el objeto FIMATH asociado con a e indique especificamente la longitud de palabra y de fraccion del producto y de la suma.
4. `F = a.fimath;`
5. `F.SumMode = 'SpecifyPrecision';`
6. `F.ProductMode = 'SpecifyPrecision';`
7. `a.fimath = F;`
8. Define en objeto FIPREF y cambie el logging a overflow o underflow.
9. `P = fipref;`
10. `P.LoggingMode = 'on';`
11. Omite el `numericType` y el `fimath` para mostrar.
12. `P.NumericTypeDisplay = 'none';`
13. `P.FimathDisplay = 'none';`
14. Especifique la longitud de fraccion de la suma y del producto.
15. `a.SumWordLength = 16;`
16. `a.SumFractionLength = 15;`

```

17. a.ProductWordLength = 16;
18. a.ProductFractionLength = 15;
19. Se puede mostrar un warnings para asignar la operacion del overflows y
    el underflows. Por ejemplo:
20. a(1) = pi
21. Warning: 1 overflow occurred in the fi assignment operation.
22. a =
23.    1.9844    1.9844    1.9844    1.9844    1.9844
24. a(1) = double(eps(a))/10
25. Warning: 1 underflow occurred in the fi assignment operation.
26. a =
27.     0    1.9844    1.9844    1.9844    1.9844
28. Warnings are displayed for overflows and underflows in addition and
    subtraction operations. For example, try:
29. a+a
30. Warning: 12 overflows occurred in the fi + operation.
31. ans =
32.     0    1.0000    1.0000    1.0000    1.0000
33. a-a
34. Warning: 8 overflows occurred in the fi - operation.
35. ans =
36.     0     0     0     0     0
37. Warnings are displayed for overflows and underflows in multiplication
    operations. For example, try:
38. a.*a
39. Warning: 4 product overflows occurred in the fi .* operation.
40. ans =
41.     0    1.0000    1.0000    1.0000    1.0000
42. a*a'
43. Warning: 4 product overflows occurred in the fi * operation.
44. Warning: 3 sum overflows occurred in the fi * operation.
45. ans =
46.    1.0000

```

El ultimo ejemplode arriba es una multiplicacion compleja que requiere tanto operaciones de multiplicacion y suma. Las advertencias indican si hay overflows o underflows en ambos.

Debido a los overflows y underflows se registraran como advertencias, se puede utilizar la funcion dbstop de Matlab para encontrar las lineas exactas en el archivo M-archivo que este causando overflows o underflows.

A.31.2 ACCESO A INFORMACION LOGGED CON FUNCIONES:

Cuando la propiedad LOGGINNMODE del FIPREF esta establecido, se puede utilizar las siguientes funciones para volver al registro de informacio acerca de la creaci3n y las operaciones en la lineas de comandos de Matlab:

- maxlog : devuelve el valor maximo de la palabra real.
- minlog : vuelve el valor minimo
- noverflows :devuelve el numero de overflow.
- nunderflows :devuelve el numero de underflow.

EI LOGGINGMODE debe estableceser antes de llevar a cabo cualquier operacion con el fin de acceder a la informacion sobre el mismo. Para borrar el registro utilice la funcion RESETLOG. Por ejemplo, considere lo siguiente: en primer lugar logging on, a continuacion realice las operaciones y por ultimo obtenga infomracion sobre las operaciones:

```
>>fipref('LoggingMode','on');
>>x = fi([-1.5 eps 0.5], true, 16, 15);
>>x(1) = 3.0;
>>maxlog(x)
ans =
     3
>>minlog(x)
ans =
    -1.5000
>>noverflows(x)
ans =
     2
>>nunderflows(x)
ans =
     1
```

a continuacion reinicie el registro y la solicitud de la misma informacion de nuevo. Tenga en cuenta que las funciones de registro estan vacias [], ya que el logging se ha restaurado, ya que las operaciones se ejecutan:

```
>>resetlog(x)
>>maxlog(x)
ans =
     []
>>minlog(x)
ans =
     []
>>noverflows(x)
ans =
     []
>>nunderflows(x)
```

```
ans =  
[]
```

A.32 USO DEL OBJETO FIPREF PARA FIJAR EL TIPO DE DATOS

A.32.1 SOBRESERIBIENDO EL TIPO DE DATOS DE UN OBJETO FI

Use la propiedad DATATYPEOVERRIDE del FIPREF para anular los objetos FI con simple, doble o escudo doble. El tipo de datos override solo ocurre cuando la función del creador FI es llamado. Los objetos que se crean al mismo tiempo sobre los tipos de datos son anulados por el nuevo tipo de dato.

```
>>p = fipref('DataTypeOverride', 'TrueDoubles')  
p =  
    NumberDisplay: 'RealWorldValue'  
    NumericTypeDisplay: 'full'  
    FimathDisplay: 'full'  
    LoggingMode: 'Off'  
    DataTypeOverride: 'TrueDoubles'  
>>a = fi(pi)  
a =  
    3.1416  
    DataTypeMode: double  
>>p = fipref('DataTypeOverride', 'ForceOff')  
p =  
    NumberDisplay: 'RealWorldValue'  
    NumericTypeDisplay: 'full'  
    FimathDisplay: 'full'  
    LoggingMode: 'Off'  
    DataTypeOverride: 'ForceOff'  
>>a  
a =  
    3.1416  
    DataTypeMode: double  
>>b = fi(pi)  
b =  
    3.1416  
    DataTypeMode: Fixed-point: binary point scaling  
    Signed: true  
    WordLength: 16  
    FractionLength: 13
```

NOTA: para restablecer el objeto FIPREF se deben resetear los valores , usando reset(FIPREF) o reset (p), donde p es el objeto FIPREF. Esto es

util para garantizar que los tipos de datos override y logging esten desactivadas.

A.32.2 USANDO LOS TIPOS DE DATOS OVERRIDE PARA ESTABLECER LA ESCALIZACION DE PUNTO FIJO

La eleccion de la escalizacion de las variables de punto fijo en los algoritmos pueden ser dificiles. En el toolbox de punto fijo, se puede usar una combinacion de tipos de datos override y logging min/max para ayudar a descubrir el rango numerico que el tipo de datos punto fijo necesita cubrir. Este rango dicta la propiedad de escalizacion para los tipos de datos punto fijo. en general, este es el procedimiento:

1. Implemente su algoritmo usando un objeto punto fijo FI, use inicialmente las mejores características de la longitud de palabra y de escalizacion.
2. Establezca la propiedad `DATATYPEOVERRIDE` del FIPREF en el modo de `scaledoubles`, `trueingles` o `truedoubles`.
3. establezca la propiedad `loggingmode` del FIPREF
4. utilice las funciones de registro `maxlong` y `minlong` para los valores maximos y minimos alcanzados por las variables del algoritmo en el modo de punto flotante.
5. establezca la propiedad `datatypeoverride` del FIPREF de modo `FORCEOFF`.
6. utilice la informacion obtenida en el paso numero 4 para ajustar la escalizacion de cada variable de punto fijo del algoritmo de tal manera que el rango numerico completo de cada variable es representada por su tipo de datos y por la escalizacion.

A.33 FUNCIONES DEL OBJETO FIPREF

Las siguientes funciones se operan directamente sobre los objetos FIPREF:

- `disp`
- `fipref`
- `get`
- `reset`
- `savefipref`

A.34 CONSTRUCCION DE OBJETOS NUMERICTYPE

A.34.1 SINTAXIS DE LOS OBJETOS NUMERICTYPE

Los objetos `numerictype` definen los tipos de datos y la propiedad de escalizacion de los objetos FI. Se pueden crear objetos `numerictype` en el toolbox de punto fijo de dos formas:

- se puede usar la function de creacion del numericitype para crear un Nuevo objeto.
- Se puede usar la function de creacion del numericitype para copier un objeto numericitype ya creado.

Crear el objeto numericitype:

```
>>T = numericitype
T =
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 16
    FractionLength: 15
```

Se puede usar la function de constructor NUMERICYTPE de las siguientes maneras:

- T = numericitype crea por defecto del objeto numercitype.
- T = numericitype(s) crea un objeto numericitype punto fijo con escalizacion, signo s y longitud de palabra de 16 bits.
- T = numericitype(s,w) crea un objeto punto fijo numericitype con escalizacion, signo (s) y longitud de palabra (w).
- T = numericitype(s,w,f) crea un objeto punto fijo numericitype con escalizacion, signo (s),longitud de palabra (w) y longitud de fraccion (f).
- T = numericitype(s,w,slope,bias) crea un objeto punto fijo numericitype con escalizacion (pendiente inclinacion), signod (s) y pendiente y inclinacion.
- T = numericitype(s,w,slopeadjustmentfactor,fixedexponent,bias) crea un objeto punto fijo numericitype con escalizacion (pendiente inclinacion), signod (s), factor de ajuste de la pendiente, punto exponente y inclinacion..
- T = numericitype(property1,value1, ...) permite configurar las propiedades de un objeto usando la propiedad numericitype de la forma nombre/valor.
- T = numericitype(T1, property1, value1, ...)permite hacer un copia de un objeto numericitype existente, mientras que se modifican algunos o todos los valores de la propiedad.
- T = numericitype('double') crea un objeto numericitype doble.
- T = numericitype('single') creat un objeto numericitype sencillo.
- T = numericitype('boolean') crea un objeto numericitype booleano.

A.34.2 EJEMPLOS PARA LA CONSTRUCCION DE OBJETOS NUMERICITYPE

Por ejemplo, lo siguiente un objeto numericitype con signo con una longitud de palabra de 32 bits y una longitud de fraccion de 30 bits.

```
>>T = numericitype(1, 32, 30)
```

```
T =
    DataTypeMode: Fixed-point: binary point scaling
        Signed: true
        WordLength: 32
        FractionLength: 30
```

Si se omite el argumento f, la escalizacion no es especificada.

```
>>T = numerictype(1, 32)
T =
    DataTypeMode: Fixed-point: unspecified scaling
        Signed: true
        WordLength: 32
```

Si se omiten los argumentos w y f, la longitud de palabra es de 16 bits de fomra automatica y la escalizacion no esta especificada.

```
>>T = numerictype(1)
T =
    DataTypeMode: Fixed-point: unspecified scaling
        Signed: true
        WordLength: 16
```

A.34.3 CONSTRUCCION DE UN OBJETO NUMERICTYPE CON LA PROPIEDAD DE VALORES PARES (NOMBRE/PROPIEDAD)

Cuando se usa la propiedad nombre/propiedad con valores de pares establecen la propiedad numerictype cuando se crea un objeto.

```
>>T = numerictype('Signed', true, 'DataTypeMode','Fixed-point: slope and
    bias', 'WordLength', 32, 'Slope',2^-2, 'Bias', 4)
T =
    DataTypeMode: Fixed-point: slope and bias scaling
        Signed: true
        WordLength: 32
        Slope: 0.25
        Bias: 4
```

A.34.4 COPIA DE UN OBJETO NUMERICTYPE

Para copier un objeto numerictype, se usa de la siguiente manera:

```
>>T = numerictype;
>>U = T;
>>isequal(T,U)
ans =
```

A.35 PROPIEDADES DEL OBJETO NUMERICTYPE

A.35.1 PROPIEDADES DE TIPO DE DATO Y ESCALIZACION

Todas las propiedades de los objetos numerictype son de escritura. Sin embargo, la propiedad numerictype de un objeto Fim no se puede grabar una vez que el objeto FI ha sido creado:

- Bias : inclinacion
- DataType: categoria del tipo de datos Data type category
- DataTypeMode: modo del tipo de dato y escalizacion
- FixedExponent : punto fijo exponencial
- SlopeAdjustmentFactor: factor de ajuste de la pendiente.
- FractionLength : longitud de fraccion del valor entero almacenado en bits.
- Scaling: modo de escalizacion en punto fijo
- Signed : con o sin signo
- Slope : pendiente
- WordLength : longitud de palabra de los valores enteros almacenados en bits.

A.35.2 CONFIGURACION DE LA PROPIEDAD NUMERICTYPE AL CREAR UN OBJETO

Se puede establecer las propiedades de un objeto numerictype al tiempo de crear un objeto incluyendo las propiedades antes de argumentar la funcion de creacion del objeto numerictype. Por ejemplo, establezca la longitud de palabra de 32 bits y la longitud de fraccion de 30 bits:

```
>>T = numerictype('WordLength', 32, 'FractionLength', 30)
```

```
T =
```

```
    DataTypeMode: Fixed-point: binary point scaling
```

```
        Signed: true
```

```
    WordLength: 32
```

```
    FractionLength: 30
```

A.35.3 USO DE LAS PROPIEDADES DIRECTAS DE REFERENCIA CON LOS OBJETOS NUMERICTYPE

SE PUEDE REFERENCIAR DIRECTAMENTE en una propiedad para establecer o recuperar el valor de la propiedad de un objeto numericitype utilizando Matlab como estructura de refeencia. Para ello se puede utiliza un periodo de indice para la propiedad por su nombre. Por ejemplo, para obtener la longitud de palabra de T:

```
>>T.WordLength
ans =
    32
```

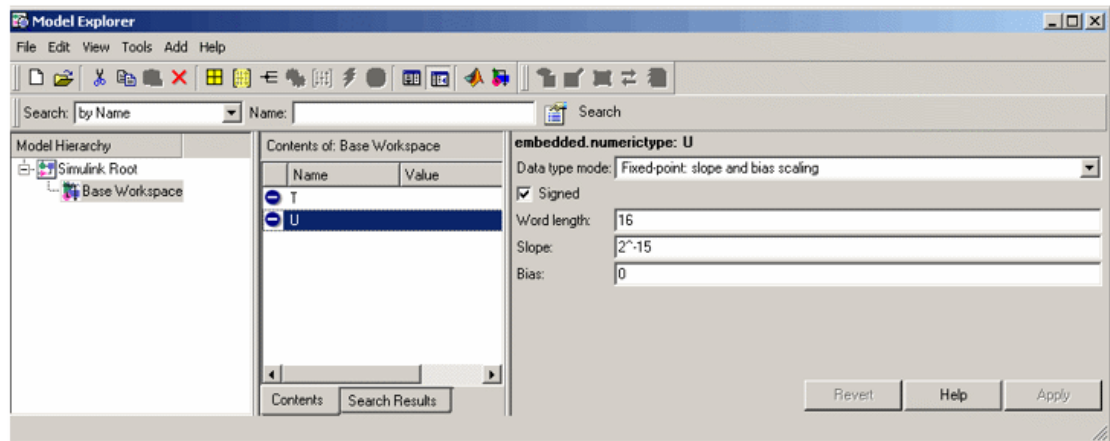
para establecer la longitud de fraccion de T:

```
>>T.FractionLength = 31
T =
    DataTypeMode: Fixed-point: binary point scaling
           Signed: true
           WordLength: 32
           FractionLength: 31
```

A.35.4 CONFIGURACION DE LA PROPIEDAD NUMERICTYPE EN EL MODELO EXPLORADOR

Se puede ver y cambiar las propiedades para todo objeto numericitype definido en el workspace de Matlab en el modelo explorador. Abra el model explorer para seleccionar **View > Model Explorer** en cualquier modelo de simulink. La siguiente figura muestra el Model Explorer cuando se define el siguiente objeto numericitype en el workspace de Matlab:

```
>>T = numericitype
T =
    DataTypeMode: Fixed-point: binary point scaling
           Signed: true
           WordLength: 16
           FractionLength: 15
>>U = numericitype('DataTypeMode', 'Fixed-point: slope and bias')
U =
    DataTypeMode: Fixed-point: slope and bias scaling
           Signed: true
           WordLength: 16
           Slope: 2^-15
           Bias: 0
```



A.36 ESTRUCTURA DEL NUMERICTYPE

A.36.1 POSIBLES VALORES DE LA ESTRUCTURA DE LA PROPIEDAD NUMERICTYPE

El objeto numerictype contiene todos los tipos de datos y los atributos de escalización de un objeto FI. Maneja la misma estructura que Matlab, con la excepción de que solo permite establecer valores válidos de los campos definidos. La siguiente tabla muestra la posible configuración de cada uno de los campos de la estructura que son válidos para los objetos FI.

MODULO DEL TIPO DEL DATO	TIPO DE DATO	ESCALIZACION	SIGNO	LONGITUD DE PALABRA	LONGITUD DE FRACCION	PENDIENTE	INCLINACION
<i>Especificación plena de los tipos de datos de punto fijo</i>							
Punto fijo: punto binario escalizado	Fijo	Punto binario	1/0 verdadero/falso	Entero positivo desde 1 a 65,536	Entero positivo o negativo	1	0
Punto fijo: pendiente y inclinación escalizado	Fijo	Pendiente Inclinación	1/0 verdadero/falso	Entero positivo desde 1 a 65,536	N/A	Solo números de punto flotante	Solo números de punto flotante
parcialmente especificado para datos de punto fijo							

MODO DEL TIPO DEL DATO	TIPO DE DATO	ESCALIZACION	SIGNO	LONGITUD DE PALABRA	LONGITUD DE FRACCION	PENDIENTE	INCLINACION
Punto fijo: escalizacion especificada	Fijo	No especificado	1/0 verdadero/falso	Entero positivo desde 1 a 65,536	N/A	N/A	N/A
<i>Especificacion plena para tipo de datos de escala doble</i>							
Scala doble: punto binario escalizado	Scala doble	Punto binario	1/0 verdadero/falsos	Entero positivo desde 1 a 65,536	Entero positivo o negativo	1	0
Scala doble: pendiente y inclinacion escalizado	Scala doble	Pendiente inclinacion	1/0 verdadero/falso	Entero positivo desde 1 a 65,536	N/A	Solo numeros de punto flotante	Solo numeros de punto flotante
<i>Parcialmente especificado para tipo de datos de escala doble</i>							
Scala doble: escalizacion sin especificar	Scala doble	Sin especificar	1/0 verdadero/falso	Entero positivo desde 1 a 65,536	N/A	N/A	N/A
<i>Tipos de datos incorporados</i>							
Doble	doble	N/A	1 verdadero	64	0	1	0
Simple	simple	N/A	1 verdadero	32	0	1	0
Booleano	booleano	N/A	0 falso	1	0	1	0

A.36.2 PROPIEDADES QUE AFECTAN A LA PENDIENTE

El campo de la pendiente en la estructura numerictype esta relacionada con el slopeadjustmentfactor y las propiedades de fixedexponent.

$$slope = slope\ adjustment\ factor \times 2^{fixed\ exponent}$$

El `fixedexponent` y la propiedad `fractionlength` estan relacionadas por:

$$fixed\ exponent = -fraction\ length$$

si se establece las propiedades de `slopeadjustmentfactor`, `fixedexponent` o `fractionlength` el campo de la pendiente esta modificado.

A.36.3 VALOR ENTERO ALMACENADO Y VALOR DE LA PALABRA REAL

En el `numerictype`, las propiedades de `storedintegervalue` y `realwordvalue` se relacion segun:

$$real\ -\ world\ value = stored\ integer\ value \times 2^{fraction\ length}$$

lo que equivale a:

$$real\ -\ world\ value = stored\ integer\ value \times (slope\ adjustment\ factor \times 2^{fixed\ exponent}) + bias$$

si alguna de estas propiedades se actualiza, las otras son modificadas en consecuencia.

A.37 USO DE LOS OBJETOS NUMERICTYPE PARA COMPARTIR LOS TIPOS DE DATOS Y LA CONFIGURACION DE LA ESCALIZACION

Se puede utilizar un objeto `numerictype` para definir el tipo de datos y la escalizacion comun que se desea usar para muchos objetos FI. A continuación, se puede crear varios objetos FI, utilizando el mismo `numerictype` para cada objeto. El siguiente ejemplo muestra la creación de un objeto `numerictype`, que luego se utiliza para crear dos objetos FI con los mismos atributos `numerictype`:

```
>>format long g
>>T = numerictype('WordLength',32,'FractionLength',28)
T =
    DataTypeMode: Fixed-point: binary point scaling
      Signed: true
    WordLength: 32
    FractionLength: 28
>>a = fi(pi,T)
a =
    3.1415926553309
```

```

        DataTypeMode: Fixed-point: binary point scaling
          Signed: true
          WordLength: 32
        FractionLength: 28
          RoundMode: nearest
          OverflowMode: saturate
          ProductMode: FullPrecision
        MaxProductWordLength: 128
          SumMode: FullPrecision
        MaxSumWordLength: 128
        CastBeforeSum: true
>>b = fi(pi/2, T)
b =
    1.5707963258028
        DataTypeMode: Fixed-point: binary point scaling
          Signed: true
          WordLength: 32
        FractionLength: 28
          RoundMode: nearest
          OverflowMode: saturate
          ProductMode: FullPrecision
        MaxProductWordLength: 128
          SumMode: FullPrecision
        MaxSumWordLength: 128
        CastBeforeSum: true

```

El siguiente ejemplo muestra la creacion de un objeto `numerictype` con escalamiento [pendiente sesagad], que se utiliza para crear dos objetos FI con los mismo atributos `numerictype`:

```

>>T = numerictype('scaling','slopebias','slope', 2^2, 'bias', 0)
T =
        DataTypeMode: Fixed-point: slope and bias scaling
          Signed: true
          WordLength: 16
            Slope: 2^2
            Bias: 0
>>c = fi(pi, T)
c =
    4
        DataTypeMode: Fixed-point: slope and bias scaling
          Signed: true
          WordLength: 16
            Slope: 2^2
            Bias: 0
          RoundMode: nearest

```

```

        OverflowMode: saturate
        ProductMode: FullPrecision
MaxProductWordLength: 128
        SumMode: FullPrecision
MaxSumWordLength: 128
        CastBeforeSum: true
>>d = fi(pi/2, T)
d =
    0
        DataTypeMode: Fixed-point: slope and bias scaling
        Signed: true
        WordLength: 16
        Slope: 2^2
        Bias: 0
        RoundMode: nearest
        OverflowMode: saturate
        ProductMode: FullPrecision
MaxProductWordLength: 128
        SumMode: FullPrecision
MaxSumWordLength: 128
        CastBeforeSum: true

```

A.38 FUNCIONES DEL OBJETO NUMERICTYPE

Las siguientes funciones operan directamente sobre los objetos numerictype:

- disp
- divide
- get
- isequal
- isnumerictype

A.39 CONSTRUCCION DE OBJETOS CUANTIZADOS

Se puede utilizar el objeto cuantizador para establecer el conjunto de datos cuantizados. Sepuede crear objetos cuantizadors en el toolbox de punto fijo de dos formas:

- Se puede utilizar la funcion constructor de cuantizacion para crear un nuevo objeto.
- Se puede utilizar la funcion constructora de cuantizacion para copiar un objeto cuantizado.

Para crear un objeto cuantizado con propiedades especificas:

```
>>q = quantizer
```

```

q =
    DataMode = fixed
    RoundMode = floor
    OverflowMode = saturate
    Format = [16 15]

```

Para copiar un objeto cuantizado, solo se tiene que utilizar la asignacion como lo muestra la siguiente ejemplo:

```

>>q = quantizer;
>>r = q;
>>isequal(q,r)
ans =
    1

```

una lista de todas las propiedades del objeto cuantizador q que acaba de crear se muestra junto con el valor de la propiedad correspondiente. Todos los valores de las propiedad se establecen por defecto cuando se haya construido un objeto cuantizador de esta manera.

A.40 PROPIEDADES DEL OBJETO DE CUANTIZACION

Las siguientes propiedades del objeto de cuantizacion son siempre escribibles:

- DataMode : tipo de aritmetica usada en la cuantizacion.
- Format : formato del dato de los objeto de cuantizacion.
- OverflowMode : modo de overflow.
- RoundMode : modo de redondeo.

Por ejemplo, para crear un objeto cuantizado de punto fijo:

- El formato del valor de la propiedad se establece para [16,14]
- El valor de la propiedad overflowmode se establece en 'saturate'
- El valor de la propiedad roundmode se establece en 'ceil'

```

>>q = quantizer('datamode','fixed','format',[16,14],'overflowmode',
    'saturate','roundmode','ceil')

```

no se debe incluir el nombre de la propiedad de cuantizacion cuando se establezcan los valores de las propiedades del objeto de cuantizacion.por ejemplo, se puede crear un objeto cuantizado q desde uno ya hecho.

```

>>q = quantizer('fixed',[16,14],'saturate','ceil')

```

Nota no se deben incluir los valores por defecto de las propiedades cuando se construyen un objeto cuantizado.

A.41 DATOS A CUANTIZAR DE UN OBJETO CUANTIZADO

Al crear un objeto cuantizado se deben especificar los parametros de cuantizacion a usar cuando se cuantizan un conjunto de datos. Se puede utilizar la funcion de cuantizacion de los datos de acuerdo con una serie de condiciones para dicha cuantizacion. Una vez que los datos a cuantizar esten dados en el objeto cuantizador, su estado puede cambiar en cualquier momento:

- Como usar `quantize` para los datos cuantizados.
- Como afecta a la cuantizacion los estados del objeto `quantizer`.
- Como reestablecer los estados de objeto `quantizer` para poder usar los valores ya establecidos por defecto.

1. construir un ejemplo de datos establecidos y un objeto cuantizado.
2. `format long g`
3. `randn('state',0);`
4. `x = randn(100,4);`
`q = quantizer([16,14]);`
5. revisar los valores de los estados del `maxlog` y `noverflows`.
6. `q.maxlog`
7. `ans =`
8. `-1.79769313486232e+308`
9. `q.noverflows`
10. `ans =`
`0`

Notese, que el `maxlog` es igual a `realmax`, que indica que el cuantizador `q` esta en un estado de reset.

11. cuantiza el dato estableciendo lo acordado para las especificaciones del objeto `quantizer`.
12. `y = quantize(q,x);`
`Warning: 15 overflows.`
13. observe el valor de `maxlog` y `noverflows`.
14. `q.maxlog`
15. `ans =`
16. `1.99993896484375`
17. `q.noverflows`
18. `ans =`
`15`

Note que el valor maximo logrado resulta despues de la cuantizacion y es `q.maxlog == max(y)`.

19. resetee los estados de cuantizacion y reviselo.

```

20. reset(q)
21. q.maxlog
22. ans =
23.     -1.79769313486232e+308
24. q.noverflows
25. ans =
    0

```

A.42 TRANSFORMACIONES DE LOS DATOS CUANTIZADOS

Se puede convertir los valores de los datos numericos en valores hexadecimales o binarios de acuerdo con las condiciones del objeto de cuantizacion.

Use

- num2bin : para convertir datos a binario.
- num2hex : para convertir datos a hexagecimal.
- hex2num : para convertir datos hexagecimal a numericos.
- bin2num : para convertir datos binarios a numericos.

Por ejemplo:

```

>>q = quantizer([3 2]);
>>x = [0.75 -0.25; 0.50 -0.50; 0.25 -0.75;0 -1];
>>b = num2bin(q,x)
b =
011
010
001
000
111
110
101
100

```

produce la representacion fraccional en complemento a dos con representacion de punto binario de 3 bits.

A.43 FUNCIONES DEL OBJETO CUANTIZADOR

Las siguientes funciones se operan directamente en un objeto cuantizador:

```

bin2num
copyobj
denormalmax
denormalmin

```

disp
eps
exponentbias
exponentlength
exponentmax
exponentmin
fractionlength
get
hex2num
isequal
length
max
min
noperations
noverflows
num2bin
num2hex
num2int
nunderflows
quantize
quantizer
randquant
range
realmax
realmin
reset
round
set
tostring
wordlength

A.44 APOYO DE LAS FUNCIONES Y LIMITACIONES DEL EMBEDDED DE PUNTO FIJO EN MATLAB

El Embedded MATLAB es un subconjunto limitado de los idiomas que ofrece matlab para proveer las optimizaciones para:

- Generación eficiente, calidad de producción de código en C para aplicaciones. Embedded de matlab restringe la semántica para satisfacer la memoria y el tipo de datos requeridos para el embedded necesario.
- Aceleración de los algoritmos de punto fijo

El subconjunto Embedded de Matlab apoya un número importante de funciones en el toolbox de punto fijo que se enumeran en la siguiente lista. Las siguientes limitaciones generales siempre se aplican a la utilización del toolbox de punto fijo con Embedded Matlab:

- Los objetos fipref y quantizer no son apoyados..

- El punto de notaciones soportado para obtener los valores de las propiedades fimath y numericity. La notacion no es compatible con objetos FI y no es compatible con el establecimiento propiedades.
- Longitud de palabra tan larga de 32 bits no tienen soporte.
- Es ilegal el cambio del fimath o del numericity de una variable que ha sido creada.
- Los valores booleanos y de escala doble de las propiedades datatype mode y del datatype no son soportadas.
- El redondeo convergente no es soportado.
- El valor falso de la propiedad CastBeforeSum property no es soportada.
- La función numel trabaja con el numel de MATLAB para objetos FI en el Embedded MATLAB, y se retorna como un 1 en el toolbox de punto fijo.

A.44.1 FUNCIONES DEL TOOLBOX DE PUNTO FIJO PARA EL USO DEL EMBEDDED MATLAB

FUNCION	OBSERVACIONES/LIMITACIONES
abs	—
all	—
any	—
bitand	—
bitandreduce	—
bitcmp	—
bitconcat	—
bitget	—
bitor	—
bitorreduce	—
bitrol	—
bitror	—
bitset	—
bitshift	—
bitsliceget	—
bitsll	—
bitsra	—
bitsrl	—

FUNCION	OBSERVACIONES/LIMITACIONES
bitxor	—
bitxorreduce	—
complex	—
conj	—
ctranspose	—
diag	—
disp	—
divide	• Cualquier entrada de FI no debe ser constante, su valor debe ser conocido en tiempo de compilación para que pueda ser emitido a un objeto FI • Los divisores complejos e imaginarios no son soportados.
double	—
end	—
eps	• Soporta solo señales para escala punto fijo. • Soporta señales con escala, vector y matrices simples y dobles.
eq	• No soporta señales con diferentes inclinaciones de punto fijo.
fi	• Use Embedded MATLAB para crear constantes o variables punto fijo. • El sintaxis que construye por defecto los argumentos de entrada no son soportados. • La sintaxis <code>fi('PropertyName',PropertyValue)</code> no es soportada. Para usar nuevas propiedades con valores pares nombre/propiedad, se especifica primero el valor <code>v</code> del objeto FI en <code>fi(v,'PropertyName',PropertyValue)</code>. • Trabaja solo con constantes de valores de entrada; que es el valor de la entrada que se conoce en el tiempo de compilación. • La información del objeto <code>numericType</code> es viable para entradas de Simulink que no sean punto fijo.
fimath	• Para crear un objeto <code>fimath</code> se usa el código en Embedded MATLAB.

FUNCION	OBSERVACIONES/LIMITACIONES
ge	Las señales con diferentes inclinaciones en punto fijo no son soportadas.
get	La sintaxis structure = get(o) no esta soportada.
getlsb	—
getmsb	—
gt	Las señales con diferentes inclinaciones en punto fijo no son soportadas.
horzcat	—
imag	—
int8, int16, int32	—
iscolumn	—
isempty	—
isfi	—
isfimath	—
isfinite	—
isinf	—
isnan	—
isnumeric	—
isnumerictype	—
isreal	—
isrow	—
isscalar	—
issigned	—
isvector	—
le	Las señales con diferentes inclinaciones en punto fijo no son soportadas.
length	—
logical	—
lowerbound	—
lsb	Soporta señales con escala punto fijo.

FUNCION	OBSERVACIONES/LIMITACIONES
	• Soporta señales escalares, vectores y matrices sencillas y dobles.
lt	• Las señales con diferentes inclinaciones en punto fijo no son soportadas.
max	—
min	—
minus	• Cualque entrada de FI no debe es constane, su valor debe ser conocido en tiempo de compilacion para que pueda se emitidos a un objeto FI
mtimes	• Cualque entrada de FI no debe es constane, su valor debe ser conocido en tiempo de compilacion para que pueda se emitidos a un objeto FI
ndims	—
ne	• No soporta señales putno fijo con diferentes inclinaciones.
numberofelements	• numberofelements y numel trabaja tanto para el numel FI de matlab como para objetos incustrados en matlab.
numerictype	• las señales punto fijo que se encuentran dentro del embedded matlab tinene una funcion de bloques desde simulink es asignado al objeto numerictype que tiene referencia con la informacion de los tipos de datos de la señal y su escalizacion. • Retorna los tipos de datos cuando la entrada no es una señal punto fijo. • Use numerictypem para crear objetos con el codigo de Embedded MATLAB.
permute	—
plus	• Cualque entrada de FI no debe es constane, su valor debe ser conocido en tiempo de compilacion para que pueda se emitidos a un objeto FI

FUNCION	OBSERVACIONES/LIMITACIONES
pow2	Para la sintaxis pow2(a, K), K puede ser una constante; que es el valor que se puede conocer en el tiempo de compilación que puede ser emitido a un objeto FI.
range	—
real	—
realmax	—
realmin	—
repmat	—
rescale	—
reshape	—
sign	—
single	—
size	—
sqrt	- Las entradas complejas y [Slope Bias] producen un error de salida. - Las entradas negativas tienen un resultado de rendimiento de 0.
subsasgn	—
subsref	—
sum	—
times	Cualquier entrada de FI no debe ser constante, su valor debe ser conocido en tiempo de compilación para que pueda ser emitido a un objeto FI
transpose	—
tril	—
triu	—
uint8, uint16, uint32	—
uminus	—

FUNCION	OBSERVACIONES/LIMITACIONES
uplus	—
upperbound	—
vertcat	—

A.45 CARACTERISTICAS DE EMBEDDED MATLAB DE PUNTO FIJO

A.45.1 EMBEDDED MATLAB MEX

Embedded MATLAB MEX convierte el código M en una función C-MEX que contiene las optimizaciones de EMBEDDED MATLAB para acelerar automáticamente los algoritmos C de punto fijo compilados a la velocidad del código de matlab.

A.45.2 UTILIZACION DE TIPOS DE DATOS OVERRIDE CON EMBEDDED MATLAB MEX

El toolbox de punto fijo envía con una demostración de cómo generar una función C-MEX desde el código M. El código M toma el promedio ponderado de una señal para crear un filtro pasabaja. Para ejecutar la prueba, haga clic en el filtrado de pasabaja de punto fijo usando el link de embedded matlabMEX y siga las instrucciones en el panel derecho de la ayuda del navegador. Se puede especificar el tipo de datos sobre este demo escribiendo en el prompt de matlab "define fixed-point parameters". Asimismo sobre el tipo de datos, escriba el siguiente comando en el prompt de matlab después de ejecutar el reset (fipref):

```
>>fipref('DataTypeOverride','TrueDoubles')
```

este comando ordena al toolbox de punto fijo para crear un objeto FI con un tipo doble. Cuando se compila el archivo M se usa el comando emlmex en la sección "compile the M-file into a MEX file" y el resultado es una función MEX que usa datos de punto flotante.

A.46 USO DE UN OBJETO FI CON SIMULINK

A.46.1 LECTURA DE LOS DATOS PUNTO FIJO DESDE EL WORKSPACE

Se puede consultar los datos de punto fijo desde el workspace de matlab dentro de un modelo de simulink desde el block del workspace. Para ello, los datos deben estar en formato de estructura con un objeto FI en el campo del valor. En el formato de orden, el block del workspace solo acepta datos reales y datos de doble precisión.

Para leer los datos en FI, el parametro de interpolador de datos del block del workspace no debe ser seleccionado y el parametro del valor de datos despues de la salida debe ser establecido para cualquier otra extrapilacion.

A.46.2 ESCRIBIR DATOS PUNTO FIJO EN EL WORKSPACE

Se puede escribir la salida de punto fijo desde un modelo hacia el workspace, ya sea en conjunto o en formato de estructura. Los datos punto fijo se pueden escribir por un block hacia el workspace en el formato de estructura y puede ser leído dentro el modelo de simulink.

Nota para escribir un dato punto fijo en el workspace de MATLAB como un objeto FI, selección el tamaño del dato punto fijo como un objeto FI, y miro en el cuadro de dialogo del workspace. De otra manera, los datos de punto fijo es convertido para doble y se escribi en el workspace como double.

Por ejemplo, se puede usar el siguiente codigo para crear una estructura en el workspace con los valores de un objeto FI. A continuación se puede utilizar el workspace para llevar los datos a simulink.

```
>>a = fi([sin(0:10)' sin(10:-1:0)'])
a =
    0 -0.5440
    0.8415  0.4121
    0.9093  0.9893
    0.1411  0.6570
   -0.7568 -0.2794
   -0.9589 -0.9589
   -0.2794 -0.7568
    0.6570  0.1411
    0.9893  0.9093
    0.4121  0.8415
   -0.5440     0
  DataTypeMode: Fixed-point: binary point scaling
        Signed: true
      WordLength: 16
  FractionLength: 15
        RoundMode: nearest
      OverflowMode: saturate
      ProductMode: FullPrecision
MaxProductWordLength: 128
        SumMode: FullPrecision
MaxSumWordLength: 128
    CastBeforeSum: true
```

```

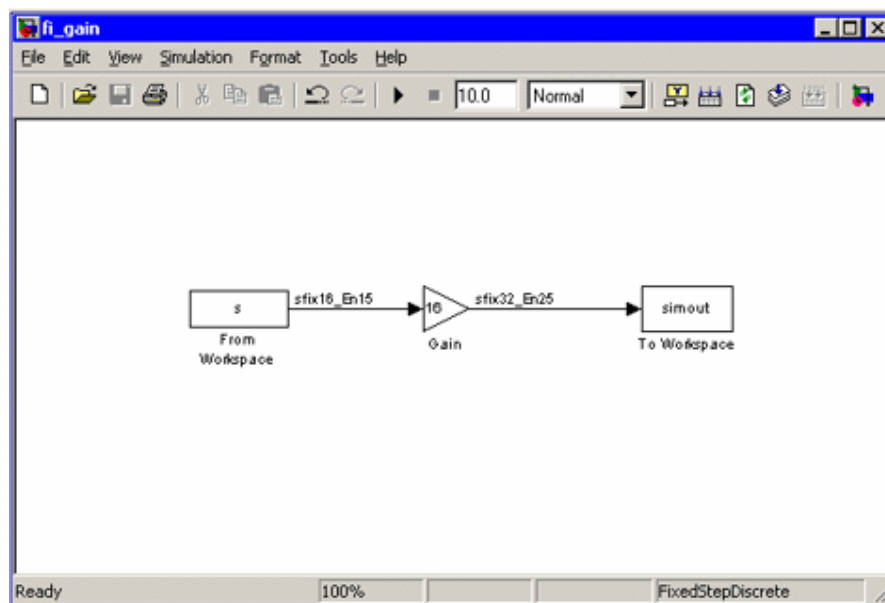
>>s.signals.values = a
s =
    signals: [1x1 struct]
>>s.signals.dimensions = 2
s =
    signals: [1x1 struct]
>>s.time = [0:10]'
s =
    signals: [1x1 struct]
    time: [11x1 double]

```

Desde el modelo del workspace se tiene la estructura FI s en el parametros de datos.

Recuerde, para escribir los datos de punto fijo en el workspace de matlab como un objeto FI, seleccione el Log del dato punto fijo como un objeto FI en el workspace. De los contrario, los datos de punto fijo son convertidos a dobles y escritos en el workspace como doble.

- **Start time** — 0.0
- **Stop time** — 10.0
- **Type** — Fixed-step
- **Solver** — discrete (no continuous states)
- **Fixed step size (fundamental sample time)** — 1.0



para el bloque del workspace escriba el resultado de la simulacion de matlab como una estructura FI:

```

>>simout.signals.values

```

```

ans =
    0 -8.7041
   13.4634   6.5938
   14.5488  15.8296
    2.2578  10.5117
   -12.1089  -4.4707
   -15.3428 -15.3428
    -4.4707 -12.1089
   10.5117   2.2578
   15.8296  14.5488
    6.5938  13.4634
   -8.7041    0
      DataTypeMode: Fixed-point: binary point scaling
        Signed: true
      WordLength: 32
    FractionLength: 25
      RoundMode: nearest
    OverflowMode: saturate
      ProductMode: FullPrecision
MaxProductWordLength: 128
      SumMode: FullPrecision
MaxSumWordLength: 128
    CastBeforeSum: true

```

A.46.3 CONFIGURAR LOS PARAMETROS DE VALOR Y TIPO DE DATOS DEL BLOCK.

Se puede utilizar las expresiones desde el toolbox de punto fijo para especificar los parámetros de valores y de los tipos de datos en simulink.

A.46.4 LOGGING DE SEÑALES PUNTO FIJO

Cuando las señales punto fijo son logged en el workspace de matlab, siempre se registran como objeto FI. Para habilitar la señal logging para una señal, seleccione la opción de datos de señal log en el cuadro de diálogo de señales. Cuando se registran señales como modelo de referencia o modelo gráfico stateflow, la longitud de palabra de los objetos FI pueden ser mas grande de lo esperado. Las longitudes de palabra de punto fijo en las señales de referencia y en los modelos gráficos de stateflow es apuntado como el mayor tamaño de los contenedores de almacenamiento de datos.

A.46.5 ACCESO DE LOS DATOS DE PUNTO FIJO DURANTE LA SIMULACION

Simulink ofrece una aplicación de interfaz (API) que permite el acceso a los datos de programa, como entradas y salidas, parámetros, estados y vectores de trabajo, durante la simulación.

Se puede utilizar esta interfaz para desarrollar programas capaces de bloquear el acceso a los datos mientras se ejecuta la simulación o el acceso a los datos desde la línea de comandos de matlab.

A.47 USO DE SIMULIN EMBEDDED MATLAB COMO BLOQUE DE FUNCION

A.47.1 USO DEL TIPO DE DATOS PUNTO FIJO CON EL BLOQUE DE FUNCION DE EMBEDDED MATLAB

El bloque de la función embedded Matlab permite componer una función en lenguaje de matlab en un modelo de simulink que genera un código incrustado usando el subconjunto de embedded de matlab. Al simular el modelo o generar el código de un objeto. Este código de memoria cumple con los requisitos estrictos y el tipo de datos de los entornos.

- [Embedded MATLAB Function](#) : página de bloque de referencia en la documentación de simulink.
- [Using the Embedded MATLAB Function Block](#) : en la documentación de matlab
- [Working with Embedded MATLAB](#) : en la documentación de Embedded MATLAB

Un significativo número de funciones del toolbox de punto fijo es soportada por el Embedded MATLAB.

Nota para simular los modelos use los tipos de datos punto fijo en simulink.

A.47.2 USO DE LA FUNCION Embedded MATLAB CON UN TIPO DE DATO OVERRIDE

Cuando se usa el embedded MATLAB en simulink se especifica el tipo de datos override, para que el bloque determine el tipo de datos equivalentes de la entrada de señal y los tipos de parámetros que se usarán en la simulación. La siguiente tabla muestra cómo la función embedded MATLAB determina el tipo de dato override equivalente desde el tipo de dato de la entrada de la señal o el parámetro y el tipo de dato override que se establece en el modelo de simulink.

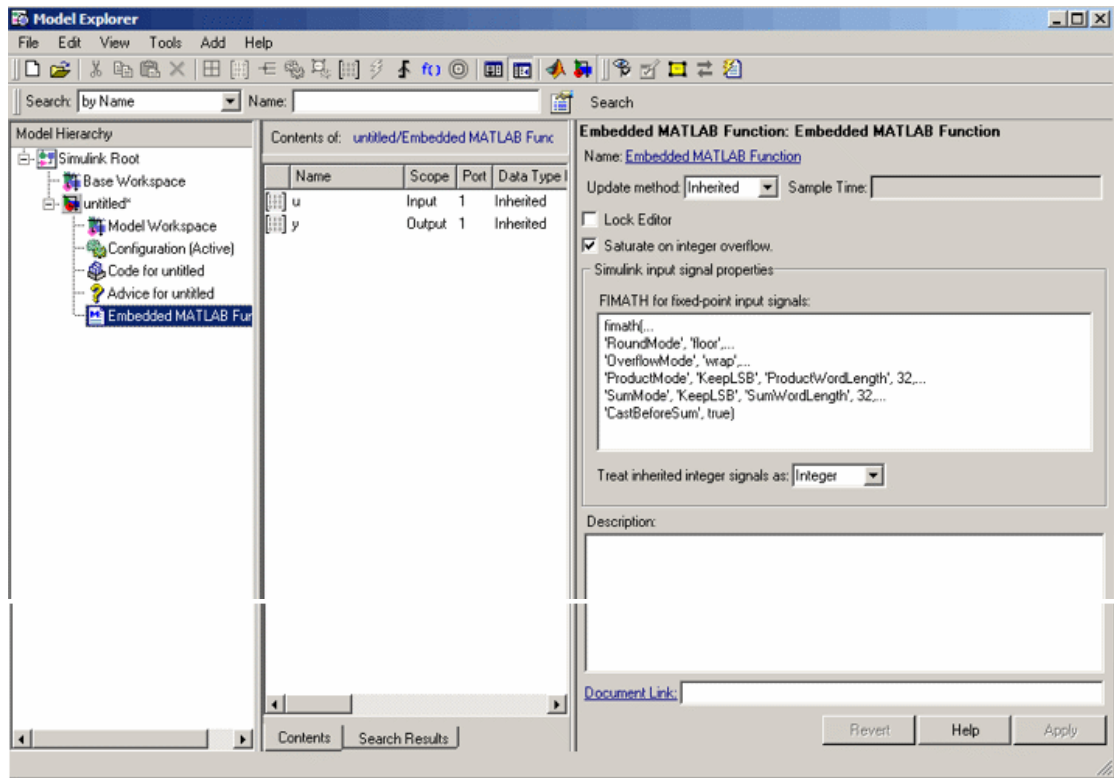
A.47.3 USO DEL MODEL Explorer CON LA FUNCION Embedded MATLAB PUNTO FIJO

Se puede especificar parametros para la funcion embedded MATLAB usando el modelo punto fijo:

1. Coloque la funcion embedded matlab en el nuevo modelo. Este bloque esta situado en las funciones definidas por el usuario en la biblioteca de simulink.
2. Abra el Model Explorer para seleccionar **View > Model Explorer** desde su modelo

ENTRADA DE SEÑAL O TIPO DE PARAMETROS	ESTABLECER TIPOS DE DATOS OVERRIDE	TIPO DE DATOS OVERRIDE EQUIVALENTES
Inherited single	True doubles	fi double
	True singles	fi single
Specified single	True doubles	Built-in double
	True singles	Built-in single
Inherited double	True doubles	fi double
	True singles	fi single
Specified double	True doubles	Built-in double
	True singles	Built-in single
Inherited Fixed	True doubles	fi double
	True singles	fi single
Specified Fixed	True doubles	fi double
	True singles	fi single

3. Expanda el **untitled*** nodo en el panel **Model Hierarchy** del Model Explorer y seleccione la funcion de **Embedded MATLAB**. El Model Explorer aparecera como:



Los parametros de la entrada de simulink muestran propiedades en el panel de embedded matlab, y se aplican a los bloques de funciones que utilizan tipos de datos punto fijo.

Las entradas de señales punto fijo FIMATH definen el objeto fimath para ser asociado con simulink punto fijo o entrada de señales entera. Se puede hacer de las siguientes formas:

- Define completamente el objeto fimath en los valores de parametros usados en el toolbox de punto fijo.
- Entre el nombre de las variables del objeto fimath que se define en el workspace de MATLAB.

A.47.4 EJEMPLO DE LA IMPLEMENTACION DIRECTA DE PUNTO FIJO DESDE UN FIR USANDO LA FUNCION EMEBEDDED MATLAB

Lo siguiente muestra como se crea un filtro pasabaja punto fijo directamente desde un filtro FIR en simulink usando del toolbox de punto fijo y la funcion embedded MATLAB Function, para eso se siguen los siguientes pasos:

I. Programa de la funcion Embedded MATLAB:

1. abra un nuevo modelo del embedded matlab .

2. guarde el modelo como eML_fi.mdl.
 3. de double-click al bloque de la función embedded MATLAB es su modelo para abrir el editor de embedded MATLAB Editor. Copie y pegue el siguiente código de MATLAB incluyendo los comentarios, dentro del editor:
 4. `function [yout,zf] = dffirdemo(b, x, zi)`
 5. `%eML_fi doc model example`
 6. `%Initialize the output signal yout and the final conditions zf`
 7. `Fy = fimath('RoundMode','Floor','OverflowMode','Wrap',...`
 8. `'ProductMode','KeepLSB','ProductWordLength',32,...`
 9. `'SumMode','KeepLSB','SumWordLength',32);`
 10. `Ty = numericity(1,12,8);`
 11. `yout = fi(zeros(size(x)), 'numericity', Ty, 'fimath', Fy);`
 12. `zf = zi;`
 13. `% FIR filter code`
 14. `for k=1:length(x);`
 15. `% Update the states: z = [x(k);z(1:end-1)]`
 16. `zf(:) = [x(k);zf(1:end-1)];`
 17. `% Form the output: y(k) = b*z`
 18. `yout(k) = b*zf;`
 19. `end`
 20. `% Plot the outputs only in simulation.`
 21. `% This does not generate C code.`
 22. `eml.extrinsic('figure');`
 23. `eml.extrinsic('subplot');`
 24. `eml.extrinsic('plot');`
 25. `eml.extrinsic('title');`
 26. `eml.extrinsic('grid');`
 27. `figure;`
 28. `subplot(211);plot(x); title('Noisy Signal');grid;`
`subplot(212);plot(yout); title('Filtered Signal');grid;`
- The Editor should now appear as follows:

```

1 function [yout,zf] = diffirdemo(b, x, zi)
2 %eML_fi doc model example
3 %Initialize the output signal yout and the final conditions zf
4 Fy = fimath('RoundMode','Floor','OverflowMode','Wrap',...
5 'ProductMode','KeepLSB','ProductWordLength',32,...
6 'SumMode','KeepLSB','SumWordLength',32);
7 Ty = numerictype(1,12,8);
8 yout = fi(zeros(size(x)), 'numerictype', Ty, 'fimath', Fy);
9 zf = zi;
10
11 % FIR filter code
12 for k=1:length(x);
13     % Update the states: z = [x(k);z(1:end-1)]
14     zf(:) = [x(k);zf(1:end-1)];
15     % Form the output: y(k) = b*z
16     yout(k) = b*zf;
17 end
18
19 % Plot the outputs only in simulation.
20 % This does not generate C code.
21 eml.extrinsic('figure');
22 eml.extrinsic('subplot');
23 eml.extrinsic('plot');
24 eml.extrinsic('title');
25 eml.extrinsic('grid');
26 figure;
27 subplot(211);plot(x); title('Noisy Signal');grid;
28 subplot(212);plot(yout); title('Filtered Signal');grid;

```

II. PREPARE LAS ENTRADAS

Define los coeficiente b, el ruido x y las condiciones iniciales zi para representar las siguiente lineas de comandos de matlab:

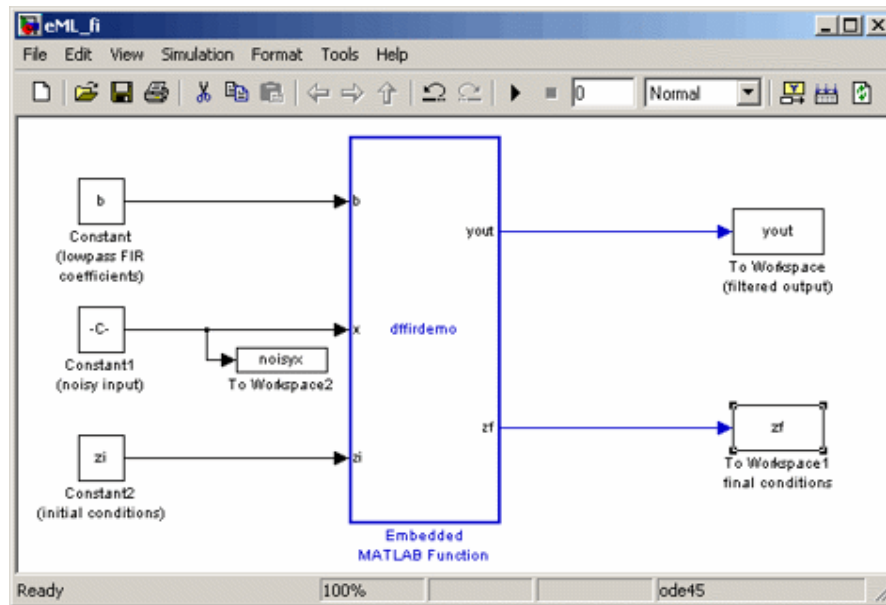
```

>>b = fi_fir_coefficients;
>>load mtlb
>>x = mtlb;
>>n = length(x);
>>noise = sin(2*pi*2140*(0:n-1)/Fs);
>>x = x + noise;
>>zi = zeros(length(b),1);

```

III. CREAR EL MODELO

1. Adiciones el bloque a su modelo creado como se muestra:



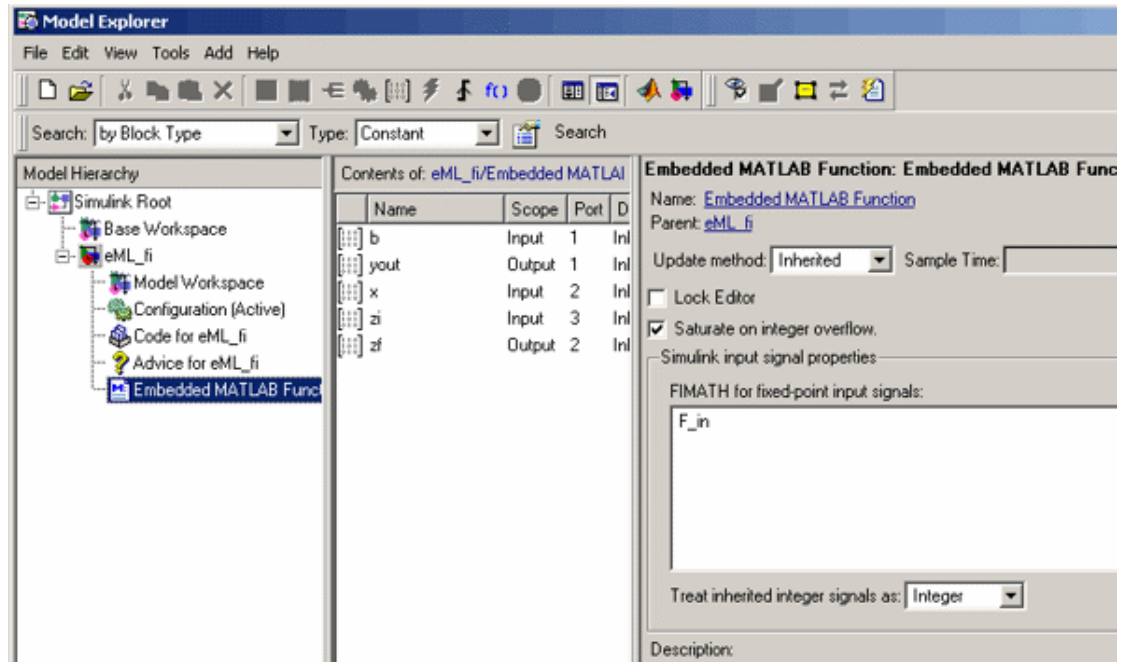
2. establezca los parametros del bloque con los siguientes valores:

Bloque	Parametro	Valor
Constant	Constant value	b
	Interpret vector parameters as 1-D	Unselected
	Sampling mode	Sample based
	Sample time	inf
	Output data type mode	Specify via dialog
	Output data type	sfix(12)
	Output scaling mode	Use specified scaling
	Output scaling value	2 ⁻¹²
Constant1	Constant value	x+noise
	Interpret vector parameters as 1-D	Unselected
	Sampling mode	Sample based
	Sample time	1
	Output data type mode	Specify via dialog
	Output data type	sfix(12)
	Output scaling mode	Use specified scaling

Bloque	Parametro	Valor
	Output scaling value	2 ⁻⁸
Constant2	Constant value	zi
	Interpret vector parameters as 1-D	Unselected
	Sampling mode	Sample based
	Sample time	inf
	Output data type mode	Specify via dialog
	Output data type	sfix(12)
	Output scaling mode	Use specified scaling
	Output scaling value	2 ⁻⁸
To Workspace	Variable name	yout
	Limit data points to last	inf
	Decimation	1
	Sample time	-1
	Save format	Array
	Log fixed-point data as a fi object	Selected
To Workspace1	Variable name	zf
	Limit data points to last	inf
	Decimation	1
	Sample time	—1
	Save format	Array
	Log fixed-point data as a fi object	Selected
To Workspace2	Variable name	noisyx
	Limit data points to last	inf
	Decimation	1
	Sample time	—1
	Save format	Array
	Log fixed-point data as a fi object	Selected

IV. DEFINIR LA ENTRADA FIMATH USANDO EL MODEL EXPLORER

1. Define el objeto `fimath` para ser usado por el embedded MATLAB como entrada en el workspace de matlab. Note que las propiedades del objeto `fimath` son definidas en el código de emdedded MATLAB en orden para alguna operaciones aritmeticas de cuantizacion:
2. `F_in = fimath('RoundMode','Floor','OverflowMode','Wrap',...`
3. `'ProductMode','KeepLSB','ProductWordLength',32,...`
4. `'SumMode','KeepLSB','SumWordLength',32)`
5. `F_in =`
6. `RoundMode: floor`
7. `OverflowMode: wrap`
8. `ProductMode: KeepLSB`
9. `ProductWordLength: 32`
10. `SumMode: KeepLSB`
11. `SumWordLength: 32`
12. `CastBeforeSum: true`
12. abra el Model Explorer para seleccionar el model **View > Model Explorer**.
13. Click en el nodo **Base Workspace** en **Model Hierarchy** en el panel de Model Explorer. Se puede ver el `fimath F_in` esta definido en la lista de constante del panel.
14. Click en el nodo **eML_fi > Embedded MATLAB Function** en **Model Hierarchy**.
15. Entre `F_in` en los parametros **FIMATH for fixed-point input signals** en la funcion Embedded MATLAB en el panel de **Dialog** del Model Explorer y de click **Apply**. Este step establece al objeto `fimath` para las 3 entradas desde el Embedded MATLAB. Ahora el Model Explorer aparece de la siguiente forma:



V. CORRER LA SIMULACION

Corra la simulacion para seleccionar el modelo por **Ctrl+T**. mientras se esta simulando, la información de salida se ve en la linea de comandos de matlab. Se puede ver las graficas de la señal ruidosa y la señal del filtro.

A.48 USE EL OBJETO FI CON BLOQUES DEL PROCESAMIENTO DE SEÑALES

A.48.1 LEER LAS SEÑALES PUNTO FIJO DESDE EL WORKSPACE

Se puede leer datos puntos fijos desde el workspace de matlab dentro de un modelo de simulink usando el [Signal From Workspace](#) y [Triggered Signal From Workspace](#) que son bloques de la procesamiento de señales. Entre el nombre de la variable dedinida fi en los parametros de la señal de [Signal From Workspace](#) o [Triggered Signal From Workspace](#).

A.48.2 ESCRIBIR LAS SEÑALES PUNTO FIJO DESDE EL WORKSAPCE

Las salidas punto fijo dede el model pueden ser escritas desde el workspace de matlaba por medio de [Signal To Workspace](#) o [Triggered To Workspace](#) que son bloques del procesamiento de señales. Los datos punto fijo siempre se escriben como arrays de 2D o 3D.

Nota: para escribir datos punto fijo desde el workspace de matlab como un objeto fi, seleccionamos el **Log fixed-point data as a fi object** .

Por ejemplo se puede usar el siguiente codigo para crear un objeto fi en el workspace de matlab:

```
>>a = fi([sin(0:10)' sin(10:-1:0)'])
a =
    0 -0.5440
    0.8415    0.4121
    0.9093    0.9893
    0.1411    0.6570
   -0.7568   -0.2794
   -0.9589   -0.9589
   -0.2794   -0.7568
    0.6570    0.1411
    0.9893    0.9093
    0.4121    0.8415
   -0.5440     0
    DataTypeMode: Fixed-point: binary point scaling
    Signed: true
    WordLength: 16
    FractionLength: 15
    RoundMode: nearest
    OverflowMode: saturate
    ProductMode: FullPrecision
    MaxProductWordLength: 128
    SumMode: FullPrecision
    MaxSumWordLength: 128
    CastBeforeSum: true
```

La señal desde el workspace en el siguiente modelo tiene las especificaciones:

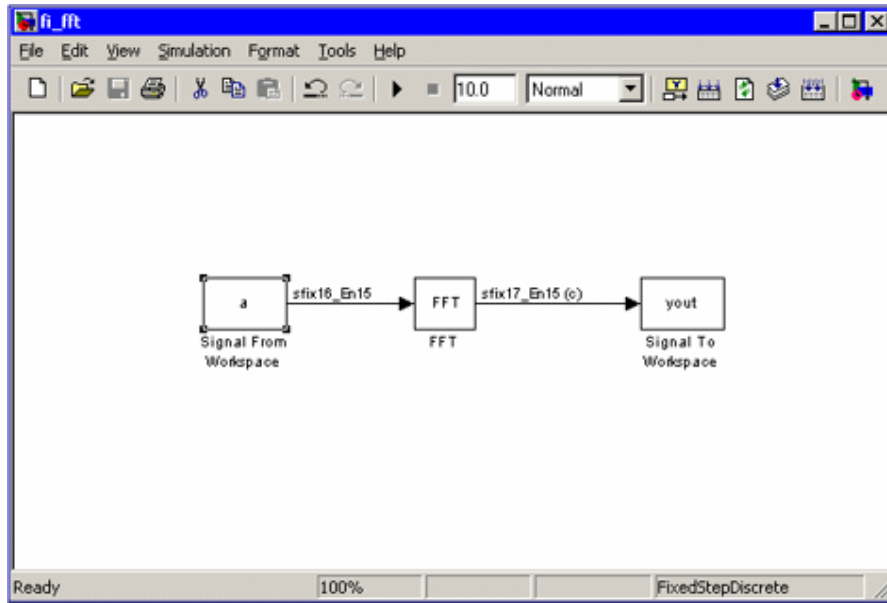
- **Signal** — a
- **Sample time** — 1
- **Samples per frame** — 2
- **Form output after final data value by** — Setting to zero

Los siguientes parametros en el solver del cuadro de dialogo de **Configuration Parameters** tiene las siguientes especificaciones:

- **Start time** — 0.0
- **Stop time** — 10.0
- **Type** — Fixed-step
- **Solver** — discrete (no continuous states)

- **Fixed step size (fundamental sample time) — 1.0**

Recuerde, para escribir los datos punto fijo en el workspace de matlab como objeto fi, se selecciona **Log fixed-point data as a fi object**



el resultado de la simulacion es:

```

yout =
(:,:,1) =
    0.8415   -0.1319
   -0.8415   -0.9561
(:,:,2) =
    1.0504    1.6463
    0.7682    0.3324
(:,:,3) =
   -1.7157   -1.2383
    0.2021    0.6795
(:,:,4) =
    0.3776   -0.6157
   -0.9364   -0.8979
(:,:,5) =
    1.4015    1.7508
    0.5772    0.0678
(:,:,6) =
   -0.5440     0
   -0.5440     0
  
```

DataTypeMode: Fixed-point: binary point scaling
Signed: true

WordLength: 17
FractionLength: 15
RoundMode: nearest
OverflowMode: saturate
ProductMode: FullPrecision
MaxProductWordLength: 128
SumMode: FullPrecision
MaxSumWordLength: 128
CastBeforeSum: true

A.49 USO DEL OBJETO FI CON EL TOOLBOX DE DISEÑO DE FILTROS

Cuando la propiedad arithmetic es establecida como 'fixed', se puede usar un objeto fi existente como la entrada, satado o coeficientes de un objeto dfilt en el toolbox de diseño de filtros. Sin embargo los filtros punto fijo en el toolbox de diseño de filtros retornan objetos fi como salidas.

A.50 PROPIEDADES DEL OBJETO FI

Las propiedades asociadas con los objeto FI son descritos a continuacion en orden alfabetico.

Nota Las propiedades FIMATH y NUMERICTYPE son propiedades del objeto FI.

A.50.1 bin

valor entero almacenado en binario de un objeto FI

A.50.2 data

Valor numerico de la palabra real de unobjeti FI.

A.50.3 dec

valor entero almacenado en decimal de un objeto FI.

A.50.4 double

valor de la palabra real de un objeto fi almacenado como doble en MATLAB®

A-50.5 fimath

objeto fimath asociado con el objeto fi.el objeto fimath por defecto es:

RoundMode: nearest
OverflowMode: saturate
ProductMode: FullPrecision

MaxProductWordLength: 128
SumMode: FullPrecision
MaxSumWordLength: 128
CastBeforeSum: true

A.50.6 hex

valor entero almacenado en hexagecimal de un objeto fi.

A.50.7 int

valor entero almacenado de un objeto fi, almacenado en un tipo de dato incorporado en matlab. Se puede usar int8, int16, int32, uint8, uint16, y uint32.

A.50.8 NumericType

Estructura que contiene todos los tipos de datos y los atributos de escalizacion de un objeto fi. El objeto numericType tiene las mismas estructuras de matlab, con la excepcion de que solo le permite establecer valores validos de los campos definidos. La siguiente tabla muestra las posibles características en que trabajan las estructuras validas para un objeto fi.

Modo del tipo de datos	Tipo de datos	Escalizacion	Signo	Longitud de palabra	Longitud de fraccion	Pendiente	Bias
<i>Fully specified fixed-point data types</i>							
Fixed-point: binary point scaling	Fixed	BinaryPoint	1/0 true/ false	positive integer from 1 to 65,536	positive or negative integer	1	0
Fixed-point: slope and bias scaling	Fixed	SlopeBias	1/0 true/ false	positive integer from 1 to 65,536	N/A	any floating-point number	any floating-point number
<i>Partially specified fixed-point data type</i>							
Fixed-point: unspecified scaling	Fixed	Unspecified	1/0 true/ false	positive integer from 1 to 65,536	N/A	N/A	N/A
<i>Fully specified scaled double data types</i>							
Scaled double: binary point scaling	ScaledDouble	BinaryPoint	1/0 true/ false	positive integer from 1 to 65,536	positive or negative integer	1	0

Modo del tipo de datos	Tipo de datos	Escalizacion	Signo	Longitud de palabra	Longitud de fraccion	Pendiente	Bias
Scaled double: slope and bias scaling	ScaledDouble	SlopeBias	1/0 true/ false	positive integer from 1 to 65,536	N/A	any floating-point number	any floating-point number
<i>Partially specified scaled double data type</i>							
Scaled double: unspecified scaling	ScaledDouble	Unspecified	1/0 true/ false	positive integer from 1 to 65,536	N/A	N/A	N/A
<i>Built-in data types</i>							
double	double	N/A	1 true	64	0	1	0
single	single	N/A	1 true	32	0	1	0
boolean	boolean	N/A	0 false	1	0	1	0

A.50.9 oct

valor entero almacenado en octal de un objeto fi.

A.51 PROPIEDADES DEL OBJETO FIMATH

Las propiedades asociadas con el objeto fimath se muestran a continuacion en orden alfabetico:

A.51.1 CastBeforeSum

Ambos operandos son emitidos a la suma de tipo de datos antes de la adición. Los posibles valores de esta propiedad son 1 (emitidos antes de suma) y 0 (no emitidos antes de la suma). El valor por defecto de esta propiedad es 1 (verdadero)

A.51.2 MaxProductWordLength

Maxima longitud de palabra permitida para el tipo de datos del producto. El valor por defecto de esta propiedad es 128.

A.51.3 MaxSumWordLength

Maxima longitud de palabra permitida para el tipo de datos de la suma. El valor por defecto de esta propiedad es 128.

A.51.4 OverflowMode

Modo de manipulacion del overflow. El valor de la propiedad overflow puede ser uno de los siguientes:

- saturate :satura al valor maximo o minimo del rango de overflow de punto fijo.
- wrap :ajusta el overflow. Este modo tambien se conoce como overflow de complemento a dos.

El valor por defecto de esta propiedad es saturate.

A.51.5 ProductBias

Inclinacion del producto de tipo de datos. Este valor puede ser cualquier numero punto flotante. El tipo de datos del producto define al tipo de datos del resultado de una multiplicación de dos objetos fi. El valor por defecto de esta propiedad es 0.

A.51.6 ProductFixedExponent

Es el tipo de datos de exponente fijo de un producto. Este valor puede ser cualquier numero entero positivo o negativo. El tipo de datos del producto define al tipo de datos del resultado de una multiplicación de dos objetos fi.

$$ProductSlope = ProductSlopeAdjustmentFactor \times 2^{ProductFixedExponent}$$

el cambio de una de esta propiedad cambia automaticamente las otras propiedades. El productfixedexponent es la versión negativa del productfractionlength. El valor por defecto de esta propiedad es -30.

A.51.7 ProductFractionLength

Longitud de fraccion en bits del tipo de datos del producto. Este valor puede ser cualquier numero entero positivo o negativo. El tipo de datos del producto define al tipo de datos del resultado de una multiplicación de dos objetos fi. El productfractionlength es la versión negativa del productfixedexponent. El valor por defecto de esta propiedad es 30.

A.51.8 ProductMode

Define la forma en que el tipo de datos del producto esta determinado. En las siguientes descripciones, los operandos a y b son reales la longitud de palabra y la longitud de fraccion estan dadas en pares [Wa,Fa] y[Wb,Fb] respectivamente. Wp es la longitud de palabra del tipo de dato del producto y Fp es la longitud de fraccion del tipo de datos del producto.

- FullPrecision : la precision completa de los resultados se mantienen. Se genera un error si la longitud de palabra es mayor que el maxproductwordlength.

$$W_p = W_a + W_b$$

$$F_p = F_a + F_b$$

☐ KeepLSB : mantiene los bits menos significativos. Especifica el tipo de datos de la longitud de palabra del producto, mientras que la longitud de fraccion se

establece para mantener el bits menos significativo del producto. En este modo se mantiene la fullprecision y es posible el overflow.

$W_p = \text{specified in the ProductWordLength property}$

$$F_p = F_a + F_b$$

□ KeepMSB mantiene los bits mas significativos. Especifica el tipo de datos de la longitud de palabra del producto, mientras que la longitud de fraccion se establece para mantener el bit mas significativo del producto. En este modo se impide el overflow, y se puede perder precision.

$W_p = \text{specified in the ProductWordLength property}$

$$F_p = W_p - \text{integer length}$$

donde

$$\text{integer length} = (W_a + W_b) - (F_a + F_b)$$

□ SpecifyPrecision se especifica la longitud de palabra y la longitud de fraccion del tipo de datos del producto.

$W_p = \text{specified in the ProductWordLength property}$

$F_p = \text{specified in the ProductFractionLength Property}$

para [pendiente inclinada] se especifica la pendiente y la inclinacion del tipo de datos del producto.

$S_p = \text{specified in the ProductSlope property}$

$B_p = \text{specified in the ProductBias property}$

- [Slope Bias] : esta matematica esta solo definida para los productos cuando el producmode esta estabecido en specifyprecision.el valor por defecto de esta propiedad es FullPrecision.

A.51.9 ProductSlope

Pendiente de los tipos de datos del producto. Este valor puede ser cualquier numero punto flotante. El tipo de dato del producto define el tipo de datos del resultado de una multiplicaron de dos objetos fi.

$$ProductSlope = ProductSlopeAdjustmentFactor \times 2^{ProductFixedExponent}$$

el valor por defecto de esta propiedad es 9.3132e-010.

A.51.10 ProductSlopeAdjustmentFactor

Factor de ajuste de la pendiente del tipo de datos de un producto. Este valor puede se cualquier punto flotante superior o igual a 1 y menor de 2. El tipo de dato del producto define el tipo de datos del resultado de una multiplicaron de dos objetos fi.

$$ProductSlope = ProductSlopeAdjustmentFactor \times 2^{ProductFixedExponent}$$

el valor por defecto de esta propiedad es 1.

A.51.11 ProductWordLength

Longitud de palabra en bits del tipo de datos del producto. Este valor debe ser un entero positivo. El tipo de datos del producto define el tipo de datos del resultado de una multiplicación de objetos fi. El valor por defecto de esta propiedad es 32.

A.51.12 RoundMode

Modo de redondeo. El valor de la propiedad roundmode puede ser uno de las siguientes características:

- ceil.
- convergent.
- Fix.
- Floor.
- Nearest.
- round .

el valor por defecto de esta propiedad es nearest.

A.51.13 SumBias

Inclinación de la suma de tipos de datos. Este valor puede ser cualquier número punto flotante. La suma se define el tipo de datos de tipo de datos del resultado de una suma de dos objetos fi. El valor por defecto de esta propiedad es 0.

A.51.14 SumFixedExponent

El exponente fijo de la suma del tipo de datos . este valor puede ser cualquier entero positivo o negativo. El tipo de datos de la suma está definida por el tipo de datos del resultado de la suma de dos objetos fi.

$$SumSlope = SumSlopeAdjustmentFactor \times 2^{SumFixedExponent}$$

el SumFixedExponent es el negativo de la SumFractionLength. El valor por defecto de esta propiedad es -30.

A.51.15 SumFractionLength

Longitud de fracción en bits del tipo de datos de la suma. Este valor puede ser cualquier número entero positivo o negativo. El tipo de datos de la suma define el tipo de datos del resultado de la suma de dos objetos fi. el SumFractionLength es el negativo de la SumFixedExponent. El valor por defecto de esta propiedad es 30.

A.51.16 SumMode

Define como el tipo de datos de la suma está determinado. En la siguiente descripción, los operandos a y b son reales, con pares de longitud de palabra y de longitud de fracción [Wa, Fa] y [Wb, Fb] respectivamente. Ws es la longitud de palabra del tipo de datos de la suma y Fs es la longitud de fracción del tipo de datos de la suma.

Nota en el caso de que hay dos operando como en a+b, el numberofsumands es de 2, y el $\text{ceil}(\log_2(\text{numebrofsumands}))=1$. en resumen A es una matriz, el numberofsumands es de tamaño (A,1). En

sum(A)

donde

A es un vector, el numberofsumands es de longitud A.

- FullPrecision : la full precision de los resultados se mantienen. Se genera un error si la longitud de palabra es mayor que el maxsumwordlength.

$$W_s = \text{integer length} + F_s$$

donde

$$\text{integer length} = \max(W_a - F_a, W_b - F_b) + \text{ceil}(\log 2(\text{NumberOfSummands}))$$

$$F_s = \max(F_a, F_b)$$

- KeepLSB : mantiene los bits menos significativos. Especificar la longitud de palabra de la suma del tipo de datos, la longitud de fraccion se fija para mantener el menor nivel de bits de suma. En este modo se mantiene la full precision, pero es posible el overflow.

$$W_s = \text{specified in the SumWordLength property}$$

$$F_s = \max(F_a, F_b)$$

- KeepMSB : mantiene los bits mas significativos. Especificar la longitud de palabra de la suma del tipo de datos, la longitud de fraccion se fija para mantener el mayor nivel de bits de suma. En este modo pierde precision, pero impide el overflow.

$$W_s = \text{specified in the SumWordLength property}$$

$$F_s = W_s - \text{integer length}$$

donde

$$\text{integer length} = \max(W_a - F_a, W_b - F_b) + \text{ceil}(\log 2(\text{NumberOfSummands}))$$

- SpecifyPrecision especifica la longitud de palabra y la longitud de fraccion del tipo de datos de la suma.

$$W_s = \text{specified in the SumWordLength property}$$

$$F_s = \text{specified in the SumFractionLength property}$$

para [pendiente inclinada] se especifica la pendiente y la inclinacion del tipo de datos del producto.

$$S_s = \text{specified in the SumSlope property}$$

$$B_s = \text{specified in the SumBias property}$$

[Slope Bias] : esta matematica esta solo definida para el summode cuando esta establecido en specifyprecision. El valor por defecto de esta propiedad es FullPrecision.

A.51.17 SumSlope

Pendiente del tipo de datos de la suma este valor puede ser cualquier numero de punto flotante. El tipo de datos de la suma se define como el tipo de datos del resultado de la suma de dos objetos fi.

$$\text{SumSlope} = \text{SumSlopeAdjustmentFactor} \times 2^{\text{SumFixedExponent}}$$

el valor por defecto de esta propiedad es 9.3132e-010.

A.51.18 SumSlopeAdjustmentFactor

Factor de ajuste de la pendiente de los tipos de datos de suma. Este valor puede ser cualquier número punto flotante superior o igual a 1 y menor que 2. La suma se define como el tipo de datos del resultado de la suma de dos objetos fi.

$$SumSlope = SumSlopeAdjustmentFactor \times 2^{SumFixedExponent}$$

El valor por defecto de esta propiedad es 1.

A.51.19 SumWordLength

Longitud de palabra en bits del tipo de datos de la suma. Este valor debe ser un entero positivo. La suma se define como el resultado de dos objetos fi. El valor por defecto de esta propiedad es de 32.

A.52 PROPIEDADES DEL OBJETO FIPREF

Las propiedades asociadas con el objeto fipref se describe a continuación y por orden alfabético.

A.52.1 DataTypeOverride

Opción del tipo de datos override de un objeto fi.

- ForceOff : No tipo de datos override
- ScaledDoubles : Override con escala dobles
- TrueDoubles : Override con dobles
- True Singles : Override simple

El tipo de datos override solo ocurre cuando el objeto fi es llamado por la función constructora. El valor por defecto de esta propiedad es ForceOff.

A.52.2 FimathDisplay

Opciones de muestra del objeto fimath atribuido al objeto fi

- full : muestra todos los atributos fimath de un objeto fi
- none : ningún atributo fimath es mostrado

El valor por defecto de la propiedad es full.

A.52.3 LoggingMode

Opción logging operaciones establecidas del objeto fi.

- Off : No logging
- on : Información de logged para futuras operaciones.

Cuando LoggingMode está en on, se puede usar las siguientes funciones para retornar la información de logged y crear operaciones para el comando de línea de MATLAB:

- maxlog : Retorna el valor máximo de la palabra real

- `minlog` — Retorna el valor minimos
 - `noverflows` — Retorna el numero de overflows
 - `nunderflows` — Retorna el nuemro de underflows
- el valor por defecto de esta propiedad es de off.

A.52.4 NumericTypeDisplay

Opciones de muestra de atributo `numerictype` de los objetos `fi`

- `full` : muestra todos los atributos `numerictype` de un objeto punto fijo.
- `none` : ningun atributo `numerictype` mostrado.
- `short` : muestra notaciones abreviadas el tipo de datos punto fijo y de escalizacion de un objeto punto fijo en el formato `xWL,FL` donde:
 - o `x` es `s` para signo y `u` para no signo.
 - o `WL` es la longitud de palabra.
 - o `FL` es la longitud de fraccion.

El valor por defecto de esta propiedad es `full`.

A.52.5 NumberDisplay

Opciones para mostrar el valor de un objeto `fi`.

- `bin` :muestra los valores enteros almacenados en formato binario.
- `dec` :muestra los valores enteros almacenados en formato decimal con signo.
- `RealWorldValue` : muestra los valores enteros almacenados en el formato especificaad por la funcion de formato de MATLAB.
- `hex` : muestra los valores enteros almacenados en formato hexadecimal.
- `int` :muestra los valores enteros almacenados en formato decimal con signo.
- `none` :Ningun valro es mostrado.

El valor por defecto de esta propiedad es `RealWorldValue`. En este modo, el valor de un objeto `fi` es mostrado en el formato especifico por la funcion `format` de MATLAB : `+`, `bank`, `compact`, `hex`, `long`, `long e`, `long g`, `loose`, `rat`, `short`, `short e`, O `short g`. Los objetos `fi` en `rat` se muestran de acuerdo a:

$$\frac{1}{(2^{\text{fixed-point exponent}})} \times \text{stored integer}$$

A.53 PROPIEDADES DEL OBBJUETO NUMEREICTYPE

Las propiedades asociadas al objeto `numerictype` se describen a continuacion en orden alfabetico.

A.53.1 Bias

Inclinacion asociada con el objeto `FI`.la inclinación es parte de la representación numerica usada para interpretar un numero punto fijo. Usada con la pendiente, las formas de la inclinacion escaliza al numero. Los numeroa punto fijopueden ser represntados como

$$real - world\ value = (slope \times stored\ integer) + bias$$

donde la pendiente puede ser expresada como

$$slope = fractional\ slope \times 2^{fixed\ exponent}$$

A.53.2 DataType

Tipo de dato asociado con el objeto fi. Los posibles valores de esta propiedad son:

- boolean
- double
- Fixed
- ScaledDouble
- single

el valor por defecto de esta propiedad es fixed.

A.53.3 DataTypeMode

Tipo de datos y escalizaicon asociados con un objeto fi. Los posibles valores de esta propiedad son:

- boolean
- double
- Fixed-point (binary point scaling)
- Fixed-point (slope and bias scaling)
- Fixed-point (unspecified scaling)
- Scaled double (binary point scaling)
- Scaled double (slope and bias scaling)
- Scaled double (unspecified scaling)
- single

el valor por defecto de esta propiedad es Fixed-point (binary point scaling).

A.53.4 FixedExponent

Punto fijo esponente asociado con un objeto fi. El exponente es parte de la representación numerica usda para expresar numeros punto fijo. Los numeros punto fijo pueden ser `representados como:

$$real - world\ value = (slope \times stored\ integer) + bias$$

donde la pendiente puede se expresada como:

$$slope = fractional\ slope \times 2^{fixed\ exponent}$$

el exponente de un numeor punto fijo es igual a la longitud de fraccion negativa:

$$fixed\ exponent = -fraction\ length$$

A.53.5 FractionLength

El valor de la propiedad FractionLength es la longitud de fraccion almacenada como un valor entero del objeto fi en bits.la longitud de fraccion puede ser cualquier valor entero. Si no se especifica la longitud de fraccion de un objeto fi, este se establece como el mejor valor de precision osible. Esta propiedad es automáticamente establecida por el valor de defecto como el mejor valor de precision basado en el valor de la longitud de palabra.

A.53.6 Scaling

Modo de escalización de punto fijo de un objeto fi. Los posibles valores de esta propiedad son:

- BinaryPoint : escalización para un objeto fi definido por la longitud de fracción.
- SlopeBias : escalización para un objeto fi definido por la pendiente y la inclinación.
- Unspecified : ajuste temporal que solo permita la creación de un objeto fi, con el fin de permitir la asignación de un punto binario de mejor escala de precisión.
- Integer : el objeto fi es un entero; the binary point is understood to be at the far right of the word, making the fraction length zero.

El valor por defecto de esta propiedad es BinaryPoint.

A.53.7 Signed

Determina si el objeto tiene signo o no. Los posibles valores de esta propiedad es :

- 1 : signo
- 0 : sin signo
- true : signo
- false : sin signo

El valor por defecto de esta propiedad es true .

A.53.8 Slope

Pendiente asociada con un objeto fi. La pendiente es una parte de la representación numérica usada para expresar un número punto fijo.

Los números punto fijo pueden ser representados como:

$$\text{real-world value} = (\text{slope} \times \text{stored integer}) + \text{bias}$$

donde la pendiente puede ser representado como:

$$\text{slope} = \text{fractional slope} \times 2^{\text{fixed exponent}}$$

A.53.9 SlopeAdjustmentFactor

El factor de pendiente asociado con un objeto fi. el factor de ajuste de la pendiente es equivalente a la pendiente fraccional de un número punto fijo. La pendiente fraccional es parte de la representación numérica usada para expresar un número punto fijo.

Los números punto fijo es representado como:

$$\text{real-world value} = (\text{slope} \times \text{stored integer}) + \text{bias}$$

donde la pendiente puede ser expresada como:

$$\text{slope} = \text{fractional slope} \times 2^{\text{fixed exponent}}$$

A.53.10 WordLength

Valor de la propiedad de longitud de palabra del valor entero almacenado de un objeto punto fijo. La longitud de palabra pueden ser valores enteros positivos. El valor por defecto de esta propiedad es de 16.

A.54 PROPIEDADES DEL OBJETO DE CUANTIZACION

Las propiedades asociadas al objeto quantizer se describen a continuacion en orden alfabetico.

A.54.1 DataMode

Tipo aritmetico usado en la cuantizacion. Esta propiedad puede tener los siguientes valores:

- fixed
- float
- double
- single
- ufixed

el valor por defecto de esta propiedad es fixed.

A.54.2 Format

Formato de dato de un objeto quantizer. La interpretacion de esta propiedad depende del valor de los valores dados a la propiedad DataMode. Por ejemplo, si se especifica la propiedad DataMode con aritmetica punto fijo o punto flotante afecta las interpretaciones del formato de dato de esta propiedad.

A.54.3 OverflowMode

Modo de Overflow. El valor de la propiedad OverflowMode puede tener unos de estos valores:

- saturate.
- wrap.

El valor por defecto de esta propiedad es saturate.

Nota los numeros punto flotante son extendidos por el rango dinamico del overflow para $\pm\text{inf}$.

A.54.4 RoundMode

Modo de redondeo: el valor de la propiedad RoundMode puede tener uno de esos valores:

- ceil
- convergent
- fix
- floor
- nearest

el valor por defecto de esta propiedad es floor.

ANEXO B

“MANUAL TOOLBOX DE DISEÑO DE FILTROS”

B.1 ¿QUÉ ES EL TOOLBOX DE PUNTO FIJO?

B.1.1 Descripción del toolbox de diseño de filtros

El toolbox de diseño de filtros es un conjunto de herramientas que proporciona técnicas avanzadas para el diseño, simulación y análisis de filtros digitales.

Se extiende las capacidades para el pensamiento de señales del Toolbox del filtro y métodos de diseño de arquitectura complejas en las aplicaciones de tiempo real en el DSP, incluyendo el filtrado de adaptación, el filtrado de multirate y el filtro de transformación.

B.1.2 Uso de otros productos de mathworks

Usando el Toolbox de punto fijo y el Toolbox de diseño de filtros, proporciona funciones que simplifican el diseño de filtros de punto fijo y análisis de los aspectos de cuantización. Cuando se usa el diseño de filtros permite generar código verilog y VHDL para filtros de punto fijo.

El Toolbox de diseño de filtros también ofrece la capacidad de diseño de filtros avanzados para simulink y el procesamiento de señales Blackset.

B.1.3 Características claves

- Método avanzado de diseño de filtros FIR, incluido el orden mínimo, la forma de stopband, halfband, la complejidad optimizada polietapico, farrow e interpolados FIR.
- Método avanzado de diseño de filtros LLR, incluida la magnitud arbitraria, ecualizadores de demora, ecualizadores perimétricos octava, halfband, fase cuasi lineal, filtros de peine.
- Método de diseño de filtros multirate, incluida el integrados cada camb (CIC), compensador, CIC, polifásicos IIR y FIR filtros Nyquist -multi estados.
- Soporta la implementación eficiente de un filtro LLR, incluyendo secciones de segundo orden, y andas de filtros digitales.
- Diseño de filtros adoptivos, análisis y aplicaciones, incluidos los basados en LMS, basados en RLS, basados en lattice, frecuencia = dominio transversal constante y proyección afín.

B.2 DISEÑO DE UN FILTRO EN DOS PASOS

B.2.1 COMO TRABAJAR EL TOOLBOX DE DISEÑO DE FILTROS

La característica del toolbox de diseño de filtros es que pone en énfasis las especificaciones del filtro del proceso de diseño del mismo.

Este enfoque es centrado en las especificaciones de las elecciones de los algoritmos de filtros especiales y en el rendimiento durante el diseño del filtro.

Por ejemplo, se puede tener un determinado conjunto de parámetros para el diseño del filtro como una frecuencia stopband, una frecuencia de paso, una atenuación stopband, que utilizan una especificación de parámetros para el diseño del filtro.

A continuación se puede aplicar el filtro utilizando dichas especificaciones, donde también es posible comparar distintos algoritmos que se aplican a un grupo de especificaciones

Hay dos objetos que participan en el diseño del filtro:

- Specification Object: captura los parámetros de diseño de un filtro.
- Implementation Object: describe el filtro diseñado, incluye la matriz de coeficientes y estructuras del filtro.

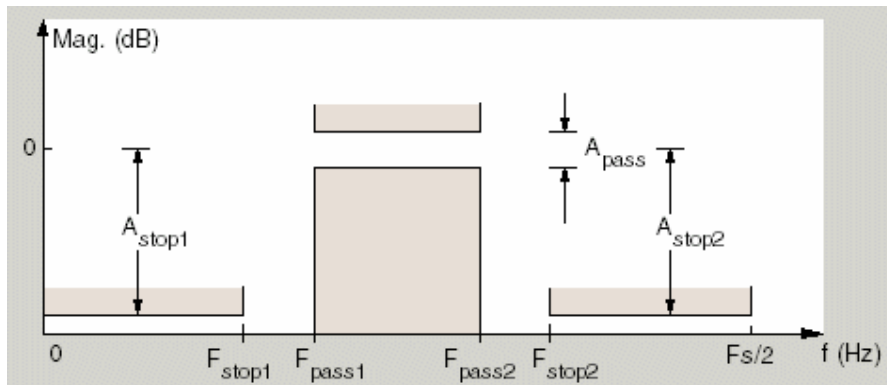
ESPECIFICACIONES DEL OBJETO	IMPLEMENTACION DEL OBJETO
Nivel bajo de especificaciones	Coefficiente del filtro
Propiedades algorítmicas	Estructura del filtro

B.2.2 PROCESO BÁSICO DEL DISEÑO DEL FILTRO:

Se utilizan los siguientes dos pasos para el diseño de un filtro sencillo.

- Crear un filtro con especificaciones
- Diseñar el filtro

Ejemplo: se quiere diseñar un filtro pasabanda en dos pasos. Normalmente un filtro pasabanda se define como se muestra en la siguiente figura:



En este ejemplo una frecuencia de muestreo $F_s = 48\text{KHz}$ es usado se el filtro pasabanda tiene las siguientes especificaciones, el código de matlab especifica que se usa es:

```
>> A_stop 1=60; % atenuación de l primera stupband= 60db
>>F_stop 1= 8400; % nivel del stupband =8400Hz
>> F_pass 1=10800; % nivel de pasa banda = 10800 Hz
>>F_pass 2= 15600; % nivel de cierre de pasa banda= 15600 Hz
>> F_stop 2=1800; % nivel del segundo stop band =1800 Hz
>>A_stop 2=60; % atenuación de la segunda stopploand=60Hz
>>A_pass =1 % cantidad de la onda permitida en el paso=1db
```

En las dos etapas siguientes estas especificaciones al método `fdesing bandpass` como parámetros.

Paso1. para crear un filtro con especificaciones, se anula el código en el prompt de matlab. Pasar el filtro de sus especificaciones como especificaciones directas (`fst1`, `fst2`, `fp2`, `fp1`, `ast1`, `ap`, `ast2`). En este ejemplo se agrega `fs` como ayudamento de última entrada para especificar la frecuencia de muestreo de 48KHz .

```
>>Bandpassspecobj:...
Fdesing bandpass (Fstp1,Fp1,Fp2, Fstp2,Ast1,Ap,Ast2"
F_Stop1, F_Pass1, F_Pass2, F_Stop2, A_Stop1, A_Pass
A_Stop2, 4800)
>>Bandpassspereobj=
Response= "object"
Specification= "Fstp1, Fp1, Fstp2, Fp2, Ast1, Ast2, Ap"
Description = (7x1cell)
Normalized frequency=false
Fs=4800
Fstop1=84000
Fpass2=15600
Fstop2=18000
```

Astop1=60
A pass=1
Astop2=60

Nota: el orden de los filtros no se ha especificado, lo que permite un grado de libertad para el diseño del algoritmo afín de lograr las especificaciones. El diseño puede ser un diseño de orden mínimo.

Las especificaciones de los parámetros como Fstop1, expresan los valores por defecto cuando no se proporcionan. Puede cambiar los valores de los parámetros a un siendo creado el filtro.

Por ejemplo, si hay dos valores que hay que cambiar, F pass y Fstop2, utiliza el conjunto de comando que toma el objeto en primer lugar y luego el valor del parámetro. Evaluando el siguiente código en el prompt de matlab:

```
>>set (Bnadpass Specobj, Fpass2, 15800, "Fstop2, 18400)
```

BandPassSpecObj es el Nuevo filtro de especificaciones que contiene todos los parametros de diseño, incluyendo el tipo de filtro

Paso 2: diseño del filtro utilizando el commando de diseño. Puede acceder a los metodos de diseño disponible, llamando a la funcion designmethods. Por ejemplo, en este caso, puede ejecutar el comando:

```
>> designmethods (BandPassSpecObj)
Design methods for class
Fdesign bandpass (fst1, Fp1, Fp2, Fst2, Ast1, Ap, Ast2)
Butter
Cheby1
Cheby2
Ellip
Equiripple
Kaiserwin
```

Después de la elección de un método de diseño a usar, se puede evaluar el siguiente prompt en Matlab:

```
>>Bandpass Filt=
Filter structure= "Direct from FIR"
Arithmetic="double"
Numerator= (1x44double)
Persistent Memory= false
```

Nota: si no se especifica un método de diseño, un método por defecto se utilizará por ejemplo.

```
>>BandpassFilt= design (Bandpass Specobj)
```

```
BandpassFilt =
```

```
Filter structure = "direct-form fiz"
```

```
Arithmetic= "double"
```

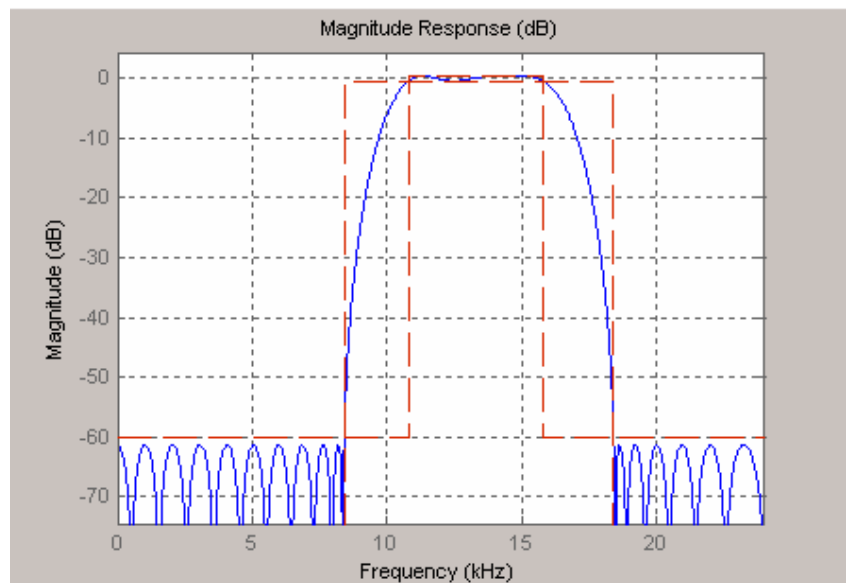
```
Numerator="1x44 double"
```

```
Persistent Memory= false
```

Y el método de diseño será seleccionado automáticamente.

Para revisar el trabajo se puede dibujar la respuesta en magnitud del filtro usando el Tool Visualization Filter. Comprobando que todos los parámetros de diseño se cumplan:

```
>>fvtool (band pass Filt) % dibuja la respuesta en del filtro.
```



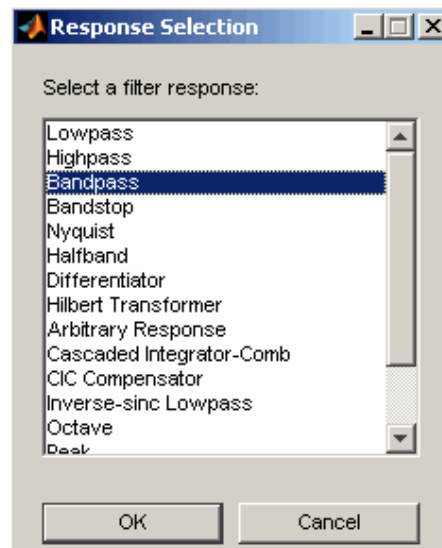
B.2.3 USO DEL FILTERBUILDER EN EL DISEÑO DE UN FILTRO

Filterbuilder presenta la opción de diseñar un filtro usando un GUI cuadro de dialogo en contraposición a la línea de comandos de instalación.

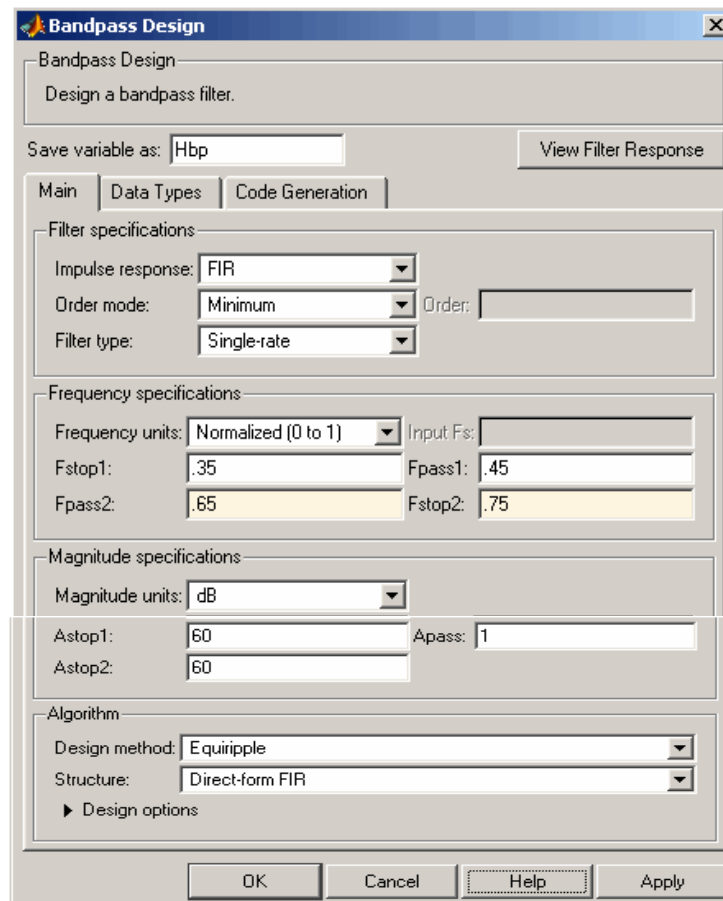
Ejemplo, uso de Filterbuilder en el diseño de un filtro simple para diseñar el filtro cuando Filterbuilder; al escribir el siguiente prompt en Matlab:

```
>>Filterbuilder
```

Esto abre un diagrama de bloque:



Se selecciona la respuesta de filtro bands pass de la lista en el cuadro de dialogo y pulse el botón OK abre la siguiente ventana.



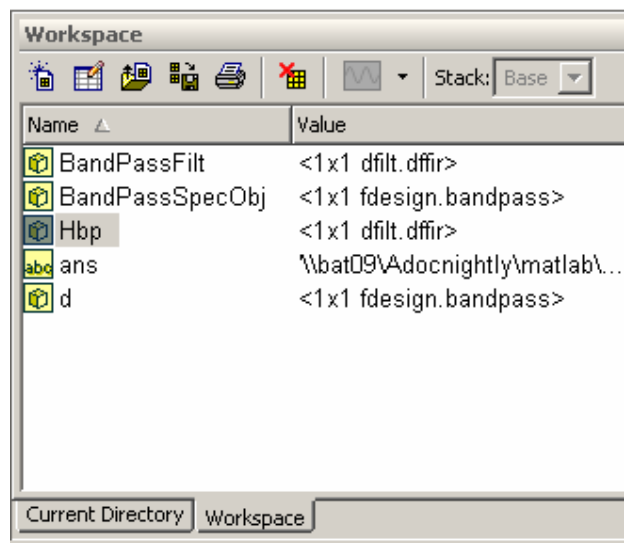
Introduzca las frecuencias correctas para Fpass2 y Fstop2, como se muestra en el grafico anterior haga clic en ACEPTAR.

La especificación usa frecuencias normalizadas de modo que el nivel stopband es expresado en fracciones de nyquist. En este caso 4812 Kh_z.

En el prompt de Mantlab aparece el siguiente mensaje

```
>>The variable "Hbp" has been export to be command window.
```

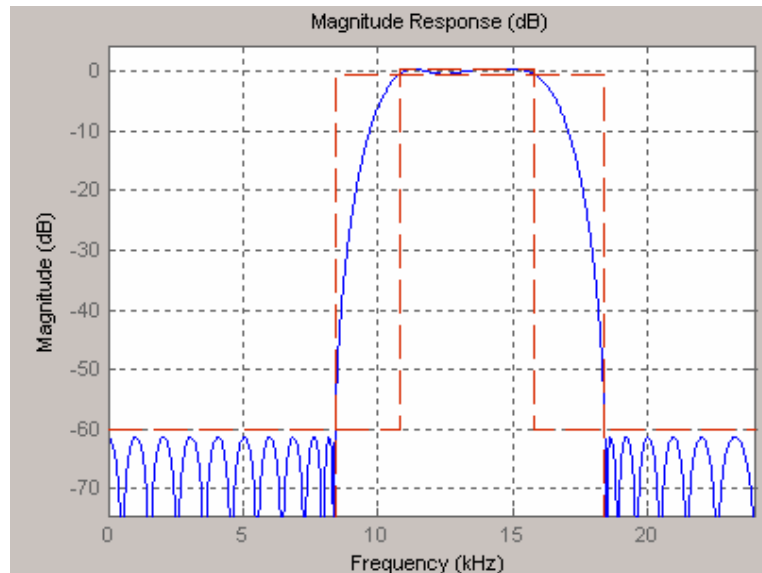
Si usted visualiza el ficha de Workspace, como se muestra en la figura siguiente se puede ver el objeto hlop que se ha colocado en el espacio de workspace.



Para revisar el trabajo, se dibuja la respuesta en magnitud del filtro utilizando el Tool de visualización del filtro.

Comprende que todos los parámetros de diseño se cumplen:

```
>>futool (Hbp) %dibujo de la respuestas en magnitud del filtro
```



B.3. DISEÑO DE FILTROS MULTIRATA Y MULTUESTADO

B.3.1 ¿POR QUÉ SON NECESARIOS LOS FILTROS MULTIRATA?

Los filtros multirata pueden apartar eficiencia a un filtro de aplicación. En general los filtros multirata son los filtros en las que diferentes partes del filtro opera a diferentes niveles.

La mas obvia aplicación de dicho filtro es cuando la frecuencia de muestreo de entrada de simple rata y la salida es un simple rata es diferida (decimacion o interpolación). Sin embargo los filtros multirata son utilizados a menudo.

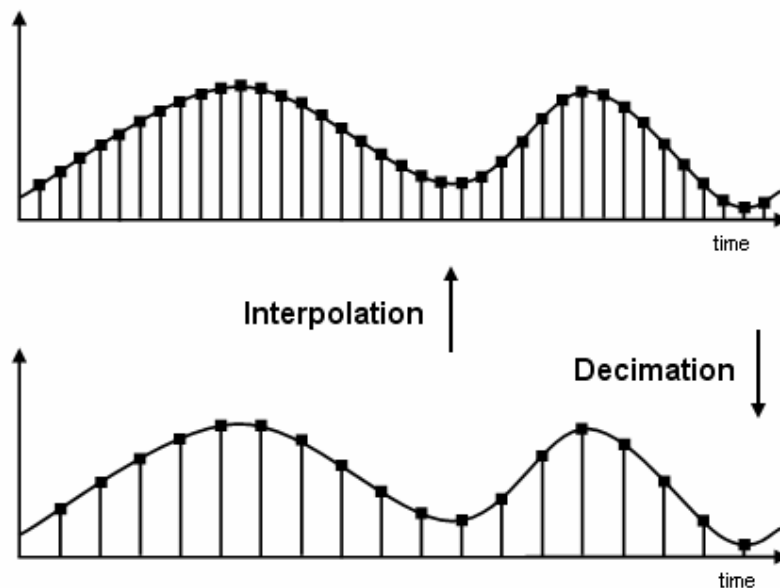
Por ejemplo puede haber un sistema en el que la frecuencia de muestreo de entrada y de salida son los mismos pero internacionalmente hay interpolación y la decimacion se puede en una señal de filtros de tal manera que el resultado final del sistema tiene la misma rata que la muestra de entrada.

Diseño de filtro por diversas razones.

B.3.2 DESCRIPCIÓN DE LOS FILTROS MULTIRATE

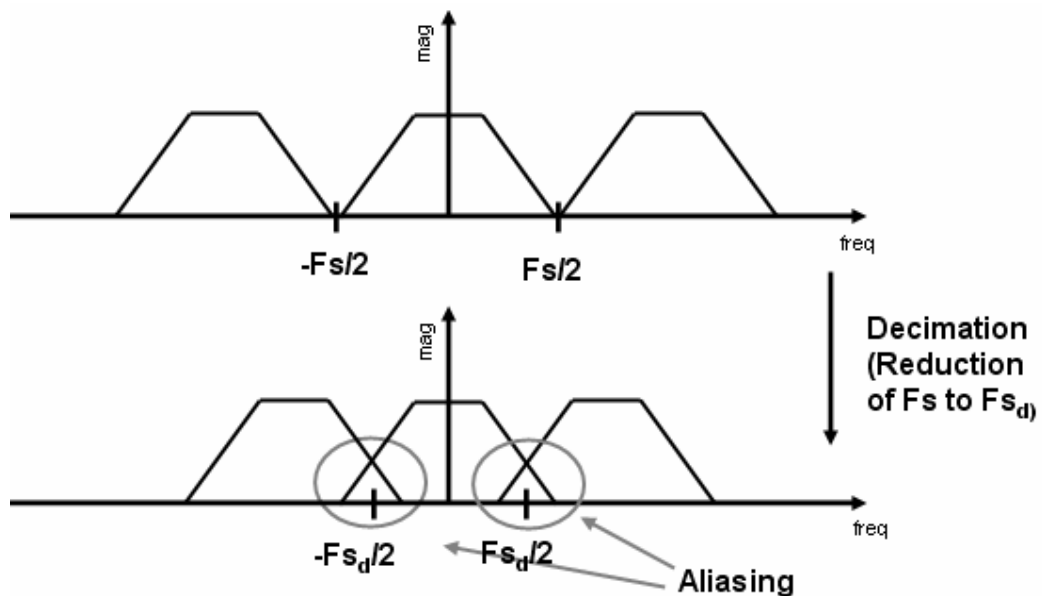
Un filtro que reduce la entrada de rata se llama un filtro decimador. Un filtro que aumenta la rata de entrada se llama interpolador.

Para ilustrar este proceso de interpolación y decimacion en el dominio del tiempo.



Si se empieza con la parte superior de la señal, a una frecuencia de muestreo F_s entonces la señal de la parte inferior muestra $F_{s/2}$. En este caso el factor de decimación M es de 2.

La siguiente figura ilustra el efecto de decimación en el dominio de la frecuencia.

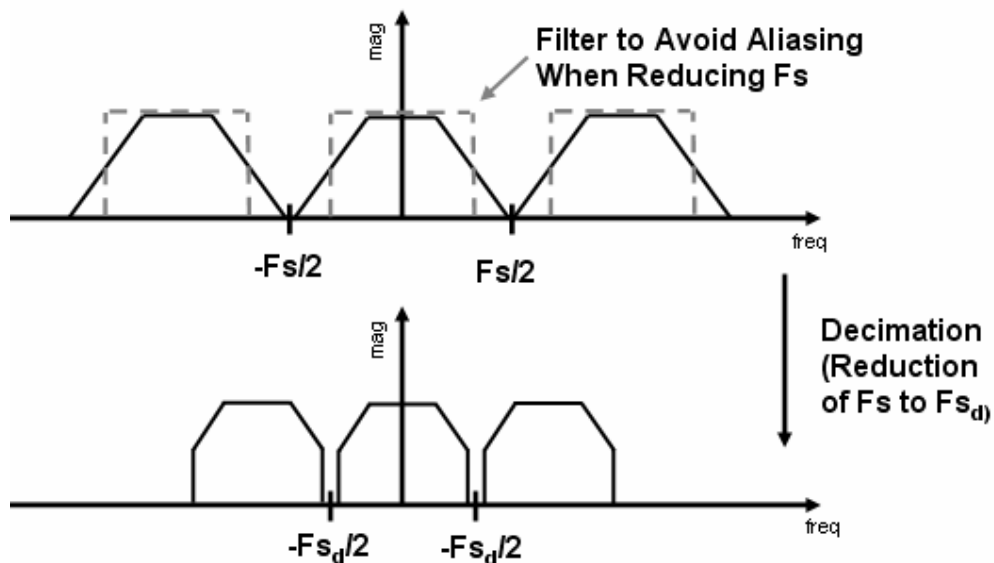


En el primer gráfico de la figura se puede ver una señal de que la muestra es crítica es decir la tasa de la muestra es igual a dos veces la frecuencia más alta de la componente de la señal de muestra.

Como tal el periodo de la señal en el dominio de la frecuencia no es mayor que el ancho de banda de la frecuencia de muestreo.

Al reducir la frecuencia de muestreo (decimacion) puede ocurrir el alising cuando las magnitudes en las frecuencias cercana a los niveles del periodo original indigustiblemente se convierte y la información acerca de estos valores se ha perdido información acerca de estos valores se ha perdido.

Para solucionar este problema la señal puede ser filtrada antes de la decimacion del proceso, evita la superposición de los aspectos de la señal $F/2$.



Un enfoque analógico debe tenerse en cuenta para evitar la interpolación de imágenes en el desempeño de la muestra en una señal. Para obtener mas información acerca de los efectos de la interpolación y decimacion en una señal de la muestra.

La siguiente lista resume algunas directrices generales y requisitos relativos a la interpolación y decimacion; por el teorema de Nyquist de las señales de banda limitado, la frecuencia de muestreo debe ser por lo menos dos veces el ancho de banda de señal. Por ejemplo se tiene un filtro pass= baja con frecuencia de máx. De 10 MHz y una frecuencia de muestreo de 60MHz.

La frecuencia mas alta puede ser manejada por el sistema sin aliasing es $60/2=30$, que es mayor de 10. Se puede establecer con seguridad $M=2$ en este caso desde:

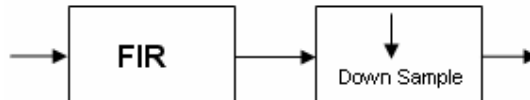
$$\left\lceil \frac{60}{2} \right\rceil = 30 \text{ que todavía es mayor de 10}$$

- Se desea una señal decimal que no cumplen con los criterios de frecuencia deben cumplir primero la interpolación y luego decimal.

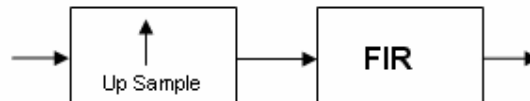
Cuando se decima, se debe aplicar el filtro en primer lugar a continuación se realiza la decimación.

Cuando se interpola una señal, se debe interpolar primer, luego filtrar la señal.

- Normalmente en la decimación de una señal, se aplica primer el filtro, lo que permite la decimación sin aliasing, como se muestra en la siguiente figura.



- Por el contrario un filtro por lo general debe aplicarse para evitar la interpolación de imágenes.



- M debe ser un número entero, aunque usted desea obtener un M de 4/5, se puede interpolar por 4 y decimar por 5, siempre que cumplan las restricciones de frecuencia. Este tipo de filtrado multirate se denomina un convertidor de frecuencia de muestreo.

B.3.3 FILTROS MULTIESTADOS:

- ¿Por qué son necesarios los filtros multiestados?: típicamente los filtros multirate y los filtros multiestados pueden aportar la eficiencia en la aplicación de los filtros.

Sin embargo tal diseño puede conservar los recursos en muchos casos y existen diferentes tipos para un filtro multiestado.

Uno de ellos es un filtro que incluye un requisito muy estrecho de transición.

Por ejemplo, es necesario diseñar un filtro pasabajo, donde la diferencia es 0.01 (normalizados).

Por ese requisito es posible diseñar un filtro único, pero será muy largo (contiene muchos coeficientes) y tiene muchas aplicaciones y adicciones por entrada de muestra.

Así este único filtro puede ser costoso y requiere memoria, que se puede ser poco práctica para aplicar en ciertas aplicaciones donde existen estrictos requisitos de Hardware. En esos casos, un filtro multiestados es una gran solución.

Otra aplicación de un filtro multiestado es para un sistema, donde hay un decimador o un factor gran interpolado.

En estos casos es aconsejable dividir el filtro multiestado en varias etapas, cada una de los múltiples decimación y múltiples factores de interpolación.

- Filtros multiestados optimas en el Toolbox del diseño de filtros: un filtro multirata interconecta cada filtro que denomina cada etapa si bien es posible llevar a cabo la automatización de un filtro multiestado.
En el diseño de un filtro manual puede ser difícil adivinar cuantas etapas proporcionaría un optimo diseño la optimación de cada etapa, y luego de filtro permite crear un especificación al objeto y el diseño de un filtro multiestado como una aplicación.
El resto de trabajo se hace de manera automática, no solamente el Toolbox de diseño, si no también optimiza el filtro de la solución total.

B.3.4 EJEMPLOS DE FILTROS MULTIRATA Y MULTIETAPA

- Ejemplo de Querniew: muestra el aumento de la eficiencia que es posible cuando se usa filtros multirata y multiestado.
- Diseño de Equiripple de rata simple y estado simple: considere las siguientes especificaciones para el diseño de un filtro pasabajo, donde las ondas se dan en unidades lineales.

```
>> Fpass=0,13;          % borde pasabanda
>> Fstop=0,14;          % borde stopband
>> Rpass=0,001;          % rizado pasabanda 0.017 db pico a pico
>> Rstop=0,0005;        % rizado stopband 66.0260 db de atenuación
>> Hf= fdesign bwpass (Fpass, Fstop, Rpass, "lineal")
```

Una fase de línea regular del diseño de equiripple utiliza especificación puede ser diseñados por evaluación de los siguientes:

```
>> Hd= diseño (Hf, "equiripple);
```

Al determinar el costo de este diseño, se puede observar que se requieren 694 multiplicadores.

```
>> costo (Hd)
Ans=
Number of multpers= 694
Number of adders= 693
Number of states= 693
Multper input sample= 694
Adder per input simple= 693
```

- Diseño de la reduccion del costo computacional usando multirata y multiestado: el número de multiplicadores para el diseño de un filtro de estado simple y multiestado es de 694. este número puede reducirse utilizando varias técnicas de multirate.

Si es un solo tipo de diseño, el número de multiplicadores requeridos por cada muestra de entradas es igual al número de multiplicadores un zeros en la aplicación.

Sin embargo se usara el diseño multirate y multiestado, la decimacion e interpolación pueden ser combinados para reducir el calculo necesario.

Para decimadores, el número medio de multiplicadores requerido por entradas de muestra esta dado por el número de multiplicadores dividido por el factor de decimacion.

```
>>Hd- multi= diseño (Hf, "varias");
```

Se puede ver el costo de diseños creados, usando el step y se puede ver ventaja de costos que se ha logrado.

```
>>costos (Hd= multi)
```

```
Ans=
```

```
Number of multipers = 392
```

```
Number of address= 383
```

```
Number of status = 348
```

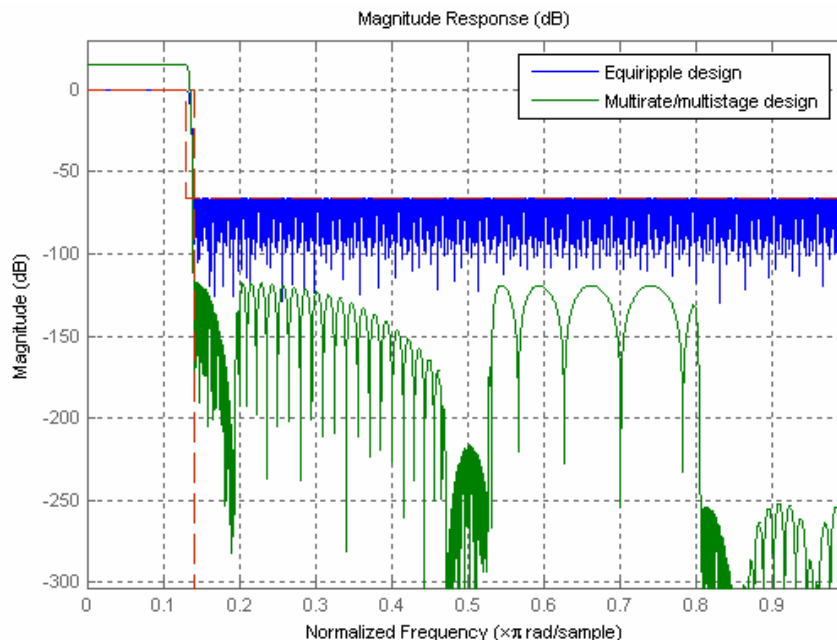
```
    Multper input simple =72.3333
```

```
    Adder per input sample = 70.1667
```

- Comparando la respuesta se puede comparar las respuestas de Equiripple y del diseño de multirate y multiestado usando ftool:

```
>>Hfvt= fvtool (Hd, Hd= multi);
```

```
>>Legend (hfvt, "Equiripple design", "multirate /multistage design")
```



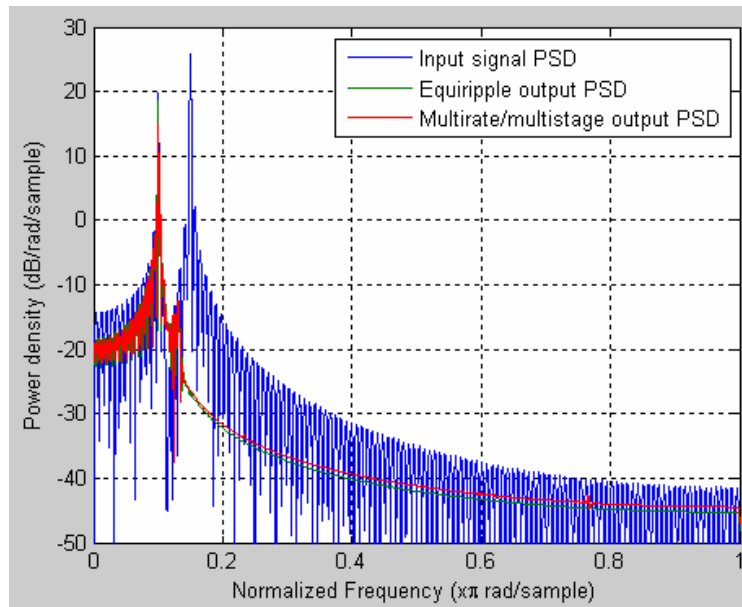
La atenuación del diseño del stopband de un multiestado es el doble que las otros diseños, esto se debe a que los decimadores deben atenuar fuera de la banda componentes por 66 dB con el fin de evitar el aliasing que viole las especificaciones.

Del mismo modo, la interpolación debe atenuar las imágenes por 66 dB. También se puede ver que la ganancia del pasabanda ya no es 0 dB, por que cada interpolador tiene su ganancia nominal igual a su factor de interpolación; y el total de factores de interpoladores para los 3 interpoladores es G.

B.3.5 COMPARACION DEL RENDIMIENTO ADICIONAL

se puede comprobar el rendimiento de un diseño multirate y multiestado trazando la densidad espectral de potencia de la entrada y de los distintos productos, y se puede comparar que la sinusoidal en 0.47* es atenuada ; comparando el diseño equiripple y el diseño multirate/ multiestado.

```
>>n= 0:1799;
>>x= sin (0.1*pi*n) +2*sin (0.115*pi*n);
>>y= filter (Hd=multi,x);
>>y= multi= filter (Hd=multi,x);
>>[pxx,w]= periodo gram (x);
>>pyy= periodo gram (y);
>>plot (w/pi, 10*log10 ([pxx,pyy,pyy=multi]));
>>xlabel ("normalized frequency (x/pi rould /sample)");
>>ylabel ("power density (dB/rad sample)");
>>legend ("input signal PSD", "equiripple output PSD", "multirate/multistage
oupput PSD");
>>axis ([0 1 = 50 30 ]);
>>grid on
```



B.4. CONVERSIÓN DE PUNTO FLOTANTE A PUNTO FIJO

Overview de los filtros punto fijo: el uso mas común de los filtros punto fijo es en chips DSO en donde la capacidad del almacenamiento de datos es limitado o sistemas embebidos y dispositivos de baja potencia, donde el consumo es necesario. Por ejemplo, la introducción de datos pueden provenir de ADE de 12 bits, el bus de datos puede ser de 16 bits, y el multiplicador puede tener 24 bits. Dentro de estas limitaciones de espacio, el toolbox de diseño filtros permite diseñar de una forma mas simple un filtro de punto fijo.

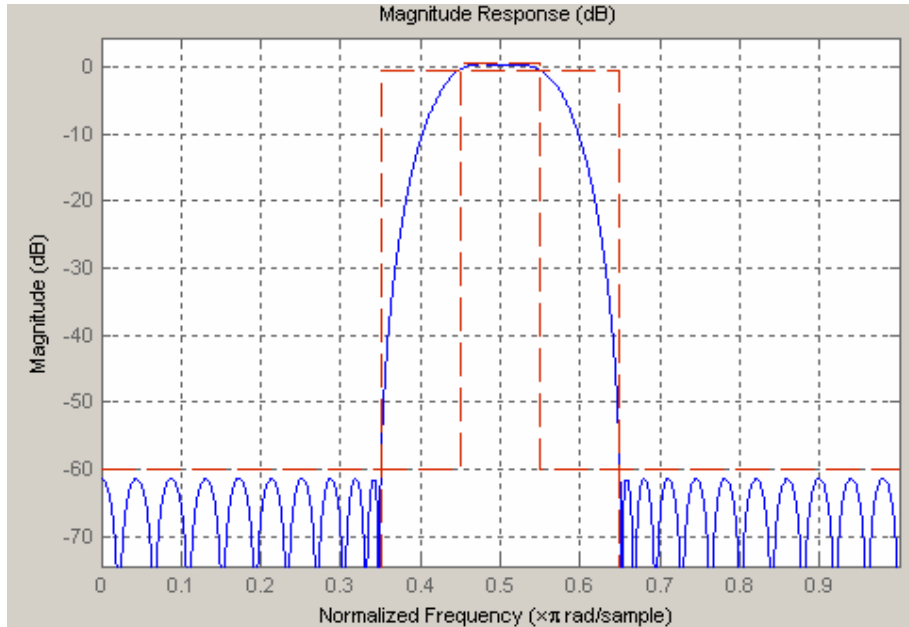
¿Qué es un filtro punto fijo?: un filtro punto fijo utiliza la aritmética de punto fijo y esta representado por una ecuación de coeficientes de punto fijo.

Descripción del proceso: la conversión de punto flotante a punto fijo y consta de dos partes principales la cuantización de los coeficientes y el análisis de la gama de dinámica.

La cuantización de los coeficientes es un proceso de conversión de los coeficientes a números punto fijo. El rango dinámico de análisis es un proceso de conversión de los coeficientes a números punto fijo. El rango dinámico de análisis es un procesote ajuste para la aplicación de cada nudo para asegurar que la longitud de función se establezca para toda la cobertura de la gama dinámica de entrada con la máxima precisión.

Diseño del filtro: se comienza por diseñar un sistema normal de punto flotante, filtro de banda como se muestra en la siguiente figura cuando el passabanda es de 0.45 a 0.55 frecuencia normalizada la cantidad de rizado aceptable en el pasabanda es de 1dB, el primer stopband es de 0 a 0.35 normalizado, el segundo

stopband es de 0.65 a 1 normalizado, y ambos proporcionan 60 dB de atenuación.



Para el diseño de este filtro, evaluando el siguiente código:

```
>>f= fdesign . bandpass (35,.45,.55,.65,.60.,1,60);
>>Hd= design (f,"equiripple")
>>fvtool(hj)
```

La última línea de código invoca el filtro herramienta de visualización, que muestra el filtro diseñado. Puede utilizar Hd que es un doble, de punto flotante de filtro como la base de referencia y un punto de partida para la conversión.

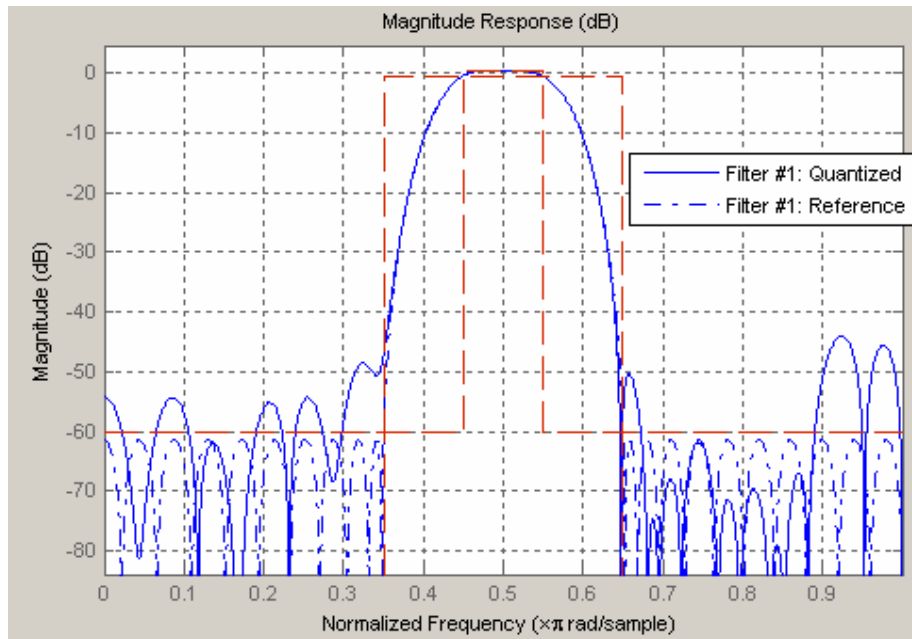
B.4.1 COEFICIENTES DE QUANTIZACION

El primer paso en la cuantización de coeficientes es valido para encontrar la longitud de palabra de los coeficientes una vez más, el Hardware suele ser utilizado al máximo, sin embargo si esta limitación es suficientemente grande hay espacio para algún ensayo y error.

Comenzando con el coeficiente de longitud de palabra de 8 y y determina si el filtro resultante es suficiente para sus necesidades.

Para establecer el coeficiente de longitud de palabra de 8, escriba el siguiente código en matlab:

```
>>Hf= Hd;
>>Hf= arithmetic="fixde";
>>set (Hf, "coeffwordlength",8);
>>fv tool (Hf)
```

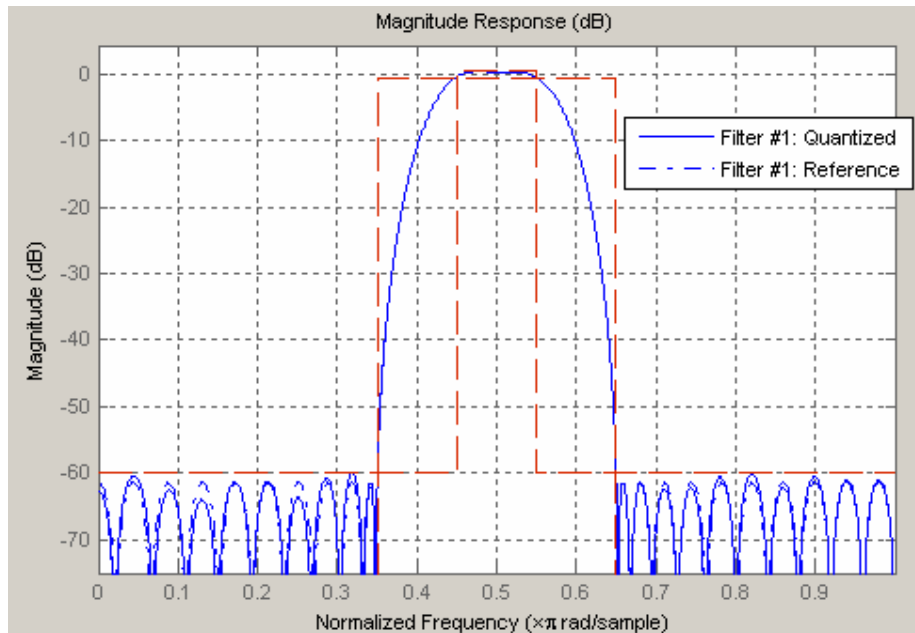


Como muestra la figura, el filtro del diseño, no cumple las limitaciones, la atenuación no es completa y no hay ruido en los bordes del stopbands.

Se puede experimentar con diferentes coeficientes de longitud de palabras si las desea. Para este ejemplo sin embargo la longitud de palabra de 12 es suficiente. Para establecer la longitud de palabra de 12 del coeficiente, evalué el siguiente código en Matlab.

```
>>set (Hf, "Coeffwordlength",12);
>>futool(Hf)
```

El filtro resultante cumple las limitaciones de diseño, como muestra la siguiente figura.



Ahora que el coeficiente de longitud de palabra se fija, hay otras limitaciones del ancho de datos que requiere atención.

```
>>info (Hf)
Discrete - time FLR filter (real)
Filter structure: direct form FIR
Filter length: 48
Stable: yes
Linear phase : yes (type 2)
Arithmetic: fixed
Numerator : s12, 14 -[-1,250000e-001-1,250000e-001]
Input: s16, 15- [-11]
Filter internals: full precision
Output : s31,29 -[-2,2 ] (auto determined)
Product: s27,29 -[-1,250000e-001 1,250000e-001] (auto determined)
Accumulator: s31,29 -[-2 2] (auto determined)
Round mode: no rounding
Overflow mode no over flow
```

La salida es de 3 bits y el multiplicador requiere 27 bits.

```
>>set (Hf, "input wordlength",12)
>>set (Hf, "filter internals", "specify precision")
>>set (Hf, "product wordlength",24); 24
>>set (Hf, "accumulator word length"); 28
>>set (Hf, "output word length"); 16
```

A pesar de que el filtro está básicamente hecho, se intenta filtrar algunos datos en esa etapa, la cual puede obtener resultados erróneos debido al overflow.

Tales overflow se produce debido a que han definido las limitaciones a trabajar, pero no han sintonizado los coeficientes del filtro para manejar adecuadamente el rango de entrada de datos, el rango dinámico se debe analizar para poder garantizar el no desbordamiento.

B.4.2 ANÁLISIS DEL RANGO DINÁMICO

El propósito del análisis del rango dinámico es de afinar la ampliación de los coeficientes.

El conjunto ideal de coeficientes es válido para todo el conjunto de datos de entrada, mientras que la longitud de fracción tiene una máxima precisión.

Si se proporciona datos que cubre el mayor rango dinámico en el filtro, el resultado de escalización es conservada y se pierde precisión.

Si se proporciona datos de rango de entrada muy estrecho, la precisión puede ser mucho mayor, pero puede producir overflow.

En este ejemplo, se utiliza el peor de los casos, la señal de entrada que abarca un amplio rango dinámico, con el fin de garantizar que no se produzca overflow. Este caso muestra que la señal de entrada es una versión escalada de la señal de la respuesta al impulso.

Para la escala de los coeficientes basados en el rango dinámico máximo, el código a evaluar es:

```
>> x = 1.9*sign(fliplr(impz(Hf)));  
Hf = autoscale(Hf, x);
```

Para comprobar que los coeficientes están en el rango, es decir no haya overflow y tener la precisión máxima posible el código a evaluar es:

```
fipref('LoggingMode', 'on', 'DataTypeOverride', 'ForceOff');  
y = filter(Hf, x);  
fipref('LoggingMode', 'off');  
R = qreport(Hf)
```

la siguiente figura muestra:

	Min	Max	
Input	-1.9003906	1.9003906	
Output	-3.2658691	3.3674316	
Product	-0.23522902	0.23522902	
Accumulator	-3.2658324	3.3674402	
	Range		
Input:	-2	1.9990234	
Output:	-4	3.9998779	
Product:	-0.5	0.49999994	
Accumulator:	-8	7.9999999	
	Number of Overflows		
Input:	0/48 (0%)		
Output:	0/48 (0%)		
Product:	0/2304 (0%)		
Accumulator:	0/2256 (0%)		

B.4.3 COMPARACIÓN DE LA RESPUESTA EN MAGNITUD Y LA RESPUESTA EN MAGNITUD ESTIMADA

Se puede utilizar el futool de GUI que hace el análisis final sobre la cuantizacion de su filtro y para ver los efectos de la atenuación de la cuantizacion del stopband.

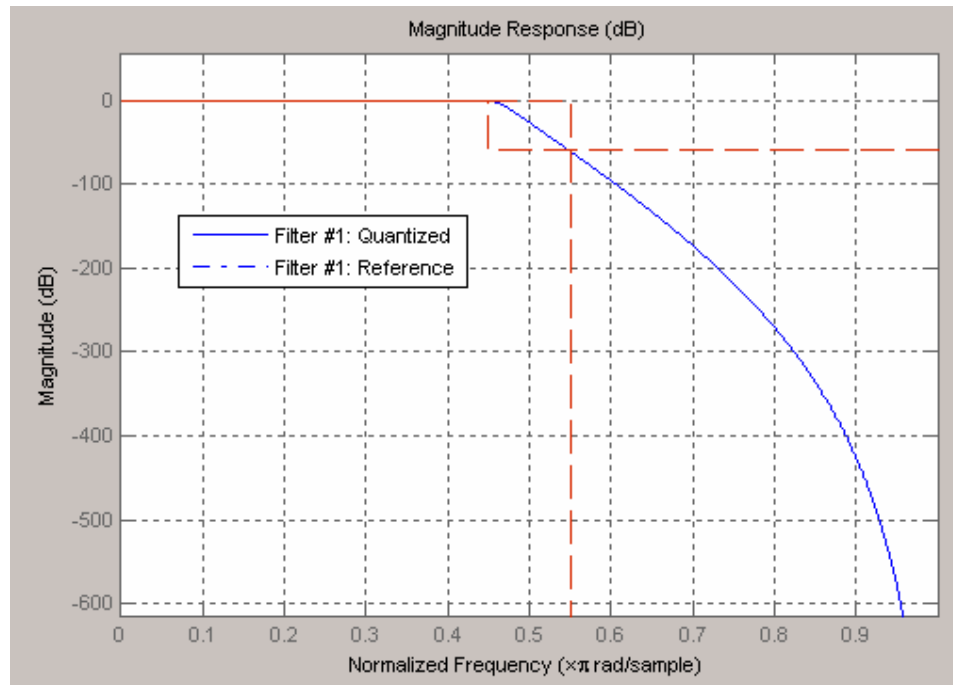
Dos controles en magnitud estimada y el espectro en el dominio del ruido de redondeo. El análisis del valor de la respuesta en magnitud estimado puede verse en el siguiente ejemplo:

```
>>h = design(fdesign.lowpass, 'butter');
```

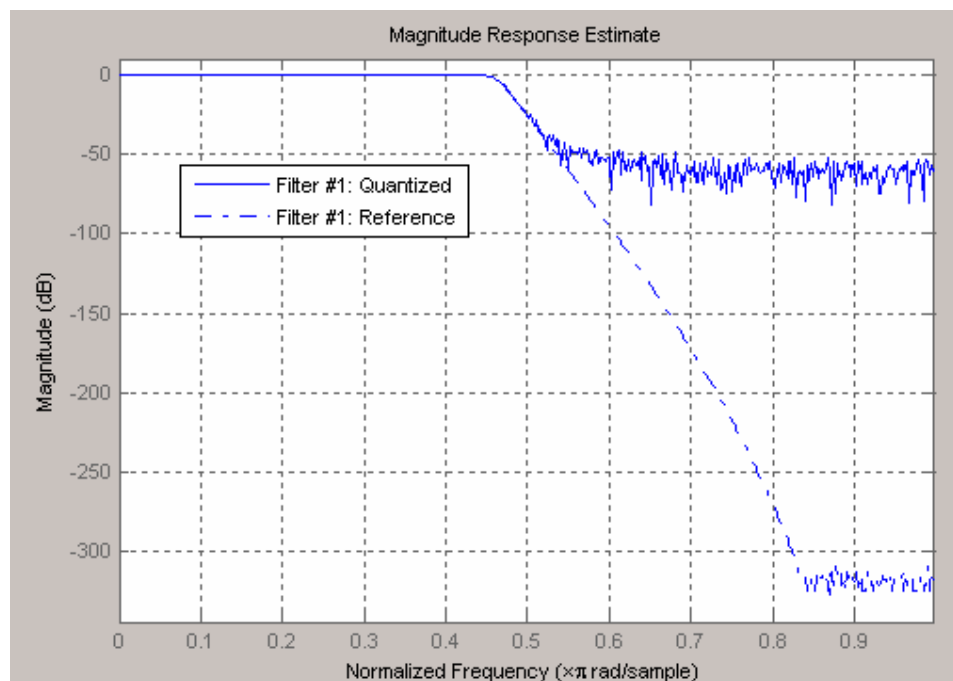
Ahora el conjunto de la aritmética de punto fijo:

```
>>h.arithmetic = 'fixed';
```

Cuando se hecha un vistazo a análisis de la respuesta en magnitud del filtro, la cuantizacion del filtro parece coincidir con el filtro original.



Sin embargo si observamos la respuesta en magnitud estimada de la grafica el menú análisis, se vera que el filtro creado no puede realizarse tan bien como indica la figura de la respuesta en magnitud.



Esto se debe a que se usa el ruido basado en el método de respuesta en magnitud estimado, se estima que la respuesta compleja en frecuencia del filtro es determinado por la aplicación de la señal a ruido de la entrada del filtro.

B.4.4 SOPORTE DEL FILTRO DE DATOS

Existen tres diferentes tipos de datos, apoyados en el Toolbox de diseño de filtros.

- Fixed: requiere el Toolbox de punto fijo y cuenta con el apoyo de los paquetes listados en el.
- Double: doble precisión, punto flotante y es el tipo de datos por defecto del Toolbox de diseño de filtros; aceptado para todas las funciones.
- Single: precisión simple, punto flotante y cuenta con los apoyos específicos.

B.4.5 SOPORTE DE TIPOS DE DATOS FIJOS

Para utilizar los tipos de datos fijos, se debe tener el Toolbox de punto fijo.

El tipo de datos fijos se reserva para cualquier filtro cuya propiedad aritmética esta ajustado a fijo.

Además de todas las funciones que trabajan con este filtro, ya se ha en diseño de análisis y acepta también el apoyo de tipos de datos fijos

Para configurar la propiedad anterior aritmética del filtro es:

```
>> f = fdesign.bandpass(.35,.45,.55,.65,60,1,60);  
>> Hf = design(f, 'equiripple');  
>> Hf.Arithmetic = 'fixed';
```

B.4.6 SOPORTE DE TIPOS DE DATOS ÚNICOS

El soporte de tipo de datos únicos se presenta en dos variedades. En primer lugar, los datos de entrada, de tipo único pueden ser alimentados en un filtro doble, en donde es convertida inmediatamente a doble. Así, mientras que el filtro trabaje en modo, la entrada de tipos de datos no es interrumpida, la segunda variedad es donde el filtro esta a precisión única. En este caso se acepta un solo tipo de datos de entrada, realizar todos los cálculos y la salida de datos en simple precisión.

Además este tipo de análisis como noisepd y freires pest también opera en el único procesamiento para configurar la precisión. Para configurar la precisión del filtro.

```
>> f = fdesign.bandpass(.35,.45,.55,.65,60,1,60);  
>> Hf = design(f, 'equiripple');  
>> Hf.Arithmetic = 'single';
```

B.5. DISEÑO DE FILTROS ADOPTIVOS

B.5.1.DESCRIPCIÓN DEL EJEMPLO DE SEÑALES MEJORES

Esta demostración ilustra una manera de utilizar, el Toolbox de los algoritmos de los filtros adoptivos si bien existe 30 algoritmos de filtros adoptivos en el Toolbox, este ejemplo demuestra dos algoritmos.

- Least jeans square: (LMS): adaptfilt 1 ms
- Normalized (LMS): adaptfilt. N 1ms

B.5.2 CREAR SEÑALES PARA ADAPTACIÓN

El objetivo es utilizar un filtro adoptivo para extraer una señal deseada desde señal por el filtrado del ruido. La señal deseada (la salida del proceso) es una señal senoidal con 1000 muestras

```
n = (1:1000)';  
s = sin(0.075*pi*n);
```

Para realizar la adaptación se requiere dos señales:

- Una señal de referencia
- Una señal – ruido que contiene una señal deseada y un componente de ruido añadido.

B.5.3 GENERACIÓN DE UNA SEÑAL / RUIDO

Para crear una señal de ruido, asumimos que el ruido v1 es auto regresivo en l sentido de que el valor del rudo en el tiempo t solo depende de sus valores anteriores y de las perturbaciones aleatorias.

```
v = 0.8*randn(1000,1); % Random noise part.  
ar = [1,1/2]; % Autoregression coefficients.  
v1 = filter(1,ar,v); % Noise signal. Applies a 1-D digital  
% filter.
```

B.5.4 DISTORSIÓN DE LA SEÑAL DESEADA PARA CREAR UNA SEÑAL RUIDOSA

Para generar una señal ruidosa que contiene la señal deseada y el ruido, se adiciona la señal del ruido v1 a la señal deseada s. El ruido distorsiona la señal senoidal x:

```
>>x= s +v1
```

Donde S es la señal deseada y v1 es el ruido. El procedimiento de filtros adaptivos trata de recuperar s de x mediante la eliminación de v1. Para completar las señales deseadas para la realización de un filtro adoptivo, el proceso de adaptación requiere una señal de referencia.

B.5.5 CREACIÓN DE LA SEÑAL DE REFERENCIA

Define una señal v2 como promedio móvil que se correlaciona con v1. Esta es la señal de referencia para los ejemplo.

```
>>Ma= [1,-8, 04, -0,2]
```

```
>>V2= filter (ma, 1,v)
```

B.5.6 CONSTRUCCIÓN DE DOS FILTROS ADOPTIVOS

Dos filtros adaptivos similares (LMS y NLMS) forman la base de este ejemplo, ambos de orden seis. Establece el orden como una variable en Matlab y se crean los filtros.

```
>>l = 7; % Seven taps or weights. Order equals 6.
```

```
>>halms=adaptfilt.lms(l)
```

```
halms =
```

```
    Algorithm: 'Direct-Form FIR LMS Adaptive Filter'
```

```
    FilterLength: 7
```

```
    StepSize: 0.1
```

```
    Leakage: 1
```

```
    PersistentMemory: false
```

```
>>hanlms=adaptfilt.nlms(l)
```

```
hanlms =
```

```
    Algorithm: 'Direct-Form FIR Normalized LMS Adaptive Filter'
```

```
    FilterLength: 7
```

```
    StepSize: 1
```

```
    Leakage: 1
```

```
    Offset: 0
```

```
    PersistentMemory: false
```

B.5.7 ELECCIÓN DEL TAMAÑO DEL PASO

Los algoritmos LMS tiene un tamaño la aplicación de correcciones aplicada al filtro adoptivo que se adapta de una interacción a la siguiente, eligiendo el tamaño del paso adecuado no es fácil, por lo general requieren experiencias en el diseño de filtros adoptivos:

- A medida que el tamaño es demasiado pequeño aumenta el tiempo para convergir el grupo de coeficientes.
- El tamaño es demasiado grande puede hacer que el filtro adaptivo diverge y nunca llegar a la convergencia. En este caso el filtro resulta no estable.

Como regla los tamaños del paso más pequeño mejoran la exactitud de la convergencia del filtro para que coincidan con las características desconocidas a expensas que el tiempo tarde en adaptarse.

El Toolbox incluye un algoritmo maxstep para determinar el tamaño máximo del paso adecuado para cada filtro LMS adaptivo que asegura que el filtro converge. A menudo, la notación para el tamaño del paso es M .

```
>>[mumaxlms,mumaxmselms] = maxstep(halms,x)
>>[mumaxnlms,mumaxmsenlms] = maxstep(hanlms) % Always equal to 2.
Warning: Step size is not in the range 0 < mu < mumaxmse/2:
Erratic behavior might result.
mumaxlms =
    0.2270
mumaxmselms =
    0.1356
mumaxnlms =
     2
mumaxmsenlms =
     2
```

B.5.8 ESTABLECIMIENTO DEL TAMAÑO DEL PASO DE UN FILTRO ADAPTIVO

La primera salida del maxstep es el valor necesario para la media de los coeficientes de convergencia, mientras que la segunda es el valor necesario para el medio cuadrado de los coeficientes a converger.

Elegir un tamaño de paso grande es a menudo causa de grandes variedades de los valores de convergencia, de modo que se elige de manera general.

```
>>halms.stepsize = mumaxmselms/30; % Can be set graphically.
>>inspect(halms) % Opens the Property Inspector in MATLAB.
>>hanlms.stepsize = mumaxmsenlms/20;
>>inspect(hanlms)
```

Si se conoce el tamaño de pass a usa, se puede establecer que las dimensiones de los pasos con la entrada de pass pueden crear los filtros.

```
>>halms = adaptfilt.lms(n,step); Adds the step input argument.
```

B.5.9 FILTROS CON FILTROS ADOPTIVOS

Ahora ya se han establecido los parámetros de adaptación de los filtros y esta listo para filtrar la señal de ruido. La señal de referencia v_2 , en la entrada al filtro adaptivo x es la señal deseada en la configuración. A través de la adaptación y , la salida de los filtros, trata de emular x , lo mas cerca posible.

Desde v_2 se correlaciona solo con la componente v_1 de x que realmente solo puede emular v_1 . la señal de error (e), menos la salida actual (y), constituye una estimación de la parte de x que no esta relacionada con v_2 -s, la señal para extraer desde x .

```
>>[ylms,elms] = filter(halms,v2,x);  
>>[ynlms,enlms] = filter(hanlms,v2,x);
```

B.5.10 CALCULO DE LA SOLUCIÓN ÓPTICA

Para calcular la solución óptica de un filtro FIR.

```
>>filterbw = firwiener(l-1,v2,x); % Optimal FIR Wiener.  
>>filteryw = filter(filterbw,1,v2); % Estimate of x using Wiener.  
>>filterew = x-filteryw; % Estimate of actual sinusoid.
```

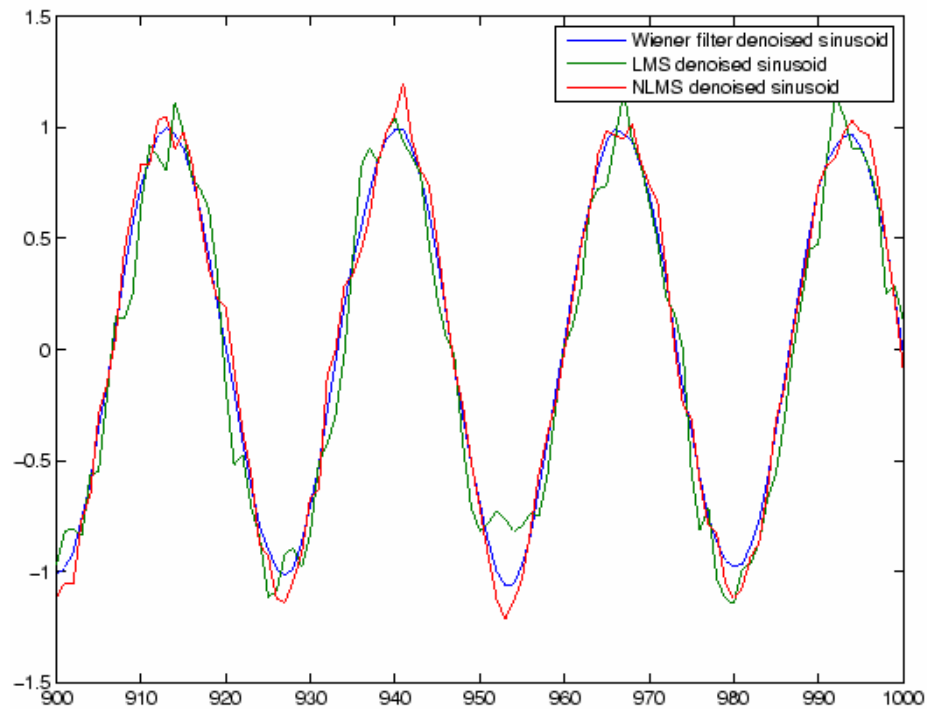
B.5.11 FIGURA RESULTANTE

El plot resultante de un filtro wiener, filtro LMS adaptivo, filtro NLMS adaptivo compara el desempeño de las diversas técnicas.

```
>>plot(n(900:end),[filterew(900:end),...  
>>elms(900:end),enlms(900:end)]);  
>>legend('Wiener filter denoised sinusoid','LMS denoised...  
>>sinusoid', 'NLMS denoised sinusoid');
```

Como punto de referencia, se incluye la señal de ruido como una línea de puntos.

```
>>hold on  
>>plot(n(900:end),x(900:end),'k:')  
>>hold off
```



B.5.12 COMPARACIÓN FINAL DE COEFICIENTES

Por ultimo hay que compara los coeficientes del filtro wiener con los coeficientes del filtro adaptivo. Si bien la adaptación de los filtros adaptivos trata de converger a los coeficientes wiener.

```
>>[filterbw.' halms.Coefficients.' hanlms.Coefficients.]\nans =
```

```
1.0221  0.8751  1.0411\n0.3345  0.1201  0.3601\n0.1217 -0.0118  0.1077\n0.0483 -0.0183  0.0081\n0.1179  0.0558  0.0420\n0.0637 -0.0049 -0.0290\n0.0216 -0.0235 -0.0222
```

B.5.13 RESETEAR EL FILTRO ANTES DE FILTRADO

Los filtros adaptivos tienen una propiedad persistentmemory que se pued4e utilizar para reproducir exactamente los experimentos por defecto el persistentmemory es falsa.

Los estados y los coeficientes del filtro se resetean antes de la filtración y el filtro no recuerda los resultados anteriores en el cual se uso el filtro. Por instantes, las

siguientes llamadas sucesorias produce la misma salida cuando persistentmemory es falsa.

```
>>[ylms,elms] = filter(halms,v2,x);  
>>[ylms2,elms2] = filter(halms,v2,x);
```

Para mantener la historia del filtro cuando hayan nuevos datos para el filtro, se habilita la memoria persistente del filtro que se encuentra en la propiedad persistentmemory y se coloca en True; en esta configuración utiliza los estados y los coeficientes de la ejecución de datos.

```
>>[ylms,elms] = filter(halms,v2,x);  
>>hlms.PersistentMemory = true;  
>>[ylms2,elms2] = filter(halms,v2,x); % No longer the same.
```

B.5.14 INVESTIGACIÓN DE LA CONVERGENCIA A TRAVÉS DE LAS CURVAS DE APRENDIZAJE

Para analizar la convergencia de los filtros adaptivos, se miran las curvas de aprendizaje. El Toolbox proporciona métodos para generar las curvas de aprendizaje, pero se necesita más de 1 interacción del experimento para obtener resultados significativos.

Esta demostración utiliza 25 muestras realizadas de la sinusoidal ruidosa.

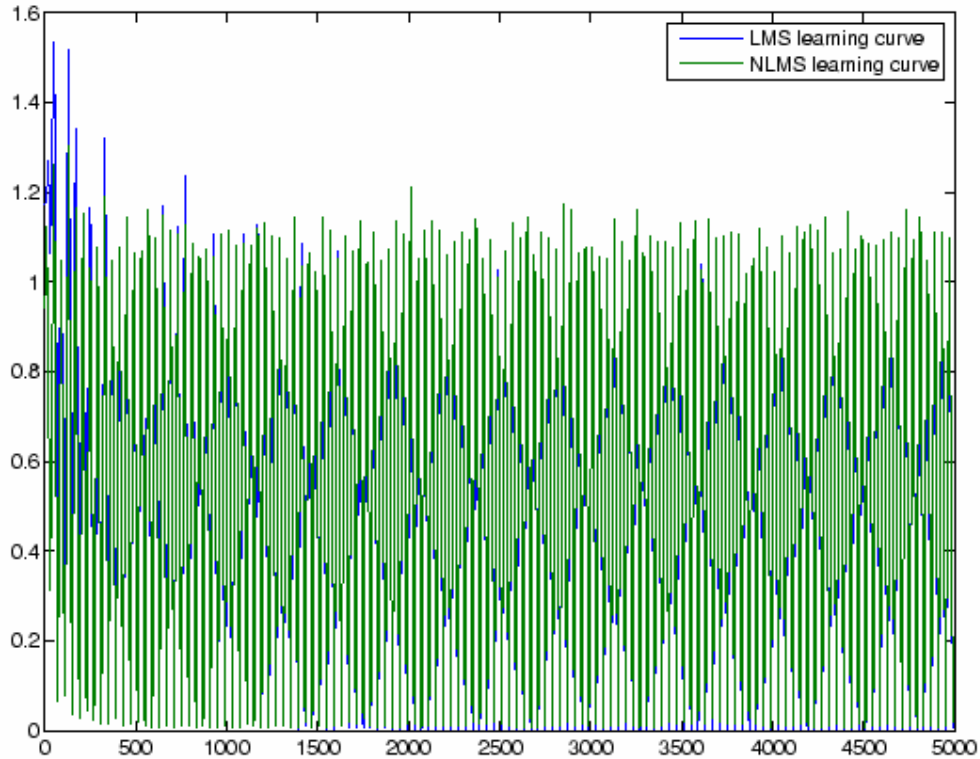
```
>>n = (1:5000)';  
>>s = sin(0.075*pi*n);  
>>nr = 25;  
>>v = 0.8*randn(5000,nr);  
>>v1 = filter(1,ar,v);  
>>x = repmat(s,1,nr) + v1;  
>>v2 = filter(ma,1,v);
```

B.5.15 CALCULO DE LAS CURVAS DE APRENDIZAJE

Para calcular el error de mean-square, se calcula cada error de cada 10 muestras. En primer lugar se restablece los filtros adaptivos para evitar el uso de los coeficientes ya calculados y de los estados que se han almacenado.

```
>>reset(halms);  
>>reset(hanlms);  
>>m = 10; % Decimation factor.  
>>mse_lms = msesim(halms,v2,x,m);  
>>mse_nlms = msesim(hanlms,v2,x,m);  
>>plot(1:m:n(end),[mse_lms,mse_nlms]);  
>>legend('LMS learning curve','NLMS learning curve')
```

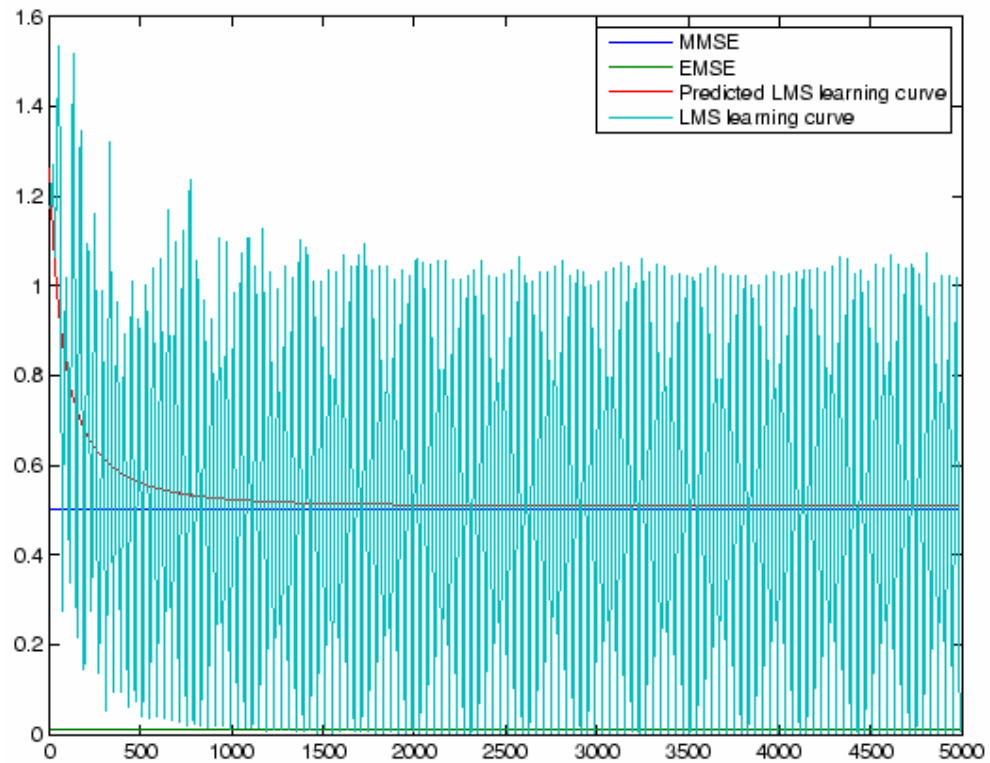
En la siguiente grafica se ven las curvas de aprendizaje calculado para los filtros adaptivos LMS y NLMS.



B.5.16 CALCULO TEÓRICO DE LAS CURVAS DE APRENDIZAJE

Para los algoritmos LMS y NLMS las funciones del Toolbox ayudan a calcular las causas de aprendizaje de manera junta con el error mínimo mean-square (MMSE) exceso del error con el mean-square (EMSE) y el valor medio de las coeficientes Matlab puede tomar algo de tiempo para calcular las curvas la figura muestra el código de las figuras y las curvas LMS actual.

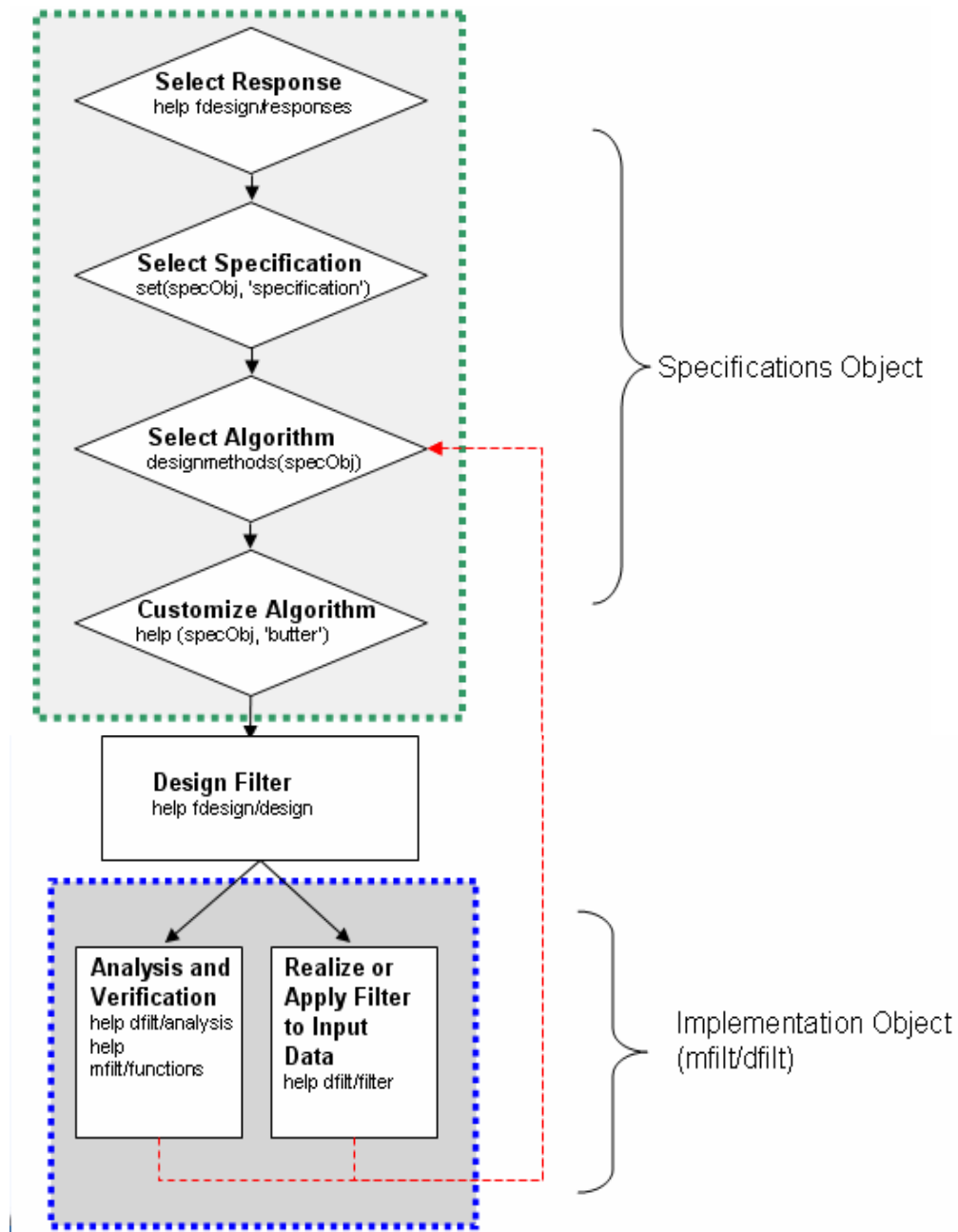
```
>>reset(halms);
>> [mmse_lms,emse_lms,meanwlms,pms_lms] = msepred(halms,v2,x,m);
>>plot(1:m:n(end),[mmse_lms*ones(500,1),emse_lms*ones(500,1),pms_lms,ms_lms])
>>legend('MMSE','EMSE','Predicted LMS learning curve','LMS learning curve')
```



B.6. “METODOLOGÍA DEL DISEÑO DEL FILTRO Y DIAGRAMA DE FLUJO DE PROCESO”

B.6.1 EXPLORACIÓN DEL PROCESO DE DIAGRAMA DE FLUJO

La siguiente figura muestra el proceso del diagrama de flujo y se muestra el orden del proceso de diseño.



Las cuatro primeras etapas del diseño de filtros se refiere a los especificaciones del objeto del filtro, mientras que los dos últimos pasos implican la aplicación de objetos del filtro.

B.6.2 SELECCIÓN DE RESPUESTA

Si escribes:

```
>>help fdesign /responses
```

En el prompt de comandos de matlab se puede ver la lista completa de todas las posibles respuestas posibles de filtro disponibles en el Toolbox de diseño de filtros.

Después de elegir una respuesta, ya sea pasabanda se empieza las especificaciones de diseño escribiendo lo siguiente:

```
>>D=f design bandpass
```

Este paso no puede ser omitido, ni es completado automáticamente por el Toolbox de diseño de filtros. Siempre, se debe seleccionar una respuesta para iniciar el proceso de diseño del filtro.

B.6.3 SELECCIÓN DE ESPECIFICACIONES

Una especificación es un conjunto de parámetros de diseños para un determinado filtro. Las especificaciones es una propiedad de las especificaciones del objeto.

Nota: una especificación no es lo mismo que las especificaciones del objeto, mas bien una especificación del objeto contiene una especificación, como una de sus propiedades.

Cuando seleccionamos una respuesta de un filtro, hay un número disponible de diferentes especificaciones, cada uno con una combinación diferente de los parámetros de diseño. En el siguiente ejemplo, el primer conjunto es la respuesta del filtro y luego la lista de especificación.

```
>> d = fdesign.bandpass; % step 1 - choose the response
>> set(d, 'specification')
ans =
    'Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2'
    'N,F3dB1,F3dB2'
    'N,F3dB1,F3dB2,Ap'
    'N,F3dB1,F3dB2,Ast'
    'N,F3dB1,F3dB2,Ast1,Ap,Ast2'
    'N,F3dB1,F3dB2,BWp'
    'N,F3dB1,F3dB2,BWst'
    'N,Fc1,Fc2'
    'N,Fp1,Fp2,Ap'
    'N,Fp1,Fp2,Ast1,Ap,Ast2'
    'N,Fst1,Fp1,Fp2,Fst2'
    'N,Fst1,Fp1,Fp2,Fst2,Ap'
    'N,Fst1,Fst2,Ast'
    'Nb,Na,Fst1,Fp1,Fp2,Fst2'
```

```

>> d = fdesign.decimator; % step 1 - choose the response
<<% get a list of available specifications
>> set (d, 'specification')
ans =
    'TW,Ast'
    'N'
    'N,Ast'
    'N,TW'

```

Después de seleccionar la especificación, que incluye la totalidad dada de los parámetros de diseño de filtros, puede configurarlo de la siguiente manera:

```

>> d = fdesign.lowpass; % step 1
>> % step 2: get a list of available specifications
>> set (d, 'specification')
ans =
    'Fp,Fst,Ap,Ast'
    'N,F3dB'
    'N,F3dB,Ap'
    'N,F3dB,Ap,Ast'
    'N,F3dB,Ast'
    'N,F3dB,Fst'
    'N,Fc'
    'N,Fc,Ap,Ast'
    'N,Fp,Ap'
    'N,Fp,Ap,Ast'
    'N,Fp,F3dB'
    'N,Fp,Fst'
    'N,Fp,Fst,Ap'
    'N,Fp,Fst,Ast'
    'N,Fst,Ap,Ast'
    'N,Fst,Ast'
    'Nb,Na,Fp,Fst'
>> %step 2: set the required specification
>> set (d, 'specification', 'N,Fc')

```

En caso de no realizar este paso explícitamente el Toolbox de diseño de filtros, selecciona un pliego de condiciones de respuesta que se eligió, e incluso proporciona valor predeterminado para todos los parámetros de diseño incluso en las especificaciones.

B.6.4 SELECCIÓN DEL ALGORITMO

La disponibilidad de los algoritmos depende de respuesta elegida del filtro y los parámetros de diseño.

En otras palabras para el mismo filtro pasabajo, la modificación de las especificaciones también cambia la cadena de algoritmos disponibles.

En el siguiente ejemplo, para un filtro pasa bajo con una especificación “N,Fe” solo se dispone se un algoritmo de ventana sin embargo para una especificación de “Fp,Ast,Fst, Ap” una sede de algoritmos este disponible.

```
>> %step 2: set the required specification
>> set (d, 'specification', 'N,Fc')
>> designmethods (d) %step3: get available algorithms
Design Methods for class fdesign.lowpass (N,Fc):
window
>> %step2: set a different specification
>> set (d, 'specification', 'Fp,Fst,Ap,Ast')
>> designmethods (d) %step3: get available algorithms
Design Methods for class fdesign.lowpass (Fp,Fst,Ap,Ast):
butter
cheby1
cheby2
ellip
equiripple
ifir
kaiserwin
multistage
```

Para aplicar el algoritmo elegido, (Butterworth), se evalúa lo siguiente:

```
>> f = design(d, 'butter');
```

El anterior comando crea el filtro, donde f es el filtro de aplicación de objeto.

B.6.5 DISEÑADO EL FILTRO

En esta tarea se introduce un nuevo objeto, el objeto del filtro o dfilt para crear un filtro se usa el sig comando

```
>>f=desing (d);
```

Donde f es el objeto del filtro, refiriéndose a dfilt, y de es el objeto de especificaron. Este código crea un filtro sin tener en cuenta sus especificaciones de algoritmo. Cuando el algoritmo se ha especificado el Toolbox de diseño de filtro selecciona al mejor. Para aplicar el algoritmo elegido se usa el comando de diseño, pero especificando el algoritmo Butterworth, de la siguiente manera:

```
>>F=desing (d,"butter");
```

Donde f es el objeto del filtro y de especificación al objeto para obtener ayuda y ver todas las especificaciones posibles.

```
>>help fdesign/ design
```

Es el comando de ayuda no solo describe las opciones para el comando de diseño si no también refiere a las opciones de los métodos de algoritmos.

En el siguiente ejemplo se diseña un filtro pasabanda con especificaciones dadas:

```
>>f=design(d, "butter", filterstructure, "df2sos")
>> f = design(d, 'butter', 'filterstructure', 'df2sos')
f =
  FilterStructure: 'Direct-Form II, Second-Order Sections'
    Arithmetic: 'double'
      sosMatrix: [7x6 double]
    ScaleValues: [8x1 double]
  PersistentMemory: false
```

El diseño del filtro, como la elección de la respuesta, se debe hacer de forma explícita. El Toolbox de diseño de filtro no crea un filtro sin que se diga específicamente lo que se requiere.

B.6.6 ANÁLISIS DE DISEÑO

Después de que el filtro está diseñado se puede analizar para determinar si el filtro cumple con los criterios de diseño. En el Toolbox de diseño de filtros el análisis se divide en tres secciones principales:

- Análisis del dominio de la frecuencia: incluye la respuesta en magnitud, grupo de retardos y poll zero.
- Análisis del dominio del tiempo incluye impulso y respuesta del paso.
- Implementación del análisis: incluye ruido de cuantización.

Para mostrar la ayuda de un filtro en tiempo discreto.

```
>>help dfilt /análisis
```

Para mostrar la ayuda de un filtro multirate

```
>>help mfilt/ functions
```

Para mostrar la ayuda de un filtro farrow

```
>>help farrow /functions
```

Para analizar un filtro, se debe especificar esto.

B.6.7 REALIZACIÓN O APLICACIÓN DE LA ENTRADA DE DATOS DE UN FILTRO

Después de que el filtro este diseñado y optimizado, se puede utilizar los datos reales de la entrada para filtrar. El comando básico del filtro toma una entrada de datos x el filtro es a través del objeto de filtro y produce una salida y .

```
>>Y= filter (filter obj, x)
```

Este paso nunca se realiza de forma automáticamente por el Toolbox de diseño de filtros. Para filtrar los datos se debe ejecutar explícitamente este paso. Para comprender las formas en que funciona este comando.

```
>>help dfilt/filter
```

Nota: si se tiene simulink, se tiene la opción de exportar este filtro a un bloque de simulink usando el comando `realizemol`. Para obtener ayuda de este comando escriba:
>>help realizemol

B.7. GRAFICA PARA FDESIGN

B.7.1 INTRODUCCIÓN AL FILTERBUILDER

La función `filterbuilder` proporciona una interfaz gráfico para el `fdesign` objeto-objeto orientado al diseño de filtros y se estima a reducir el tiempo de desarrollar durante el proceso de diseño del filtro.

El `filterbuilder` utiliza un enfoque centrado de especificaciones par encontrar el algoritmo para la respuesta deseada.

El `filterbuilder` GUI contiene muchas características que no están disponibles en el `FDATool`, por ejemplo.

- Filtros FIR multiestado y multirata para diseños eficientes de bandas estrecha y transición.
- Interpoladores y decimadores con óptima multiestado generado automáticamente.
- Diseño de filtros IIR halfband: (incluye filtros IIR con fase lineal aproximados).
- Filtros Nyquist optimo multiestado.
- Diseños de CIC (incluye compensadores CIC)
- Diseños de filtros farrow.
- Onda de diseños de filtros digitales
- Magnitud arbitraria y diseño de fase.

B.7.2 PROCESO DE DISEÑO DEL FILTERBUILDER

El proceso de diseño cuando se usa el filterbuilder es al proceso descrito en la selección de diseñando un filtro.

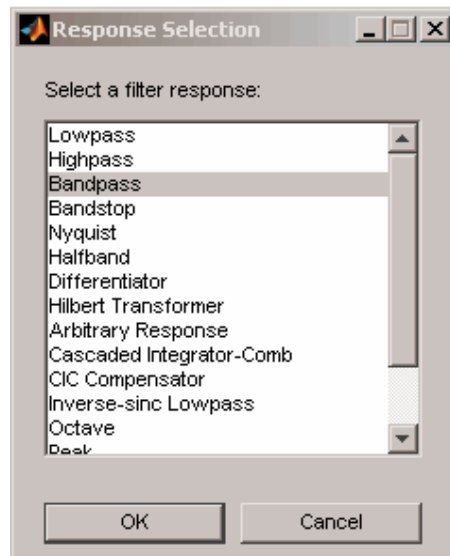
La idea es elegir las limitaciones y especificaciones del filtro y utilizarlo como punto de partida del diseño

B.7.3 SELECCIONE LA RESPUESTA DEL FILTRO

EL Toolbox de filterbuilder se abre escribiendo

```
>>FILTERBUILDER
```

En el prompt de Matlab, se abrirá el siguiente cuadro de dialogo con una lista de todas las posibles respuestas de filtro disponible en el Toolbox de diseño de filtros.



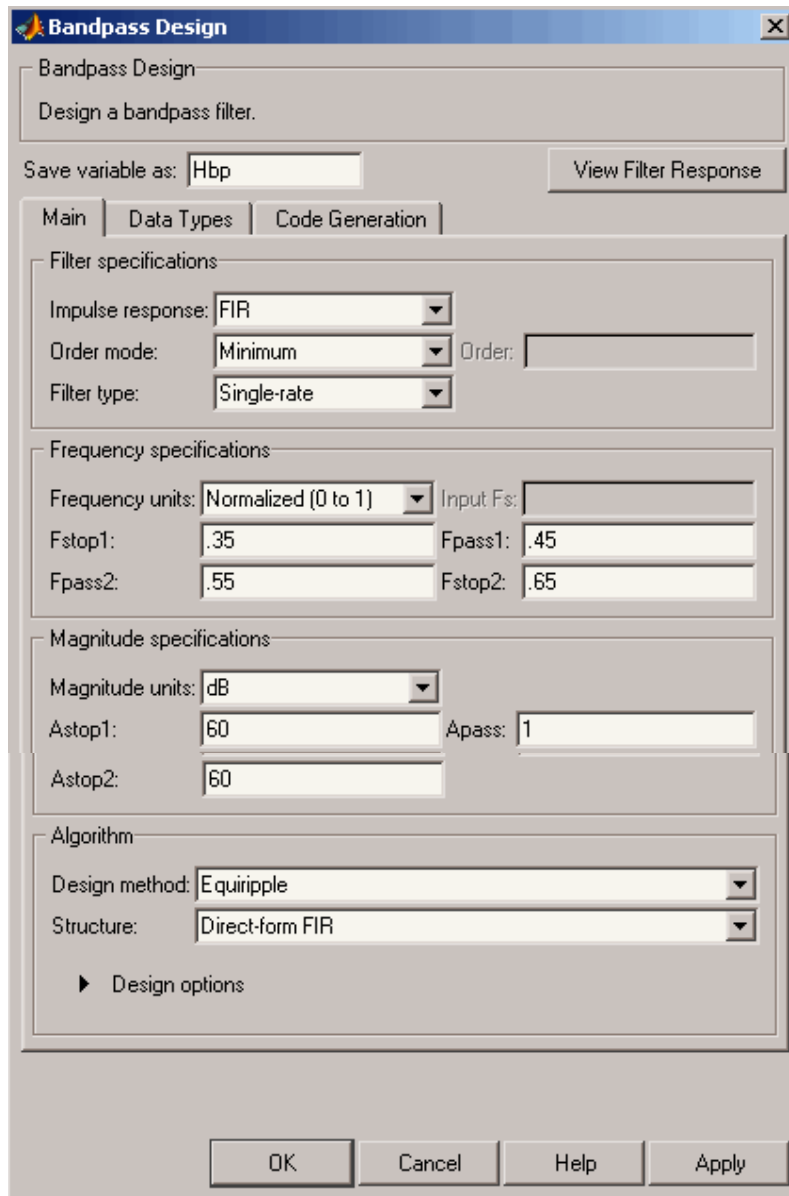
Nota: este paso no puede ser omitido por que no se completa automáticamente por el Toolbox de diseño de filtros.

Se debe seleccionar una respuesta para iniciar el proceso de diseño del filtro.

Después de elegir una respuesta, por decir passbanda, se empieza el diseño de las especificaciones del objeto y el cuadro de dialogo diseño Bandpass aparecerá. Este cuadro de dialogo contiene un panel principal, un panel de tipos de datos y un papel de generación de código.

Las especificaciones del filtro son generalmente fijados en el panel principal del cuadro de dialogo.

El panel de tipos de datos proporciona la precisión y los tipos de datos, y el panel de generación de código contiene diversas opciones para las implementaciones de el complemento de diseño de filtro.



Bandpass Design

Design a bandpass filter.

Save variable as:

Main | Data Types | Code Generation

Filter specifications

Impulse response:

Order mode: Order:

Filter type:

Frequency specifications

Frequency units: Input Fs:

Fstop1: Fpass1:

Fpass2: Fstop2:

Magnitude specifications

Magnitude units:

Astop1: Apass:

Astop2:

Algorithm

Design method:

Structure:

► Design options

El cuadro de dialogo de diseño bandpass contiene todos los parámetros que se necesita para determinar las especificaciones de un filtro pasaband. Los parámetros enumerados en el panel principal dependerán del filtro que este diseñado.

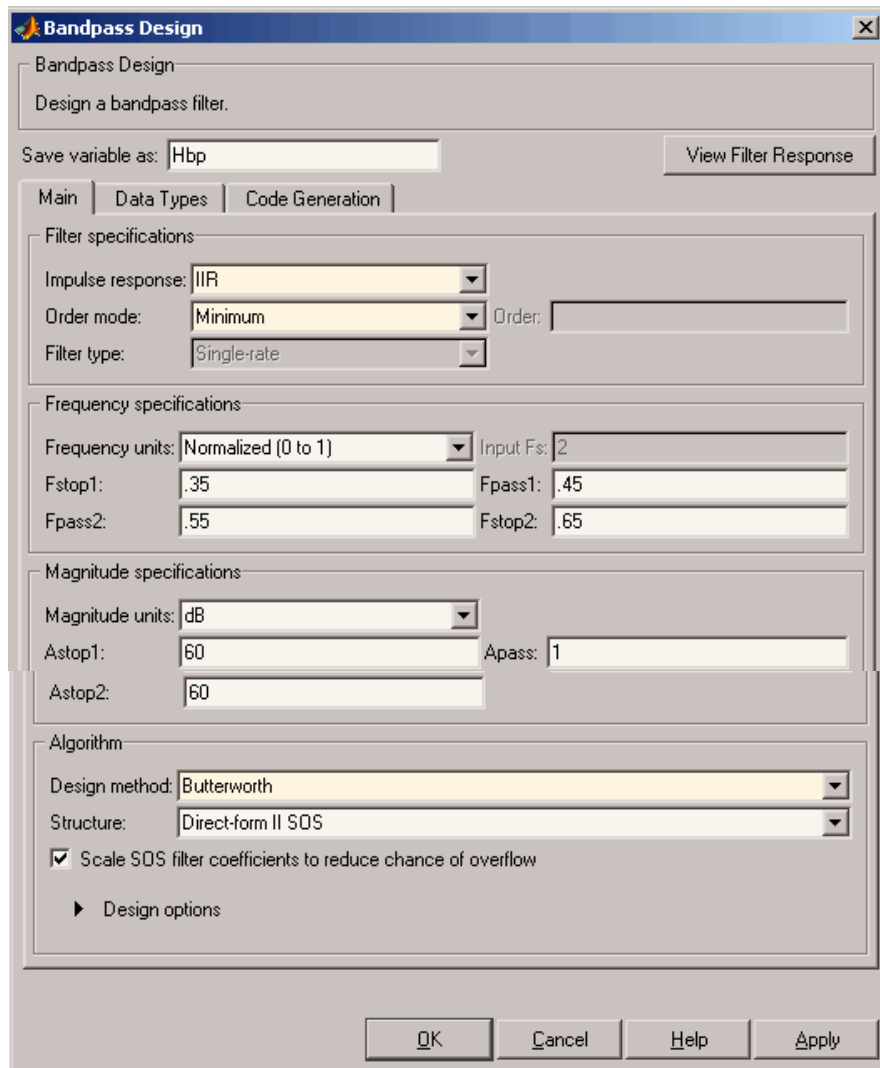
B.7.4 SELECCIÓN DE LA ESPECIFICACIÓN

Para elegir las especificaciones para el filtro pasabanda se puede empezar por selecciona una respuesta en impulso, modo orden y tipo de filtro en el marco de las especificaciones del filtro en el panel principal.

Nota: la frecuencia, la magnitud y las especificaciones del algoritmo son independientes y pueden cambiar sobre base de selección de las especificaciones del filtro.

B.7.5 SELECCIÓN DEL ALGORITMO

Los algoritmos disponibles para su filtro dependerá de la respuesta del filtro y los parámetros de diseño que ha seleccionado anteriormente por ejemplo, en el caso de un filtro pasa banda, si la respuesta de impulso es IIR y el modo orden es establecida como mínimo, el método de diseño disponible en el Butterworth, chebyshev tipo I o II, o elípticas y el método de diseño disponible IIR es menor p-norm.



The image shows a 'Bandpass Design' dialog box with the following sections and settings:

- Bandpass Design**: Design a bandpass filter.
- Save variable as:** Hbp
- View Filter Response** button
- Tabs:** Main (selected), Data Types, Code Generation
- Filter specifications:**
 - Impulse response: IIR
 - Order mode: Minimum
 - Filter type: Single-rate
- Frequency specifications:**
 - Frequency units: Normalized (0 to 1)
 - Input Fs: 2
 - Fstop1: .35, Fpass1: .45, Fpass2: .55, Fstop2: .65
- Magnitude specifications:**
 - Magnitude units: dB
 - Astop1: 60, Apass: 1, Astop2: 60
- Algorithm:**
 - Design method: Butterworth
 - Structure: Direct-form II SOS
 - ☒ Scale SOS filter coefficients to reduce chance of overflow
 - ▶ Design options
- Buttons:** OK, Cancel, Help, Apply

B.7.6 PERSONALIZAR EL ALGORITMO

Al ampliar las opciones de diseño de la estructura del algoritmo, se puede personalizar el algoritmo especificado. Las opciones disponibles dependerán del algoritmo y ajustes que se han seleccionado en el cuadro de dialogo.

Bandpass Design

Design a bandpass filter.

Save variable as:

Main | Data Types | Code Generation

Filter specifications

Impulse response:

Order mode: Order:

Filter type:

Frequency specifications

Frequency units: Input Fs:

Fstop1: Fpass1:

Fpass2: Fstop2:

Magnitude specifications

Magnitude units:

Astop1: Apass:

Astop2:

Algorithm

Design method:

Structure:

☒ Scale SOS filter coefficients to reduce chance of overflow

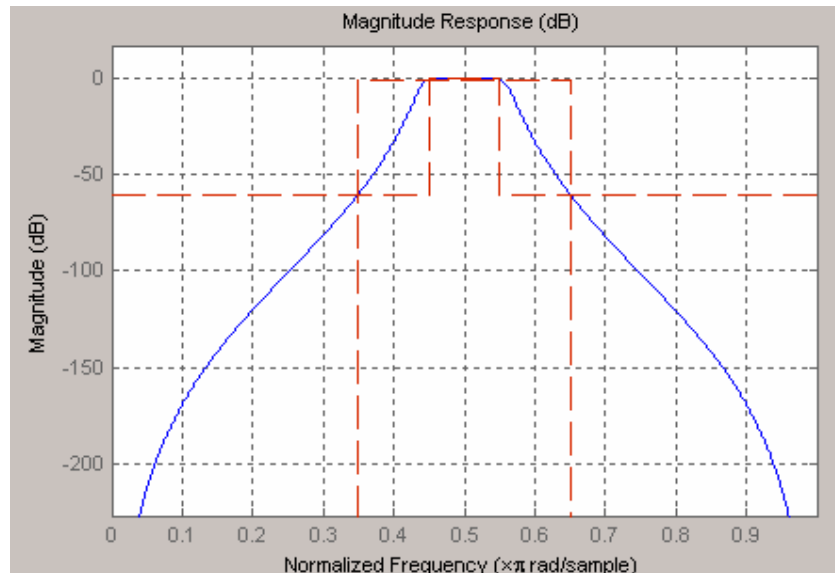
Design options

Match exactly:

En el caso de un filtro IIR pasa banda se usa el método Butterworth, las opciones de diseño están disponibles en Match exactamente como se muestra en la figura.

B.7.7 ANÁLISIS DEL DISEÑO

Para analizar la respuesta del filtro, haga clic en el botón ver respuesta filtrada. El filtro de visualización del Toolbox abre la grafica de magnitud de la reopuesta filtro.



B.7.8 REALIZACIÓN O APLICACIÓN A LA ENTRADA DE DATOS DE UN FILTRO

Cuando se haya logrado la respuesta del filtro a través de los interacciones de diseños y el análisis usando el tool de visualización del filtro, aplicando el filtro a la entrada de datos.

En el Tool de visualización de filtros, haga clic en aceptar y el Toolbox de diseño de filtros crea el objeto filtro con el nombre especificando en el campo “guardar como” “variable” y las exportaciones al work de Matlab. Entonces el filtro esta listo para ser usado en el filtrado de datos reales de entrada. La entrada de datos es x y la salida y :

```
>>y= filter(filterobj,x)
```

Para comprender la forma en que trabaja el comando:

```
>>help dfilt/filter
```

Si se tiene simulink, la opción de explotar este filtro a un bloque de simulink es usar el comando realizemol.

```
>>help realizemol
```

B.8. DISEÑO DE UN FILTRO FIR USANDO FILTERBUILDER

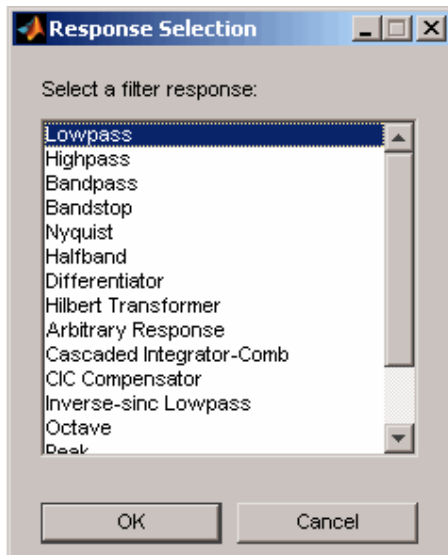
B.8.1 DISEÑO DE FILTRO FIR

Usando el filterbulder se diseña un filtro finito con respuesta al impulso. Para diseñar un filtro FIR pasabaja usando filterbuilder:

1. abra el filterbuilder GUI colocando en el prompt de matlab:

```
>>filterbuilder
```

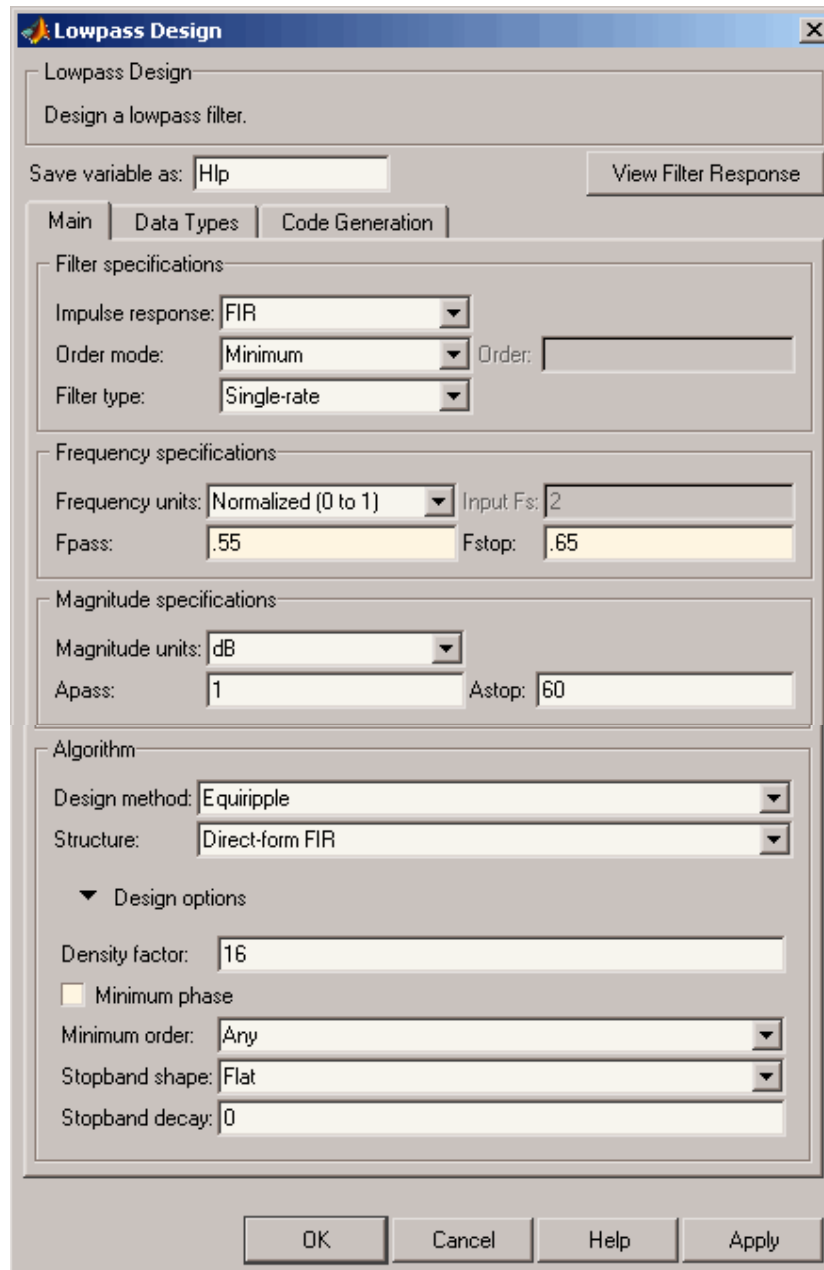
aparezca el cuadro de dialogo de “selección de respuesta”,



donde se puede seleccionar el tipo de respuesta del filtro. Seleccione en la lista lowpass de clic en el botón ok. Se abrirá el cuadro de dialogo design lowpass. En este cuadro se hacen los siguientes cambios:

Fpass valor de 0.55

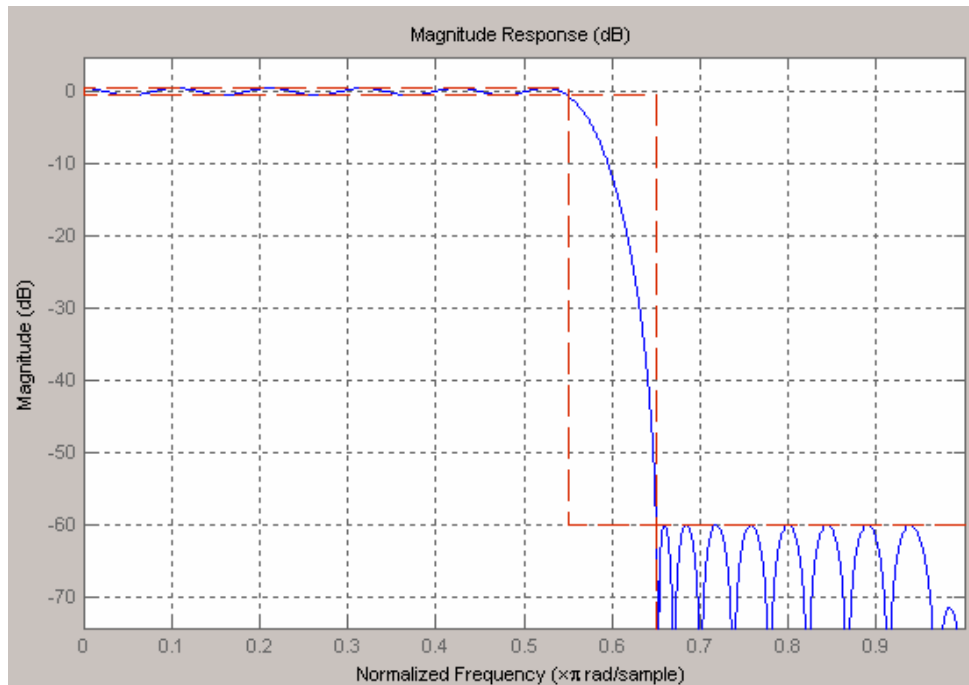
Fstop valor de 0.65



Dar clic en Apply, y aparece el siguiente mensaje en el prompt de Matlab

>>the variable"HLP" has been exported to the command window.

Para comprobar el diseño, haga clic en View Filter Response. Aparezca el Tool de visualización de filtro, mostrando una grafica de el filtro en respuesta en magnitud.



B.9. DISEÑO AVANSADO DE FILTROS EN LA HERRAMIENTA FDA.

B.9.1 DESCRIPCIÓN DE LAS CARACTERÍSTICAS FDTAOL

El toolbox de diseño de filtros añade nuevo cuadro de diálogos y nuevos modos de funcionamiento. La herramienta de análisis y diseños de filtros FDToll proporciona Toolbox de procesamiento de señales.

De los nuevos cuadros de diálogos esta el de fijación de parámetros de cuantización y el de transformaciones de frecuencia pueden:

- Diseño avanzado de filtros de Toolbox de procesamiento de señales no prevé Toolbox de diseño de desarrollo.
- Ver modelos de filtro de Simulink de estructuras disponibles en el Toolbox.
- Filtros cuantizados de doble presicion que se usa en el modo de diseño de la GUI.
- Importación de filtros cuantizados en doble presicion.
- Análisis de filtros cuantizados.
- Escala de filtros de segundo orden.

Seleccione los ajustes de la cuantizacion es en la herramienta de filtro cuantiado.

- Coeficiente: selecciona las opciones de cuantizacion aplicado a los coeficientes del filtro.
- Input/output: controla la entrada y salida de datos en el proceso de filtros.

- Filterinternals: especifica como la forma aritmética del comportamiento del filtro
- Diseño del filtro multirata.
- Transformación de filtros FIR e IIR de una respuesta a otra.

Después de importar filtro para el FDATOOL, las opciones del cuadro de dialogo de cuantizacion permite la cuantizacion del filtro e investigación de los efectos de cuantizacion de diversas opciones.

El cuadro de dialogo de las opciones de transformación de frecuencia permite cambiar la respuesta en frecuencia del filtro.

B.9.2 USO DEL FDATOOL CON EL TOOLBOX DE DISEÑO DE FILTROS

Al adicionar el toolbox de diseño de filtros al FDATOOL, se utiliza una nueva respuesta del filtro para desarrollar filtros que cumplan con los requisitos mas complejos que se puedan diseñar en toolbox de procesamiento de imágenes.

Utiliza el nuevo filtro de las respuestas para desarrollar filtros que cumplan con los requisitos mas complejos que los que se pueden diseñar en el toolbox de procesamientos de señales.

Si bien los diseños de FDATOOL es tan disponibles como funciones de comandos de líneas, la interfaz graficas de un cuadro FDATOOL hace que el proceso de diseño es mas claro y mas facil de lograr.

Como se relaciona un tipo de respuesta, en las acciones de las paneles en FDATOOL permite configurar los valores que definen el filtro.

También se puede ver que el área de análisis incluye un diagrama que describa las opciones de respuesta del filtro que se elige.

Mediante el examen de la macara se puede ver como se definen las opciones y como utilizarlos. Aunque esto puede ser sencillos para la respuesta del filtro lowpass el establecimiento de la respuesta o de las opciones de arbitrariedad son mas complejas. Al cambiar el método de diseño del filtro se cambia el tipo de opciones de respuesta disponible.

B.9.3 EJEMPLO DE UN FILTRO NOCTH

El objetivo de un filtro nocth es el de eliminar una o varias frecuencias de un espectro mas amplio. Se debe especificar las frecuencias para eliminar el filtro mediante el establecimiento de opciones de diseño en FDATOOL adecuadamente:

Tipo de respuesta

Método de diseño

Especificaciones de frecuencia

Especificaciones de magnitud

Aquí esta como diseñar un filtro nocth con una señal de entrada musiol de HOH_z

:

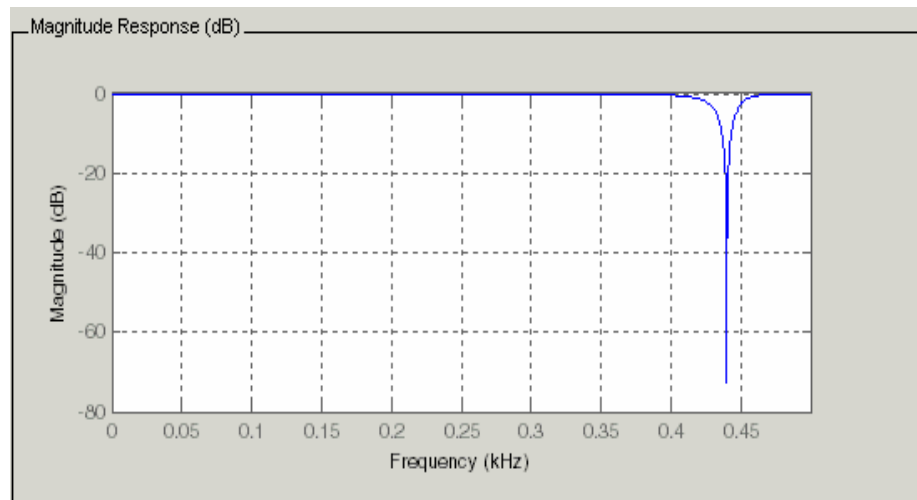
1. seleccionar el modo notching de la lista de tipos de respuestas.

2. seleccionar el método de diseño del filtro IIR y elegir el single notch de la lista
3. para las especificaciones de frecuencia establecer las unidades de H_z y F_s la escala máxima de frecuencia en 10.000.
4. establecer la localización del centro del notch en frecuencia normalizada H_z para el centro del notch de 440 H_z , introduzca 440.
5. para conformar el notch colocar el ancho de banda B_w de 40.
6. las especificaciones de magnitud debe estar en dB y el apach.
7. haga clic en el diseño del filtro.

FDATool calcula las coeficientes del filtro y la grafica de la respuesta en magnitud del filtro es descrito en el área de análisis.

Cuando se diseña un solo filtro Notch no se tiene la opción de fijar el orden del filtro; esta opción esta desactivada.

Su filtro debe verse así:

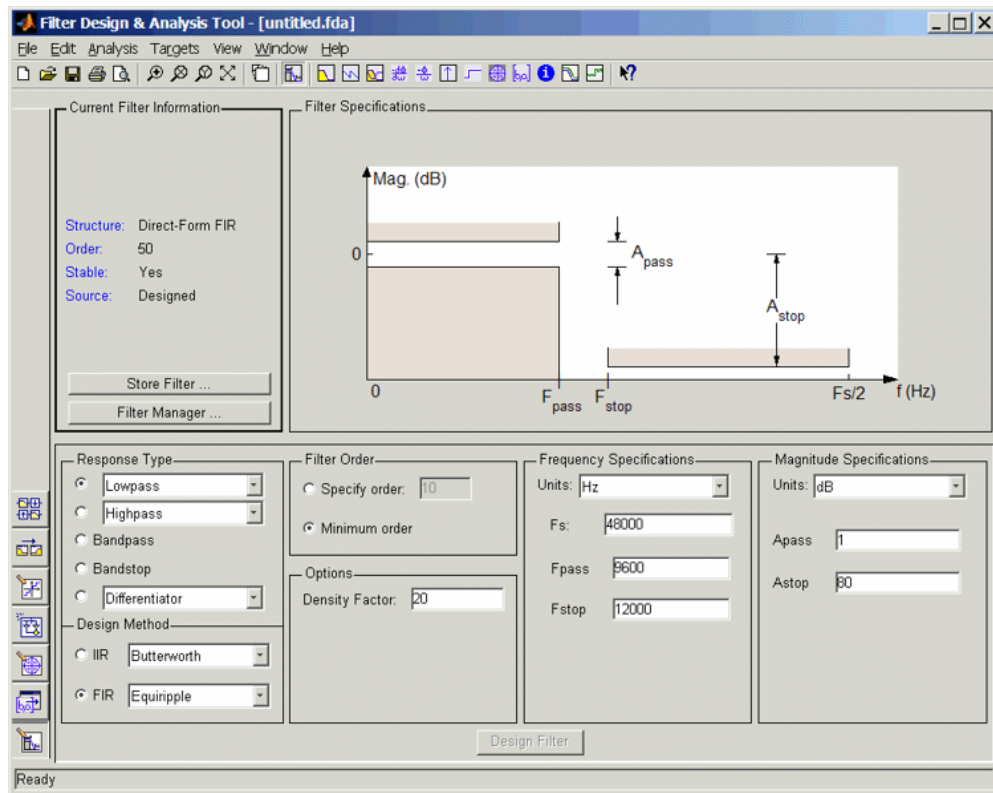


Para obtener más información acerca de un método de diseño, por ejemplo para obtener mayor información de la que para el filtro Notch en FDATool, introduzca:

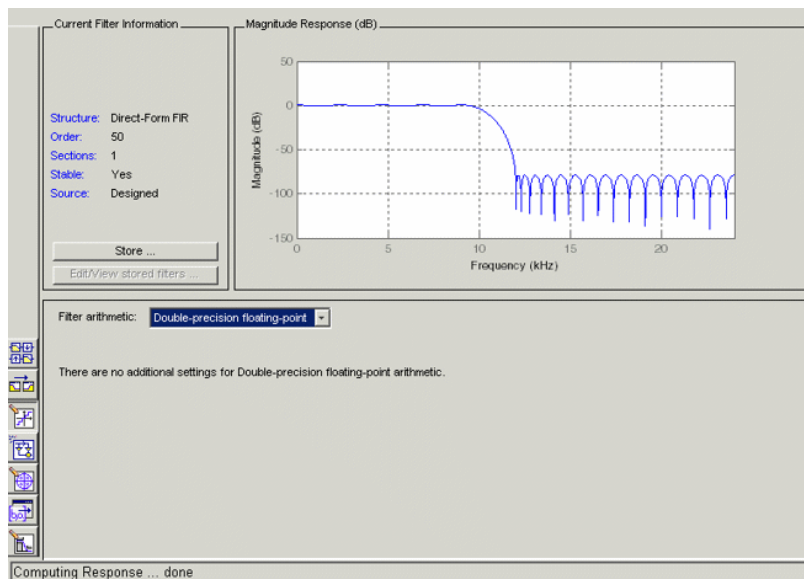
```
>>Doc iirnotch
>>Doc signal/ FDATool
```

B.10. CONMUTACIÓN FDATool AL MODO QUANTIZADO

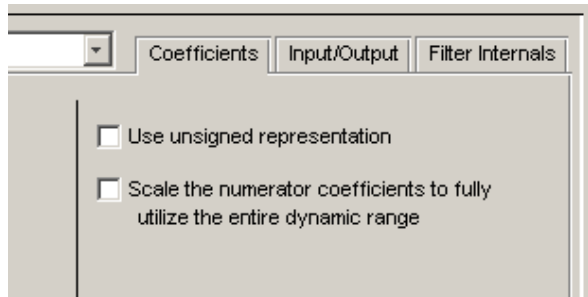
Se puede utilizar el quantizacion del FDAToll a filtros cuantizados. La cuantizacion representa el cuarto modo de funcionamiento para FDATool, junto con el diseño del filtro, transformación de filtro, y los modos de imputación.



Cuando se abre el FDATool, haga clic en el botón de cambios parámetros cuantizados en la barra lateral. FDATool cambia el modo de cuantización y aparece el siguiente panel:



La opción aritmética del filtro permite cuantizar los filtros e investigar los efectos de los cambios en la configuración de cuantización. Para permitir ajustes en la cuantización de FDATool, seleccione la aritmética de punto fijo del filtro.



Se puede usar las siguientes pestañas en el cuadro de diálogos para realizar y tareas relacionadas con la cuantización de filtros en el FDATool:

- coeficientes: prevé la configuración de acceso para definir los coeficientes. Este es el grupo activo por defecto cuando se usa el modo de cuantización de filtros de FDATool. Cuando importe un filtro punto fijo en FDATool, este es panel activado.
- Input/output: interruptores FDATool de las opciones de cuantización de entrada y salida de filtro.
- Filtros internos: permite establecer una variedad de opciones para el cálculo del filtro, como la forma en que el filtro se encarga de los resultados de las operaciones de multiplicación o de la forma en que el filtro utiliza el array.
- Aplicación: se aplica a los cambios que utiliza la cuantificación de los parámetros de su filtro.

B.11. CUANTIZACION DE FILTROS EN EL TOOL DE ANÁLISIS Y DISEÑO DE FILTROS

B.11.1 CONFIGURACIÓN DE LOS PARÁMETROS DE CUANTIFICACIÓN

La cuantificación de filtros tiene propiedades que define la forma en que se cuantizan los datos de un filtro.

Para establecer las propiedades en el FDATool se hace el cuadro de dialogo “Quantization parameters” , dicho dialogo permite el desarrollo de las siguientes tareas:

- Crear una cuantización para filtros de doble precisión después de haber sido importado del workspaces o utilizado el diseño de filtros de FDATool.
- Crea cuantización de un filtro de estructura forma directa II o cualquier estructura que se elija.
- Cambio de los valores de la propiedad de cuantización de un filtro.

Cuando se hace clic en “Quantization parameters”, y cambia la aritmética del filtro a punto fijo, se abre el panel de quantización del filtro, con las opciones de cambio de los valores de los coeficientes de cuantización.

Filter arithmetic: Fixed-point

Coefficients | Input/Output | Filter Internals

Coefficient word length: 16 ☐ Best-precision fraction lengths ☐ Use unsigned representation

☐ Numerator frac. length: 13 ☒ Scale Values frac. length: 14

☒ Numerator range (+/-): -2 ☐ Scale Values range (+/-): 1

☒ Denominator frac. length: 14

☐ Denominator range (+/-): 1

Apply

B.11.2 OPCIONES DEL COEFICIENTE

Para establecer las propiedades de los coeficientes del filtro que se utilizara a continuación del filtro, el FDATool tiene una lista de opciones para longitud de palabras del numerador (longitud de palabras del denominador si es un filtro IIR). La siguiente tabla muestra los coeficientes de cada opción y una breve descripción de lo que hace la opción de configuración del filtro.

OPCION DE NOMBRE	CUANDO SE USA	DESCRIPCION
Longitud de palabra del numerador	Solo filtros FIR	Establece la longitud de palabra utilizada para representar los coeficientes del numerador del filtro FIR.
Longitud de fraccion del numerador	FIR IIR	Establece la longitud de fraccion utilizada para interpretar los coeficientes del numerador de los filtros.
Rango del numerador (+/-)	FIR IIR	Permite definir el rango de los numeradores representados. Se puede utilizar la longitud de fraccion del numerador para ajustar la precision. Al introducir un valor x, el rango resultante es de $-x$ a x . el rango debe ser un numero

		entero positivo.
Coeficiente de longitud de palabra	Solo filtros FIR	Establece la longitud de palabra utilizada para representar tanto el numerador y denominador de los coeficientes del filtro IIR. No se puede establecer diferentes longitudes de palabras para el numerador y denominador.
Longitud de fraccion del denominador	IIR	Establece la longitud de fraccion del denominador utilizada para interpretar los coeficientes en los filtros IIR.
Rango de denominador (+/-)	IIR	Permite definir el rango de los denominadores representados. Se puede utilizar la longitud de fraccion del denominador para ajustar la precision. Al introducir un valor x, el rango resultante es de $-x$ a x . el rango debe ser un numero entero positivo.
Mejor precision de la longitud de fraccion	Todos los filtros	El FDATool selecciona el numerador de la longitud de fraccion (denominador cuando este disponible) valores para maximizar el rendimiento de fracciones del filtro.
Escala de valores de longitud de fraccion	Filtros IIR SOS	Establece la longitud en fraccion utilizada para interpretar la escalizacion de los valores de los filtros SOS.
Escala del rango (+/-)	Filtros IIR SOS	Permite definir el rango SOS de la escala de valores. Se puede utilizar un filtro SOS para ajustar el escalamiento. Al configurar estos valores se desactiva la longitud de fraccion variante de la escala de valores. Al introducir un valor x, el rango resultante es de $-x$ a x . el rango debe ser un numero entero positivo.
Uso de la representación	Todos los filtros	Indica que el FDATool

sin signo		interpreta los valores sin signo del coeficiente.
Escala del coeficiente del numerador a utilizar plenamente todo el rango dinamico	Todos los filtros	La escala del numerador en el FDATool utiliza eficazmente el rango dinamico definido por la longitud de palabra del numerador y el formato de longitud de fraccion.

B.11.3 OPCIONES DE INPUT/OUTPUT

La opción que especifica las opciones de cuantizacion del filtro usan valores se entrada y salida que figuran en el cuadro de la siguiente figura. En la siguiente imagen se puede ver las opciones para un filtro SOS.

The screenshot shows the 'Input/Output' configuration window for a filter. It has three tabs: 'Coefficients', 'Input/Output' (selected), and 'Filter Internals'. Under 'Filter arithmetic', 'Fixed-point' is selected. The settings are organized into three columns: Input, Output, and Stage. Each column has fields for 'word length', 'fraction length', and 'range (+/-)', along with a checkbox for overflow handling. The 'Apply' button is located at the bottom center.

OPCION DE NOMBRE	CUANDO SE USA	DESCRIPCION
Entrada de longitud de palabra.	Todos los filtros	Establece la longitud de palabra utilizada para representar la entrada de un filtro.
Entrada de longitud de fraccion.	Todos los filtros	Establece la fraccion de longitud utilizada para interpretar los valores de entrada al filtro.
Entrada de rango (+/-)	Todos los filtros	Permite definir el rango de entrada representada. Se puede utilizar esta entrada en lugar de la opcion de longitud de fraccion para fijar la precision. Al introducir un valor x, el resultado del rango es $-x$ a x . El rango debe ser un numero entero positivo.
Salida de longitud de palabra	Todos los filtros	Establece la longitud de palabra utilizada para representar la salida de datos.
Avoid del overflow	Todos los filtros	Encarga al filtro para establecer la longitud de fraccion de entradaEncarga al filtro para

		establecer la longitud de fraccion de entrada para impedir la salida de valores por encima del rango disponible, tal como se define por la longitud de palabra.
Salida de longitud de fraccion	Todos los filtros	Establece la longitud de fraccion usada para representar los valores de salida del filtro.
Rango de salida (+/-)	Todos los filtros	Permite definir el rango de salida representado. Al introducir un valor de x, el rango es de $-x$ a x . El rango debe ser un valor entero positivo.
Estado de la entrada de longitud de palabra	Solo filtros SOS	Establece la longitud de palabra utilizada para representar la entrada a un filtro de la seccion SOS.
Avoid del overflow	Solo filtros SOS	Permite al filtro utilizar una etapa de longitud de fraccion que impide el valro del overflow.
Estado de entrada de la longitud de fraccion	Solo filtros SOS	Establece la longitud de fraccion utilizada para representar la entrada del filtro SOS.
Estado de salida de la longitud de palabra	Solo filtros SOS	Establece la longitud de palabra utilizada para representar la salida de un filtro SOS
Avoid overflow	Solo filtros SOS	Permite utilizar la longitud de fraccion que impide el overflow en la salida del filtro.
Estado de salida de la longitud de fraccion	Solo filtros SOS	Establece la longitud de fraccion utilizada para la salida del filtro SOS.

B.11.4 OPCIONES INTERNAS DEL FILTRO

Las opciones que especifican la forma en que la cuantizacion del filtro realiza las operaciones aritméticas que se enumeren en la siguiente tabla.

Filter arithmetic: Fixed-point Coefficients Input/Output Filter Internals

Round towards: Nearest (convergent) Overflow Mode: Wrap

Product mode: Full precision Accum. mode: Keep MSB State word length: 16

Product word length: 32 Accum. word length: 40

Num. fraction length: 29 Num. fraction length: 29 ☐ Avoid overflow

Den. fraction length: 29 Den. fraction length: 29 State fraction length: 15

☒ Cast signals before accum.

Apply

OPCION	PROPIEDAD EQUIVALENTE DEL FILTRO	DESCRIPCION
Modo de redondeo	<i>Roundmode</i>	Establece el modo del filtro para la cuantizacion del valor cuando se encuentra entre valores representables para el formato de datos (longitud de palabra y longitud de fraccion). <i>Ceiling</i>: redondea al valor mas cercano hacia la derecha de la cuantizacion. <i>Floor</i>: redondea al mas cercano hacia la izquierda valor de la cuantizacion. <i>Nearest</i>: redondea al valor mas cercano de la cuantizacion. Los numeros que estan en la mitad de los dos valores mas cercanos admisibles de la cuantizacion se redodean. <i>Nearest (convergent)</i>: redondes al valor admisible siguiente de la cuantizacion. Para los valores que estan enla mitad del rango, redondea hacia la derecha al valor mas cercano, sólo cuando el bit menos significativo antes del redondeo es 1. <i>Round</i>: redondea hacia el valor mas cercano. redondea hacia la derecha del valor absoluto de la cuantizacion.

		Zero: redondea los numeros negativos y positivos hacia el cero mas cercano.
Modo overflow	<i>OverflowMode</i>	Establece el modo a usar para responder las condiciones de overflow en la aritmetica de putno fijo. Se puede elegir <i>saturate</i> (limite de salida del valor negativo mayor o positivo mayor) o se puede elegir <i>wrap</i> (<i>overflow</i> de los valores establecidos para el valor mas cercano representable usando la aritmetica anular).
OPCIONES DEL PRODUCTO DE FILTROS (MULTIPLICACION)		
Modo del producto	<i>ProductMode</i>	Determina la manera en que el filtro se encarga de las operaciones de salida del producto. Se puede elegir precision completa (<i>FullPrecision</i>), con el bit mas significativo (<i>KeepMSB</i>), con el bit mas significativo (<i>KeepLSB</i>).
Producto de la longitud de palabra	<i>ProductWordLength</i>	Establece la longitud de palabra que se aplica para interpretar los resultados de las operaciones de la multiplizacion.
Longitud de fracción del numerador	<i>NumProdFracLength</i>	Establece la longitud de fraccion utilizada para interpretar los resultados de las operaciones de productos que involucran los coeficientes del numerador.
Longitud de fraccion del denominador	<i>DenProducFracLength</i>	Establece la longitud de fraccion utilizada para interpretar los resultados de las operaciones de productos que involucren los coeficientes del denominador.
OOPCIONES DE ADICION DEL FILTRO (SUMA)		
Modo acumulador	<i>AccuMode</i>	Determina la forma en que se acumulan los valores almacenados de la suma en la salida.
Longitud de palabra del acumulador	<i>AcumWordLength</i>	Establece la longitud de palabra que se utiliza para almacenar los datos en el acumulador/ <i>buffer</i> .
Longitud de fraccion del numerador	<i>NumAcumFracLength</i>	Establece la longitud de fraccion usados para interpretar los coeficientes del numerador.

Longitud de fraccion del denominador	<i>DenAcumFracLength</i>	Establece la longitud de fraccion que el filtro usa para interpretar los coeficientes del denominador.
<i>Cast</i> antes del signo de la suma	<i>CastBeforeSum</i>	Especifica si los datos numericos emiten el formato adecuado del acumulador antes de llevar a cabo las operaciones de suma.
OPCIONES DEL ESTADO DEL FILTRO		
Longitud de palabra del estado del filtro	<i>stateWordLength</i>	Establece la longitud de palabra utilizada para representar los estados del filtro. Aplicando a la vez los estados relacionados al numerador y al denominador.
<i>Avoid overflow</i>	Ninguno	Previene el <i>overflow</i> en los calculos aritmeticos mediante el establecimiento de la longitud de fraccion adecuada.
Longitud de fraccion del estado del filtro	StateFracLength	Permite definir la longitud de fraccion aplicada a la interpretacion de los estados del filtro. Aplicados a su vez a la vez a la relacion de los estados del numerador y denominador.

B.11.5 EJEMPLO – CUANTIZAR FILTROS DE DOBLE PRECISIÓN

Cuando se cuantiza un filtro de doble precisión, cambiando a punto fijo o aritmética de punto fijo de única precisión, hay que seguir los siguientes pasos:

1. haga clic en parámetros de cuantización para mostrar el panel de parámetros de cuantización en el FDATool.
2. seleccione precisión – single punto flotante o punto fijo para filtros aritmética cuando se selecciona una de las opciones de la aritmética el FDATool cuantiza el filtro actual de acuerdo a los valores de las opciones del panel de parámetros de cuantización y los cambios son informados en el área de análisis que muestra los datos cambiados del filtro cuantizado.
3. establecer la opción del filtro, las opciones para los coeficientes, Entrada y salida de fluidos internos
4. clic en aplicar.

B.11.6 EJEMPLO: CAMBIO DE PROPIEDADES DE UN FILTRO CUANTIZADO

La configuración de cuantización de un filtro cuantizado o después de o de pues de haber importado el filtro, hay que seguir los siguientes pasos:

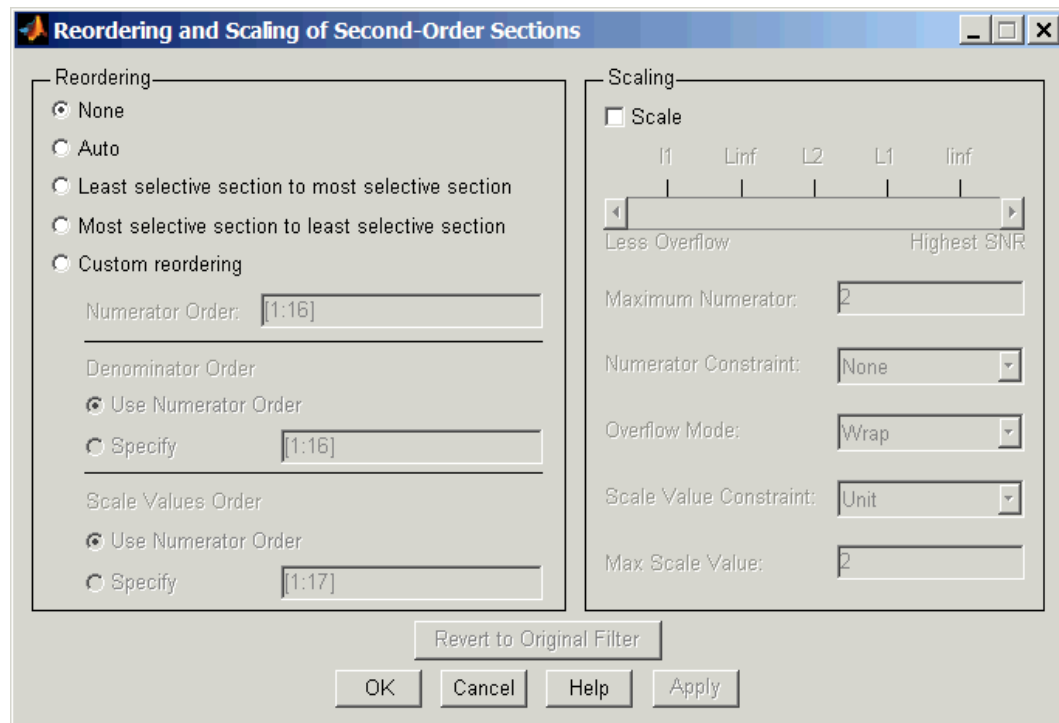
1. Verifique que el filtro actual este cuantizado.

2. Haga click en los perímetros de cuantización para ver en el panel de parámetros de cuantización
3. Revise y seleccione y las propiedades que se quieran cambiar: Coeficientes, entrada/salida, filtros internos.
4. Haga clic en aplicar

B.12. ESCALIZACION DE UN FILTRO DE SECCION DE SEGUNDO ORDEN (SOS)

B.12.1 USO DEL REORDENAMIENTO Y DE LA ESCALIZACION

El cuadro de dialogo de la sección de segundo orden (SOS) del FDATool proporciona capacidad de escala de un filtro SOS después de haberlo creado. Usando las opciones de reordenamiento y escalamiento del cuadro de dialogo de la sección de segundo-orden. El FDATool escalización uno o ambos numeradores del filtro y el valor de escalización del filtro acuerdan las opciones para la opción de escalización.



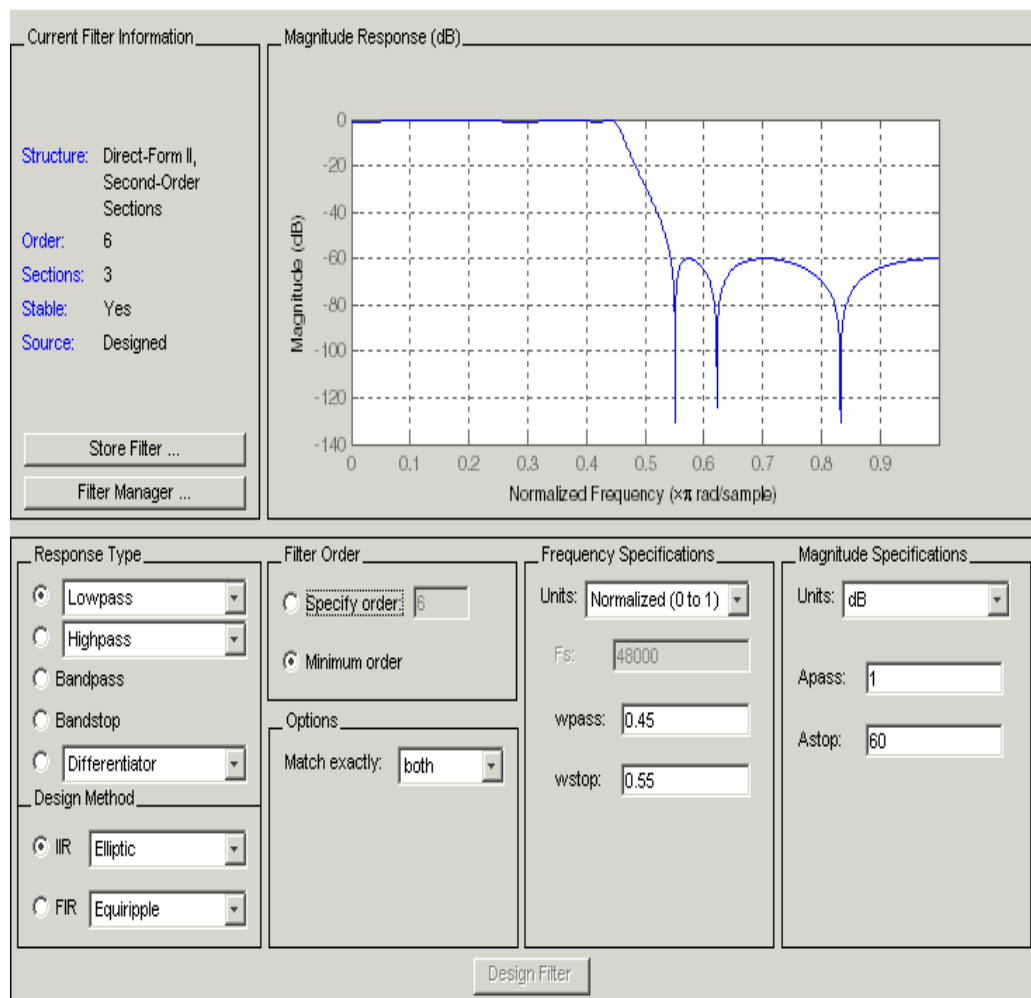
PARAMETROS	DESCRIPCION
Escala	Aplica cualquier opción que tiene el filtro para realizar la escalización. Seleccione esto cuando se reordena el filtro SOS y se desea escalizar el tiempo o cuando se escaliz el filtro con o sin reordenamiento. Al escalización es

	desactivada por defecto.
Sin <i>overflow</i> . <i>High SNR slider</i>	Permite definir si la escalización favorece la reducción aritmética del <i>overflow</i> en el filtro o la maximización de la relación señal/ruido (SNR) en la salida del filtro. Al mover el <i>Slider</i> hacia la derecha se incrementa el énfasis en el SNR a expensas del posible <i>overflow</i> . Los marcos indican la norma_p aplicada para lograr el resultado deseado en el SNR o protección del <i>overflow</i> .
Maximo numerador	Valor máximo permitido para los coeficientes del numerador después de la escalización.
Constante del numerador	Especifica si y como se limita los valores del coeficiente del numerador. Las opciones son: ninguna, normalizado, potencia de dos y unidad. Ninguno: permite la escalización usando cualquier valor de escala para el numerador removiendo solo las constantes del numerador, excepto que los coeficientes pueden ser cortados si exceden el numerador máximo. Normalizada: el valor absoluto máximo del numerador es forzado a la igualdad del valor máximo del numerador (para otras constantes, el numerador máximo es solo sobre el límite superior, el coeficiente puede ser limitado). Potencia de dos: utiliza la escalización para usar los valores del numerador que es de potencia de 2 o 0.5. Unidad: el principal coeficiente de cada numerador se ve forzado a un valor de 1.
Modo de <i>overflow</i>	Establece la forma en que el filtro se ocupa de la aritmética del <i>overflow</i> durante la escalización. Se puede elegir <i>saturate</i> (límite de salida en el positivo o negativo mayor del valor representable) o <i>wrap</i> (<i>overflow</i> de los valores para representación del valor cercano usando la aritmética modular).
Escala del valor constante	Especifica si el valor de la escala del filtro es constante y como limitarlo. Las opciones válidas son unidad, potencia de dos o ninguno. Unidad: la limitante máxima está deshabilitada. El valor de la escala debe ser 1. Potencia de dos. Limita la escala de valores como 2 o 0.5 Ninguno: elimina cualquier limitación en la escala de valores con la excepción que no puede exceder al máximo valor de la escala.
Valor máximo de la escala	Fija la escala máxima permitida. La escalización del filtro SOS aplica al valor máximo de escala y limita solo con el valor limitante de la escala a otros valores distintos a la unidad (configuración por defecto).
Regresar al filtro original	Devuelve al filtro la escalización original. La posibilidad de volver a utilizar el filtro hace que sea más fácil evaluar los

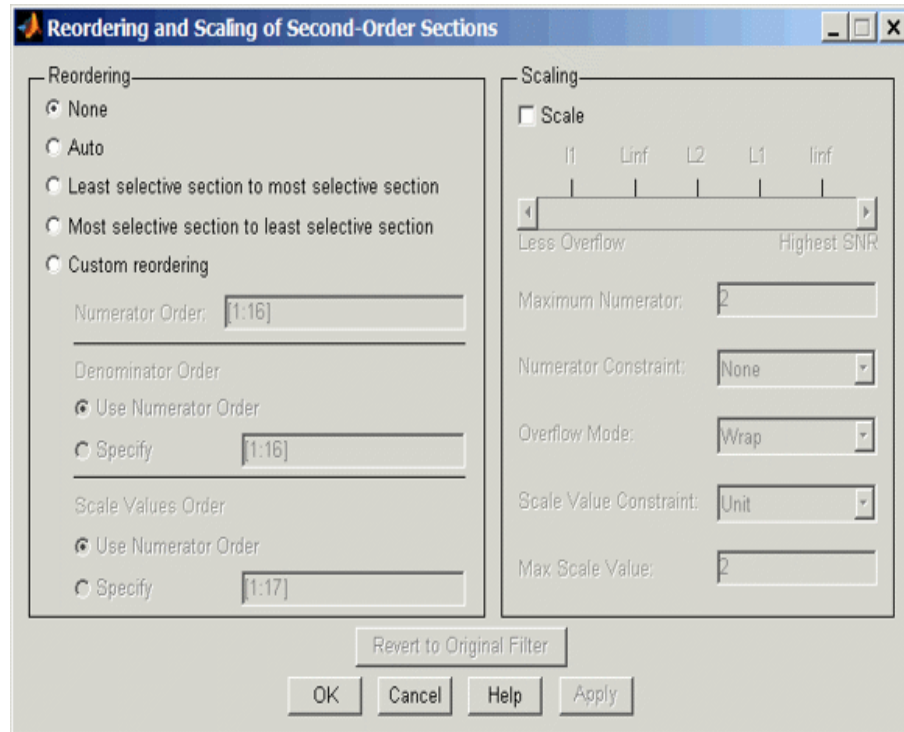
resultador de escalizacion.

B.12.2 EJEMPLO DE LA ESCALA DE UN FILTRO SOS:

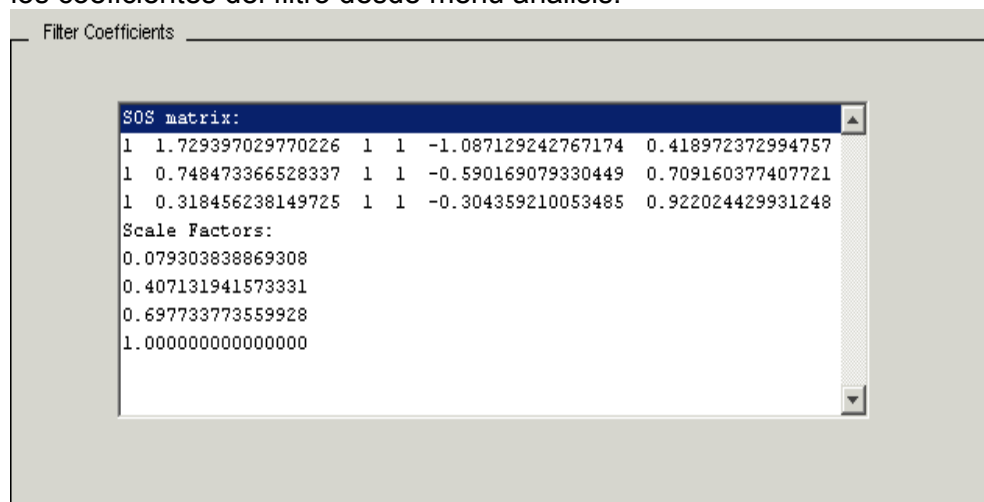
1. Iniciar el proceso de un diseño de un filtro pasabajo.
2. Lanzamiento (lounch) FDAToll
3. Tipo de respuesta, seleccionar pasabaja
4. Método de diseño, seleccione IIR y formar elíptica de la lista del método de diseño.
5. Seleccionar la orden mínima del filtro
6. Cambiar la unidad de frecuencias mediante la elección normalizada (o a1) de las unidades de lista.
7. Para definir el pasabanda entrar 0,45 como wpass y 0.50 como stop y 60 en Astop.
8. Haga click en el diseño de filtros para diseñar el filtro.



9. Para cambiar la escalización del filtro, seleccione edit, seleccione en barra del manu Rearder and scale second-order sections.
Su selección abre el cuadro de dialogo de reordering and scaling secand-order sections.

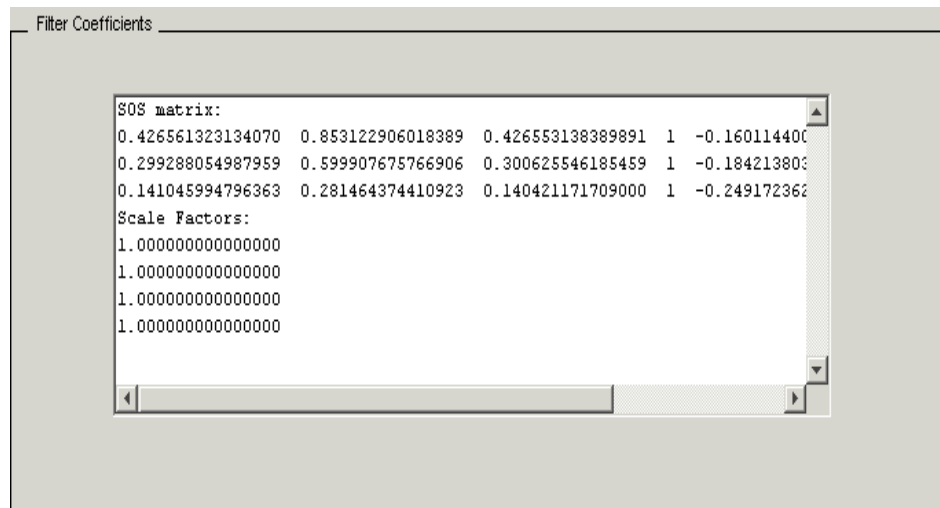


10. Para ver los coeficientes del filtro retornando al FDATool y seleccionar los coeficientes del filtro desde menú análisis.



Ahora escalizando el filtro de diferentes formas. Primera escala del filtro a máximo de SNR.

1. Retorna al cuadro de dialogo del reording and scaling second order sections, seleccione la opción none reordeninig en el panel izquierdo.
2. Mueva el numero de overflow-high SNR slider desde el numero de overflow to high SNR.
3. Haga click en aplicar la escala del filtro y salga del cuadro de dialogo. Luego el FDAToll muestra los coeficientes.



Todos los factores de la escala son 1, y la matriz de coeficientes SOS significan que ninguno de los coeficientes del numerador es 1 y el primer coeficiente del denominador es 1.

4. Haga click en volver al filtro original para establecer los parámetros de socialización y los coeficientes.

B.13 REORDENACION DE UN FILTRO DE SECCION DE SEGUNDO ORDEN

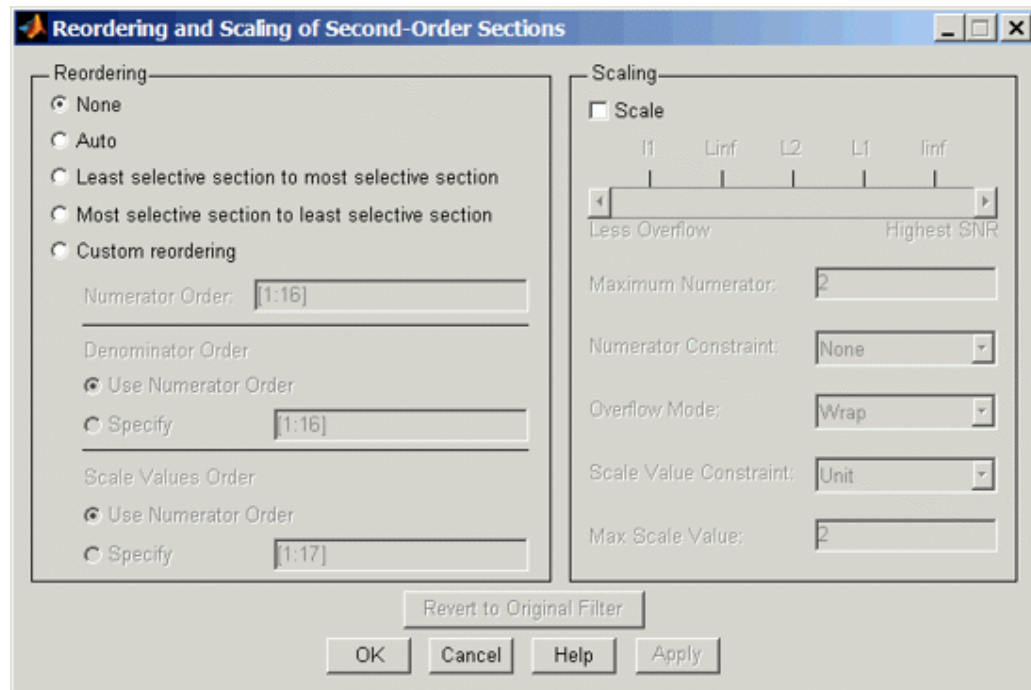
B.13.1 CAMBIO DE FDATool A REORDENAR EL FILTRO

En el diseño de FDATool de filtro de tiempo discreto en secciones de segundo orden. Generalmente los filtros SOS resistentes los cambios del efecto de cuantizacion cuando se era un filtro de punto fijo.

Para reordenar las secciones de un filtro se debe acceder al cuadro de diálogos del reorder and scaling of second-order sections del FDATool, seleccione edit y en la barra del menú reorder and scale.

El FDA Toll devuelve el reordenamiento según el cuadro de dialogo.

B.13.2 CUADRO DE DIALOGO DE LOS CONTROLES SOBRE EL REORDENAMIENTO Y ESCALIZACION DE LAS SECCIONES DE SEGUINDO ORDEN.



En este cuadro de dialogo, el lado izquierdo contiene opciones para reordenar filtros SOS. A la derecha se puede ver las opciones de escalizacion. Al reordenar el filtro no se requiere escalizacion y cuando se escala no se requiere reordenar. La siguiente tabla muestra cada opción de reordenar y proporcionar una descripción de lo que hace la opción.

OPCION DE CONTROL	DESCRIPCION
Auto	Reordena las secciones del filtro para minimizar la potencia del ruido de la salida del filtro. Notese que las aplicaciones diferentes de la ordenanza se aplica a cada tipo de cuantizacion como para filtros <i>lowpass</i> y <i>highpass</i> .
ninguna	La seccion ninguna permite ajustar la escala del filtro sin aplicar el reordenamiento al mismo tiempo.
Seccion enos	Reorganiza las secciones de modo que el filtro menos

selectiva a la seccion mas selectiva	restrictivo (<i>lowest Q</i>) de seccion es la primera seccion y la seccion mas restrictivo (<i>highest Q</i>) es la ultima seccion.
Seccion mas selectiva a la seccion menos selectiva	Reorganiza el filtro. La seccion mas selectiva es la primera seccion y la seccion menos selectiva es la ultima seccion.
Uso del reordenamiento	Permite especificar la seccion de reordenamiento a usar al habilitar el orden del numerador y la opcion de orden del denominador.
Orden del numerador	Especifica nuevos ordenamientos para las secciones del filtro SOS. Introduzca un vector de indices de secciones en el orden que desee. Por ejemplo, un filtro con cinco secciones tiene un indice 1, 2, 3, 4 y 5. para pasar de la segunda a la cuarta seccion seria un vector [1, 4, 3, 2, 5].
Uso del orden del numerador	Reorganiza los denominadores en el orden asignado a los numeradores.
Especificación	Permite especificar el orden de los denominadores, en lugar de utilizar el orden del numerador. Introduzca un vector de indices de secciones para especificar el orden de los denominadores a usar. Por ejemplo, un filtro con cinco secciones tiene un indice 1, 2, 3, 4 y 5. para pasar de la segunda a la cuarta seccion seria un vector [1, 4, 3, 2, 5].
Uso del orden del numerador	Reordena la escala de valores según el orden de los numeradores.
Especificación	Permite especificar el orden de la escala de valores, en lugar de utilizar el orden del numerador. Introduzca un vector de indices de secciones para especificar el orden de los denominadores a usar. Por ejemplo, un filtro con cinco secciones tiene un indice 1, 2, 3, 4 y 5. para pasar de la segunda a la cuarta seccion seria un vector [1, 4, 3, 2, 5].
Regresar al filtro original	Devuelve el filtro a la ordenanza de la seccion original.

B.13.3 EJEMPLO: REORDENAR UN FILTRO SOS

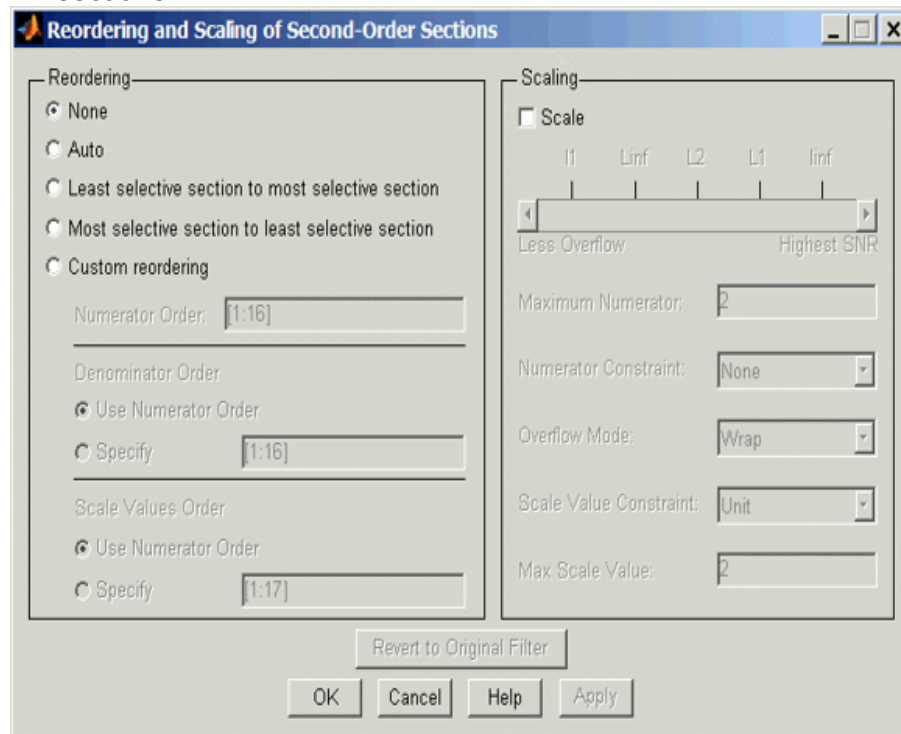
Con FDATOOL y un filtro de actual, se une el signo procedo para acceder ala capacidad del coordinamiento y reordenar el filtro

1. ENTRAR EL FDATOOL
2. Diseñe un filtro butterworth pasa bajo de orden 10 y la especificación de la frecuencia predeterminada introduciendo los siguientes valores. Tipo de respuesta, seleccione lowpass. Método de diseño. Seleccione IIR y Butherworth

Especifique el orden igual a 10. Valores de F_s y F_e en especificaciones de frecuencia.

4. clic en diseño de filtros.

5. para reordenar el filtro, seleccione edit y en reorder y sale second-order sections.

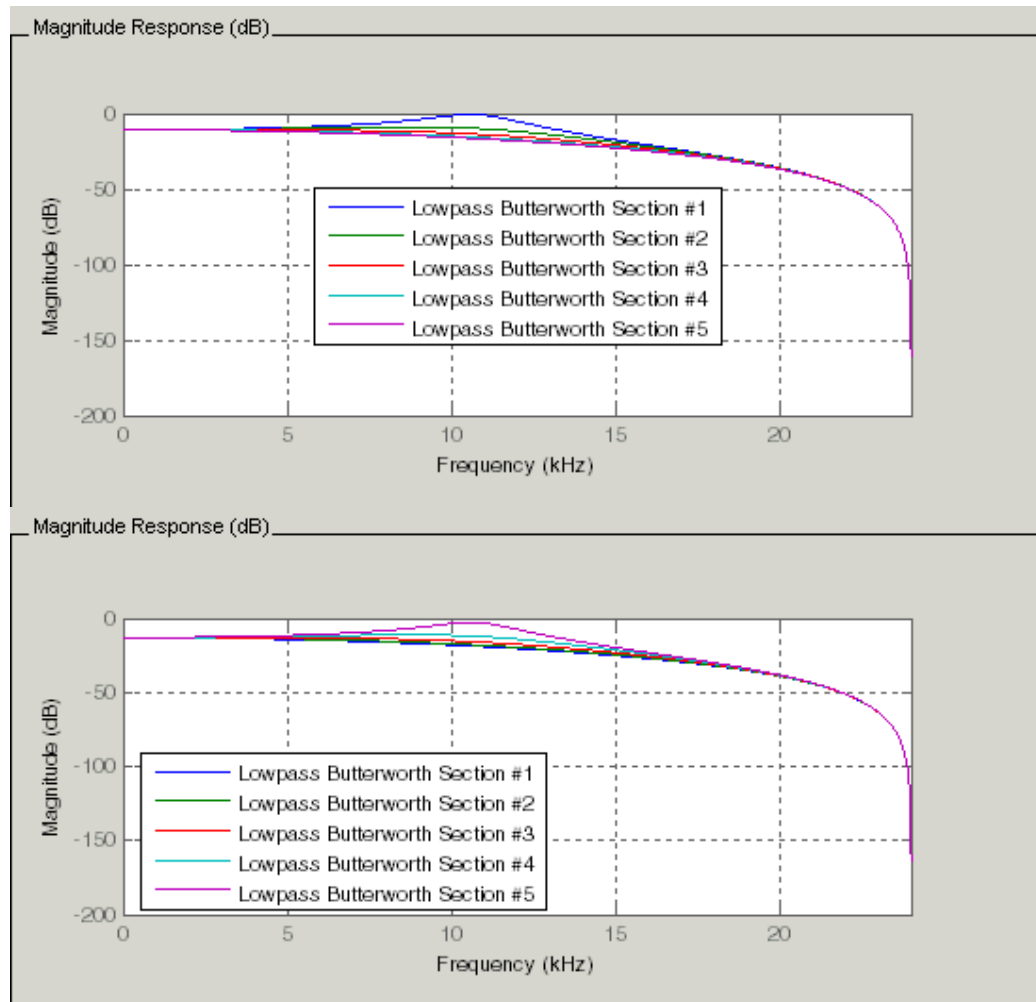


B.13.4 USO DEL MENOS SELECTIVO AL MAS SELECTIVO DE LA SECCIÓN DE REORDENAMIENTO

Para que FDATool reordene su filtro del menos selectivo al mas selectivo, lleve acabo los siguientes pasos en la escalizacion y ordenamiento de segundo orden de sección del cuadro de dialogo.

1. en el ordenamiento, seleccione la sección menos selectiva a la sección mas significativa.
2. para prevenir la escalizacion del filtro al mismo tiempo, limpiar la escala en escalizacion.
3. en FDATool,. Seleccione VIEW > sos VIEW
4. en el cuadro de dialogo del sos view seleccione section individual.
5. de aceptar en el cuadro de dialogo de Saling or Second order sections.

En los 2 siguiente se puede comparar la ordenanza de las secciones del filtro. En la primera figura esta el filtro original y en la segunda esta el filtro ordenado.



B.14 VISUALIZACIÓN DE LAS SECCIONES DEL FILTRO SOS

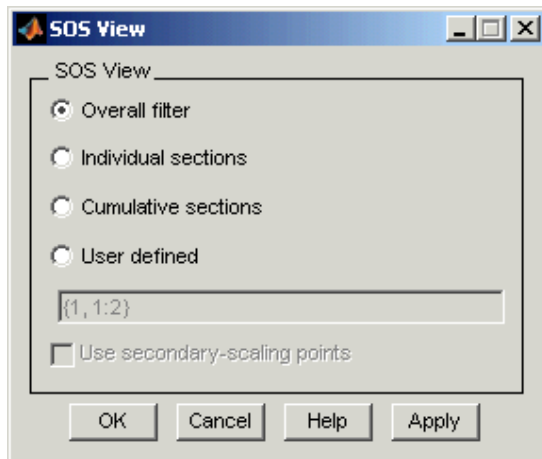
B.14.1 USO DEL CUADRO DE DIALOGO DE LA DESCRIPCIÓN SOS

Debido a que se puede diseñar y reordenar las secciones de un filtro sos, FDATool ofrece la posibilidad de ver secciones del filtro en la sección de análisis.

B.14.2 CONTROLES DEL CUADRO DE DIALOGO DE LA OPCIÓN SOS VIEW

SOS view utiliza varias opciones para controlar el modo de las secciones que muestra el FDATool.

Cuando se selecciona `view >sos` muestra el siguiente cuadro.



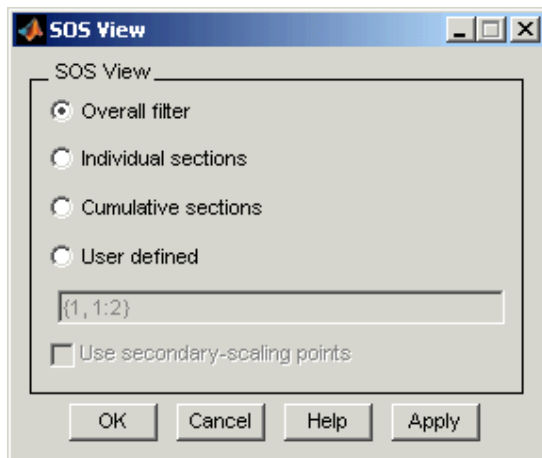
Estas opciones permiten cambiar la pantalla. La siguiente tabla incluye todas las opciones y describe los efectos de cada uno.

OPCION	DESCRIPCION
Filtros en general	Esta es la pantalla general del <i>FDATool</i> . Para un filtro de seccion de segundo orden. Se ve la respuesta general en lugar de las respuestas de cada seccion. Esta es la configuración por defecto.
Secciones individuales	Al seleccionar esta opcion el <i>FDATool</i> muestra la repuesta para cada seccion como una curva. Si el filtro tienen cinco secciones se verán cinco curvas de respuesta, una para cada seccion. Compara el acumulativo de las secciones.
Secciones acumulativas	Al seleccionar esta opcion, el <i>FDATool</i> muestra la respuesta de cada una de las secciones y la respuesta acumulada de todas las secciones del filtro. Para el filtro de cinco secciones, la primera curva la respuesta para la primera seccion del filtro. La segunda curva es la respuesta de la primera y segunda seccion del filtro. La tercera curva es la respuesta a la primera, segunda y tercera seccion combinada. Y así hasta que todas las secciones del filtro aparezcan en la grafica. La ultima curva representa la respuesta del filtro.
Defindo por el usuario	Aquí se puede definir que secciones mostrar y en que orden. Al seleccionar esta seccion se abre un cuadro de dialogo y en la celda que aparece se coloca el numero de secciones del filtro. Por ejemplo, para un filtro de dos secciones se escribe 1:2 y para un filtro se 4 secciones 1:4. Tenga en cuenta que al hacer eso se debe dar clic en OK o en aceptar para guardar los cambios.
Uso de puntos secundarios escalizados	Utiliza los puntos secundarios escalizados para determinar la admisión de las secciones. Solo se aplica cuando el filtro es df2sos o df1tsos.

B.14.3 EJEMPLO PARA VER LA SECCIÓN DE LOS FILTROS

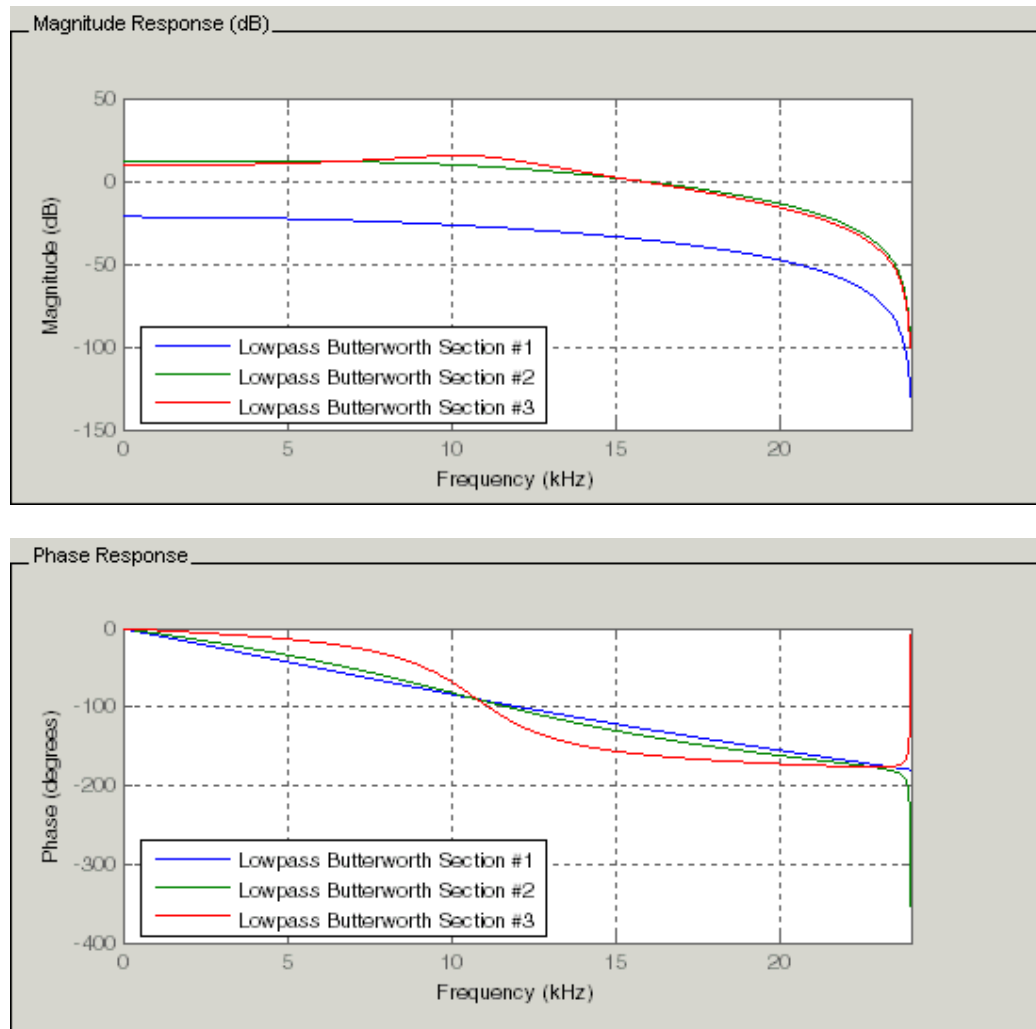
Después de diseñar o importar el filtro en el FDATool. La opción permite ver el rendimiento de cada sección del filtro.

1. correr el FDATool.
2. crear un filtro pasabanda sos butterworth.
3. diseño del nuevo filtro, haciendo clic en diseño de filtro. El FDATool diseña el filtro y muestra en el área de análisis la respuesta en magnitud.
4. habilitar el view sos, donde se cambian las opciones de especificación de cómo mostrar las secciones del filtro.



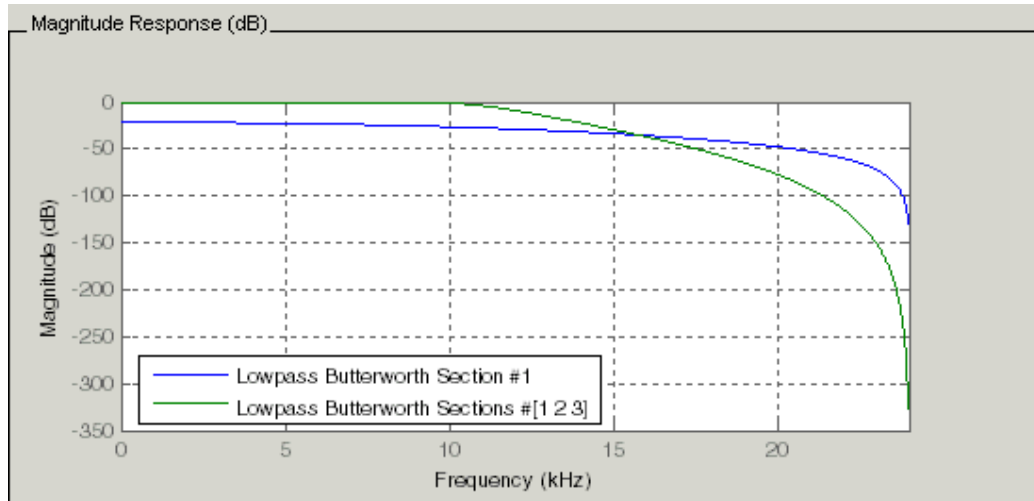
Por defecto, el área de análisis del FDATool muestra la respuesta del filtro global y no la respuesta del filtro en secciones. Este cuadro de dialogo permite cambiar las configuraciones de pantalla para ver las secciones.

5. para ver la respuesta en magnitud de cada sección, seleccione distintas secciones.
6. haga clic en aplicar, para actualizar el FDATool para mostrar la respuesta de cada sección del filtro. El área de análisis muestra lo siguiente:



7. para definir las propiedades de las secciones de la pantalla, se puede usar las opciones de usuario definidos y entrar un vector de la sección para mostrar los índices.
 En la pantalla se puede ver la respuesta de la primera sección, y el comunicado de la primera , segunda y tercera respuesta:
 - seleccione el uso definido para habilitar el cuadro de texto, en el cuadro de dialogo.
 - Introduzca la matriz (1,1:3) para especificar que el FDATool muestre la respuesta de la primera sección y la respuesta acumulativa de la primera hasta la tercera sección del filtro.
8. para aplicar una nueva selección de view sos haga clic en aplicar o en OK (cierra el cuadro de dialogo de view sos). En el área de análisis del FDATool

se ven dos curvas una para la primera sección de la respuesta y la otra para la respuesta combinada de las secciones 1,2 y 3.



B.15 IMPORTAR Y EXPORTAR UN FILTRO CUANTIZADO

B.15.1 DESCRIPCIÓN Y ESTRUCTURA

Cuando se importa o exporta un filtro cuantizado al workspace, las funciones de importar y exportar usa objetos y especifica el filtro como una variable.

Esto contrasta con la importancia y exportación de filtros no cuantizados, en la que seleccionan la estructura del filtro y se escribe la función de transferencia del numerador y el denominador del filtro.

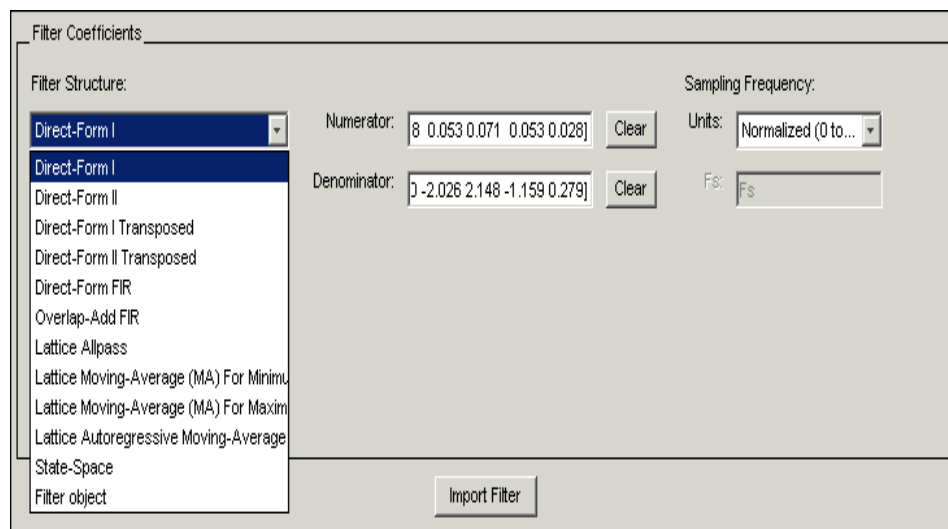
Además se exporta filtros cuantizados al workspace de matlab, se puede exportar archivos de texto o exportar archivos. MAT se puede importar un filtro cuantizado con la siguientes estructura.

- Directa forma I
- Forma directa II
- Forma directa I transpuesta
- Forma directa II I transpuesta
- Forma directa simétrica FIR
- Forma directa asimétrica FIR
- Lattice todopass o pasatodo
- Lattice AR
- Lattice fase minimaza
- Lattice fase máxima MA
- Lattice ARMA
- Lattice coupled – pasatodo
- Lattice coupled – pasatodo potencia complementarias.

B.15.2 EJEMPLO. IMPORTAR FILTRO CUANTIZADO

Después de diseñar el filtro y cuantizarlo se puede importar el filtro siga los siguientes pasos.

1. abrir el FDATool
2. seleccionar filtro> importar filtro. En la parte inferior del FDATool, el panel de diseño de filtros se convierte en importación del filtro y aparece con las opciones para importar un filtro cuantizado.



3. de la lista de la estructura del filtro, seleccione objeto filtro. Las opciones para importar filtro incluye los siguientes cambios:
 - filtros discretos: escríble nombre de la variable para tiempo discreto, filtro punto fijo en el workspace.
 - Unidad de frecuencia: de la lista de unidades, seleccione la unidad de frecuencia para la frecuencia de muestreo y especificación el valor de la frecuencia de muestreo F_s si es necesario.
4. haga clic en importar para importar el filtro.

B.15.3 PARA EXPORTAR FILTROS CUANTIZADOS

Al guardar el filtro, FDATool le permite exportar el filtro cuantizado, se puede seleccionar una de estas secciones:

- exportar al workspace de Matlab
- exportar a un archivo de texto
- exportar aun archivo MAT

B.15.4 EJEMPLO. EXPORTAR COEFICIENTES U OBJETOS AL WORKSPACE

Para guardar el filtro como variable de coeficientes del filtro o como variable de objeto dfilt del filtro se hace lo siguiente:

1. seleccionar exportar en le menú archivo y abrirá un cuadro de dialogo.
2. seleccione de la lista workspace para exportar
3. seleccione los coeficientes para exportar AS para guardar el filtro de los coeficientes.
4. para coeficientes, asignar el nombre de las variables usando las opciones del numerador y el denominador. Para objetos, asignar el nombre de la variable en la opción del filtro discreto o cuantizado.
5. haga clic en el botón ok.

Si se intenta exportar el filtro a un nombre de variable que existe en el workspace y no se selecciona sobre escribir variables, el FDATool detiene la operación de exportación y devuelve una advertencia de que la variable que s explica el nombre del filtro cuantizado existe en el workspace.

Para descartar este cuadro de advertencia haga clic en aceptar y seleccione sobrescribir variables y haga clic en aceptar.

B.15.5 OBTENCIÓN DE LOS COEFICIENTES DEL FILTRO DESPUÉS DE LA EXPORTACIÓN

Para extraer los coeficientes del filtro cuantizado después de exportar el filtro a Matlab, se usa la fracción celldisp.

Por ejemplo crea un filtro cuantizado en FDATool y exporte filtro como Hq, para extraer los coeficientes del filtro para Hq use.

```
>>cell disk (Hq referente coefficients)
```

Que devuelve la matriz que contiene los coeficientes del filtro de referencia.

```
>>celldisp (Hq quiantizedcoefficients)
```

Devuelve los coeficientes de cuantizacion.

B.15.6 EJEMPLO EXPORTAR COMO ARCHIVO DE TEXTO

Para guardar un filtro cuantizado como archivo de texto, siga los siguientes pasos:

1. seleccionar exportar en el menú de archivos.
2. seleccionar texto –archivo para exportar.

3. haga clic en aceptar para exportar el filtro y crear el cuadro de dialogo.
Haga clic en aplicar para exportar el filtro sin crear el cuadro de dialogo.
La exportación de coeficientes de filtros a archivos de texto abre un cuadro de dialogo.
4. seleccionar o introduzca un directorio y nombre del archivo para el archivo de texto y haga clic en aceptar.
El FDATool exportar el filtro cuantizado como un archivo de texto con el nombre proporcionado y el editor de matlab lo abre y muestra el archivo para editar.

B.15.7 EJEMPLO. EXPORTAR COMO ARCHIVO MAT

Para guardar el filtro cuantizado como archivo Mat, siga los siguientes pasos:

1. seleccione exportar en el menú Archivo
2. seleccione archivo- Mat para exportar
3. haga clic en aceptar para el filtro y cerrar el cuadro de dialogo.
Haga clic en aplicar para exportar el filtro sin cerrar el cuadro de dialogo.
La exportación de coeficientes del filtro a archivo Mat abre un cuadro de dialogo.
4. seleccione o introduzca un directorio y un nombre del archivo Mat y haga clic en aceptar.
El FDATool exporta el filtro cuantizado como archivo MAT con el nombre proporcionado y el editor de Matlab la abre y muestra el archivo.

B. 16 IMPORTAR EL ARCHIVO DE COEFICIENTE (.COE) XILINX

B.16.1 EJEMPLO. IMPORTAR ARCHIVOS XILINX.COE

Se puede importar xilinx coeficientes (.coe) archivos dentro del FDATool para crear directamente filtros cuantizados usando los coeficientes del filtro importado. Para usar el archivo importado:

1. seleccione archivos > importar filtro desde archivo de coeficientes (.coe) xilinx del FDATool.
2. en el cuadro de dialogo de archivo importado desde xilinx coeficientes (.coe), buscar y seleccionar el archivo .coe para importar.
3. abra clic en abrir para descartar el cuadro de dialogo e iniciar el proceso de importación.
El FDATool importar el archivo de coeficientes y crea un filtro FIR de forma directa de una sola sección y cuantizada.

B.17 TRANSFORMACIONES DEL FILTRO

B.17.1 CAPACIDADES DE LA TRANSFORMACIÓN DE UN FILTRO EN FDATAOL

El Toolbox proporciona funciones para la transformación de un filtro de distintas formas. Al seleccionar transformations, se puede transformar filtros FIR pasabaja y filtros IIR a una variedad de forma de pasabanda.

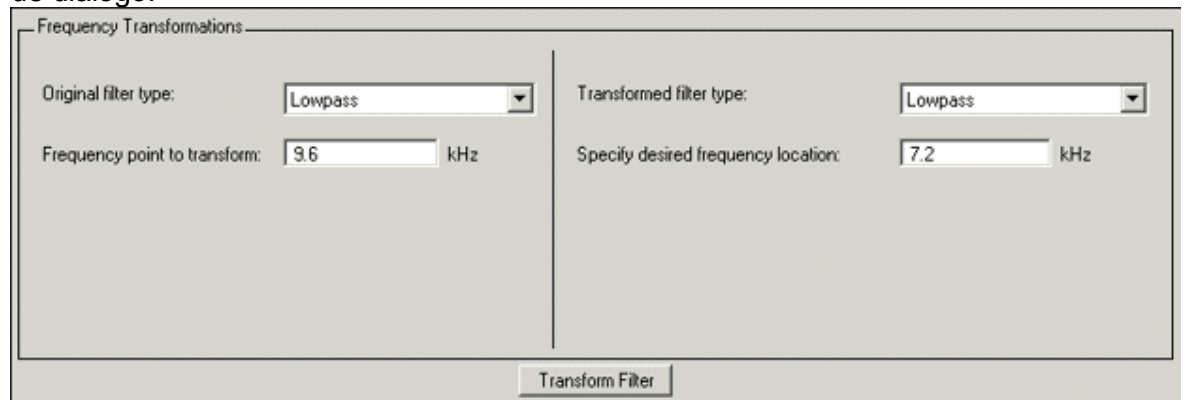
Se puede convertir los filtros FIR en:

- pasabaja a pasabaja
- pasabaja a pasaalto

Se puede convertir los filtros IIR en:

- pasabaja a pasabaja
- pasabaja a pasaalto
- pasabaja a pasabanda
- pasabaja a bandstop

Cuando se da clic en el botón de transformar filtros, se abre el siguiente cuadro de dialogo:



The dialog box titled "Frequency Transformations" is divided into two main sections. The left section contains "Original filter type:" with a dropdown menu showing "Lowpass" and "Frequency point to transform:" with a text input field containing "9.6" and "kHz". The right section contains "Transformed filter type:" with a dropdown menu showing "Lowpass" and "Specify desired frequency location:" with a text input field containing "7.2" and "kHz". At the bottom center is a button labeled "Transform Filter".

Las opciones del filtro original se refieren a la configuración actual del filtro a transformar.

Si selecciona lowpass se puede transformar el filtro a lowpass u otros formatos de filtros.

Nota: cuando un filtro original es un filtro FIR, tanto el filtro transformado FIR e IIR aparecen en la lista de transformación del filtro. Ambas opciones permanecen activos por que se pueden aplicar IIR a la transformación del filtro FIR. Si la fuente es el filtro FIR solo aparece el filtro IIR.

B.17.2 TIPOS DE FILTROS ORIGINALES

selecciona la respuesta en magnitud del filtro que esta en la lista. Seleccione los cambios de los tipos de filtros a transformarse. Por ejemplo, cuando se seleccione lowpass con un filtro IIR, los filtros a transformar pueden ser:

Pasabaja
 Pasaalto
 Pasabanda
 Bandstop
 Multibanda
 Pasabanda (complejo)
 Bandstop (complejo)
 Multibanda (complejo)

Cuando se selecciona lowpass con un filtro FIR, los filtros a transformar pueden ser:

Pasabaja
 Pasabaja
 Pasaalto
 Pasaalto (FIR) banda estrecha
 Pasaalto (FIR) windeband
 Pasabanda
 Bandstop
 Multibanda
 Pasabanda (complejo)
 Bandstop (complejo)
 Multibanda (complejo)

En el siguiente cuadro se puede ver cada tipo de filtros originales disponibles y todos los tipos de filtro a la que se puede transformar el filtro original.

FILTRO ORIGINAL	TIPO DISPONIBLE DE TRANSFORMACIÓN DE FILTROS	
<i>Lowpass FIR</i>	<i>Lowpass</i> <i>Highpass</i> <i>Highpass (FIR) wideband</i> <i>Bandstop</i> <i>Bandpass (complex)</i> <i>Multiband (complex)</i>	<i>Lowpass (FIR)</i> <i>Highpass (FIR) narrowband</i> <i>Bandpass</i> <i>Multiband</i> <i>Bandstop (complex)</i>
<i>Lowpass IIR</i>	<i>Lowpass</i> <i>Bandpass</i> <i>Multiband</i> <i>Bandstop (complex)</i>	<i>Highpass</i> <i>Bandstop</i> <i>Bandpass (complex)</i> <i>Multiband (complex)</i>
<i>Highpass FIR</i>	<i>Lowpass</i>	<i>Lowpass (FIR) narrowband</i>

	Lowpass (FIR) wideband Highpass Bandstop Bandpass (complex) Multiband (complex)	Highpass (FIR) Bandpass Multiband Bandstop (complex)
Highpass IIR	Lowpass Bandpass Multiband Bandstop (complex)	Highpass Bandstop Bandpass (complex) Multiband (complex)
Bandpass FIR	Bandpass	Bandpass (FIR)
Bandpass IIR	Bandpass	
Bandstop FIR	Bandstop	Bandstop (FIR)
Bandstop IIR	Bandstop	

B.17.3 TRANSFORMACIÓN DEL PUNTO DE FRECUENCIA

Al introducir el punto de frecuencia en este campo identifica el valor de la respuesta en magnitud (en dB) en la curva de la respuesta en magnitud.

Al introducir los valores de la frecuencia deseada en la opción, la transformación de la frecuencia trata de establecer la magnitud de respuesta del filtro transformado por el valor de frecuencia introducido.

Si bien se puede especificar cualquier valor, se debe siempre colocar un nivel de bandstop.

B.17.4 TRANSFORMADA TIPO DEL FILTRO

Seleccione el filtro deseado de la lista de respuesta en magnitud. La lista completa es:

Lowpass
Lowpass (FIR)
Highpass
Highpass (FIR) narrowband
Highpass (FIR) wideband
Bandpass
Bandstop
Multiband
Bandpass (compleja)
Bandstop (compleja)
Multiband (compleja)

No todos los tipos de filtros transformados están disponibles para todos los tipos de filtros que están en la lista.

B.17.5 ESPECIFICACIÓN DE LA URBANIZACIÓN DE LA FRECUENCIA DESEADA

En cada frecuencia que introduzca aquí la transformación intenta en magnitud del valor a transformar.

Si bien, se puede colocar cualquier valor se desea especificar el tipo de filtro y el nivel de stopband.

B.17.6 EJEMPLO. TRANSFORMAR FILTRO

Para transformar la respuesta en magnitud del filtro, se usa la magnitud de transformación de filtros.

1. diseñe o importe el filtro en el FDATool.
2. haga clic en transformación del filtro y abre el siguiente cuadro de dialogo.

Frequency Transformations

Original filter type: Lowpass	Transformed filter type: Lowpass
Frequency point to transform: 9.6 kHz	Specify desired frequency location: 7.2 kHz

Transform Filter

3. de la lista original de tipo de filtros, seleccione la respuesta del filtro que se esta transformando.

Cuando seleccione el tipo, si es lowpass, highpass, bandpass o bandstop el FDATool reconoce si el filtro es FIR o IIR utilizando tanto el tipo y la forma del filtro, el FDATool ajusta las entradas en la lista de tipo de filtros transformados para mostrar solo en los que se aplican a su filtro original.

4. introduzca el punto de frecuencia (valor). Obsérvese que el valor que introduzca debe ser en frecuencia en Hz ; por ejemplo 0,1 por 100Hz o 1.5 por 1500Hz.

5. de la lista de tipo de filtros a transformar seleccione el tipo de filtro que desea transformar.

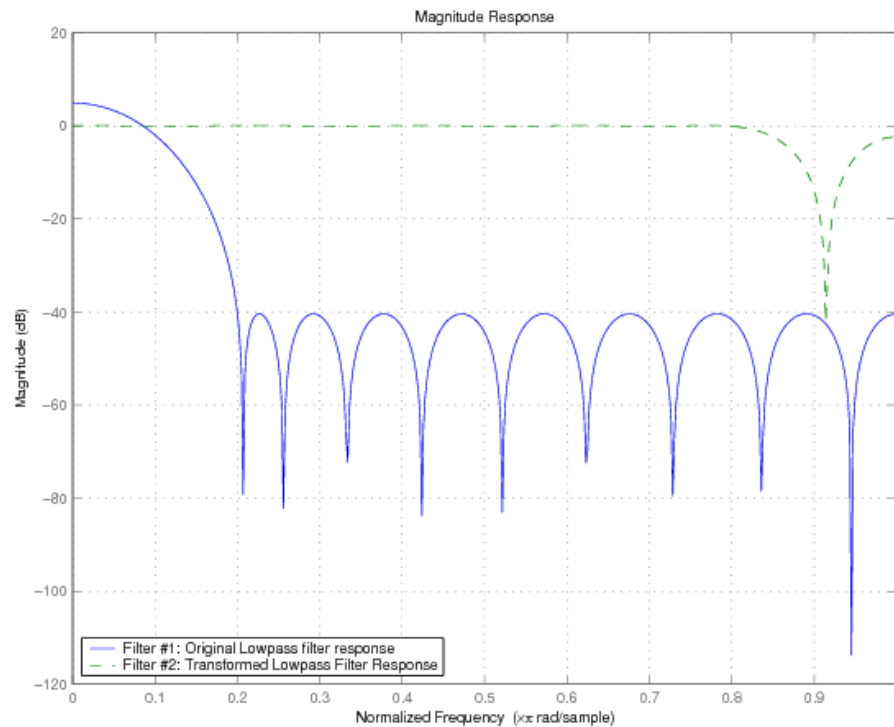
Cuando se elige un filtro lowpass o highpass se introduce un valor en "especificación de frecuencia deseada".

Cuando se elige un filtro bandpass o bandstop, se introduce dos valores; uno en “especificación de frecuencia baja deseada” y el otro en “especificación de la frecuencia alta deseada”. Estos valores definen los niveles de stopband y passband.

Cuando se elige un filtro multibanda, se introduce los valores como elementos de un vector en “especificación d3 frecuencia deseada en un vector”. Los valores define el nivel de passband y stopband.

Después haga clic en transform filter, muestra la longitud de la respuesta en magnitud del nuevo filtro y actualiza la información del filtro actual para indicar que el filtro se ha transformado.

Por ejemplo la cifra que se muestra la repuesta en magnitud de dos curvas de filtro.



El original es un filtro lowpass con rolloff entre 0.2 y 0.25.

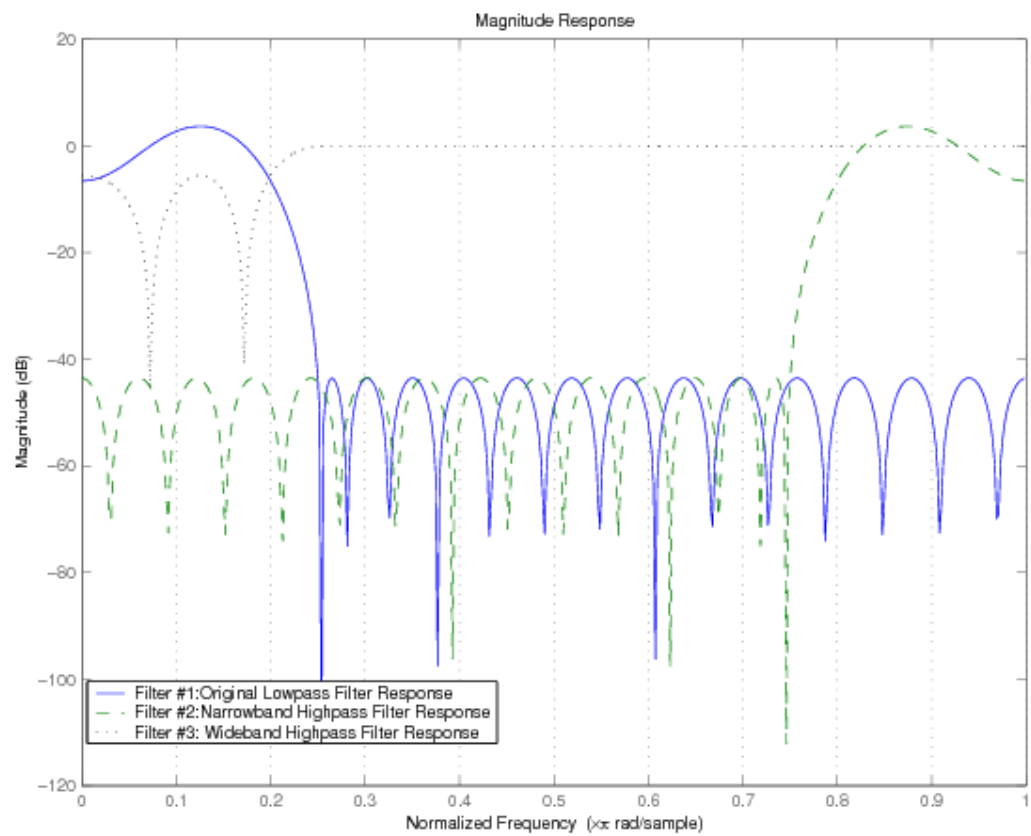
El filtro transformado es un filtro lowpass con rolloff entre 0.8 y 0.85.

Para transformar el filtro lowpass a un filtro Highpass, seleccione lowpass a highpass.

Al seleccionar esto devuelve el siguiente cuadro de dialogo:



La siguiente grafica muestra la respuesta en magnitud del filtro después de haberse transformado.



B.18 DISEÑO DE UN FILTRO MULTIRATA EN EL FDATAOL

B.18.1 INTRODUCCIÓN

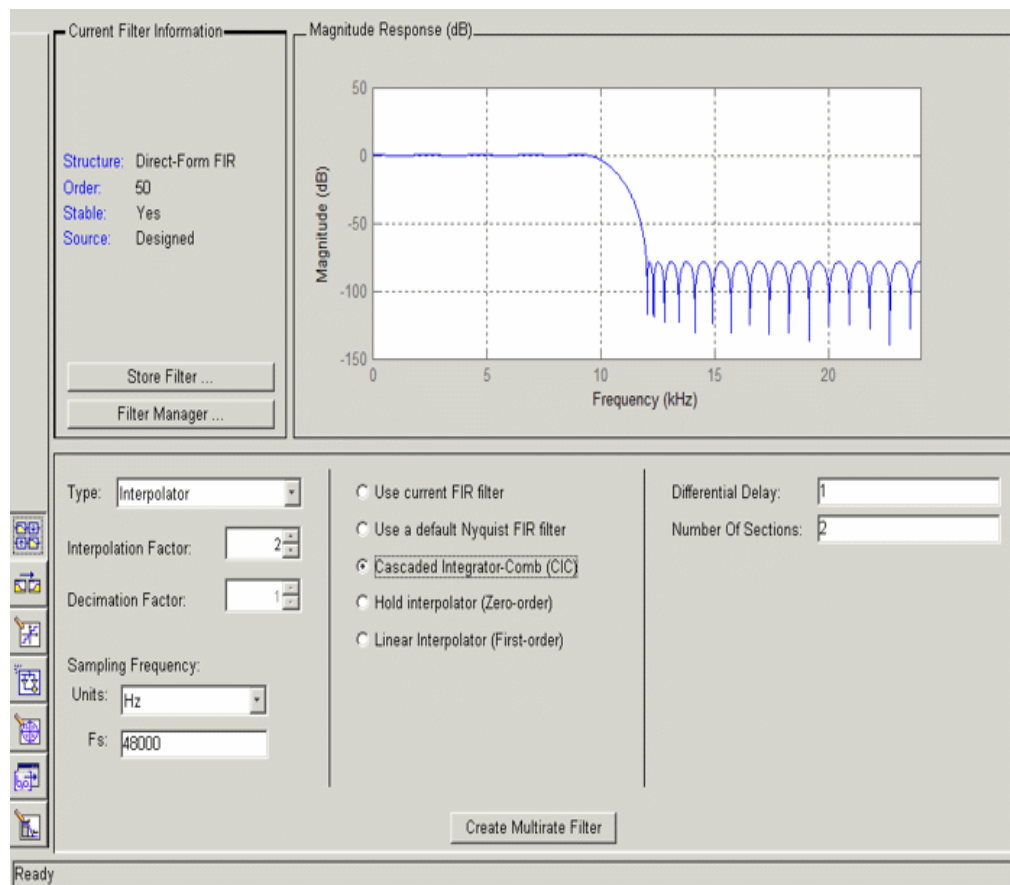
No solo se puede diseñar filtro multirate en comando de Matlab, si no que FDATool ofrece la misma capacidad de diseño de una interfaz grafica.

B.18.2 MODO DEL DISEÑO DEL FILTRO MULTIRATA EN FDATOOL

El modo de diseño de un filtro multirate en el FDATool permite especificar y diseñar una amplia gama de filtros multirate incluyendo decimadores e interpoladores.

Con el FDATool abierto, haga clic en crear un filtro multirate se puede ver el modo de diseño que muestra las opciones de diseño del filtro.

En la siguiente figura muestra la siguiente configuración por defecto del diseño multirate, que diseña un filtro interpolado con un factor de interpolación de 2. el diseño usa un filtro FIR en el FDATool.



Cuando el filtro actual en FDATool no es un filtro FIR, el diseño del filtro multirate elimina el filtro FIR y selecciona la opción por defecto Nyquist. Use filtro FIR en lugar de la opción predeterminada.

B.18.3 CONTROL SOBRE EL PANEL DEL DISEÑO MULTIRATE

Se pueden ver las diferentes opciones tipo es el punto de partida a partir de esta lista se selecciona el filtro multirata a diseñar.

B.18.4 SELECCIÓN Y CONFIGURACIÓN DEL FILTRO

OPCION	DESCRIPCION
Tipo	Especifica el tipo de diseño del filtro multirata. Se puede elegir Decimador, Interpolador o Conversión de fraccion/rata. Decimador: el factor decimador especifica la deicmacion a aplicar. Interpolador: el factor de interpolacion especifica la cantidad de interpolacion aceptada. Conversor fraccion/rata: ubica el factor de interpolacion y el factor de decimacion. El <i>FDATool</i> usa para determinar los cambios de la fraccion de la rata tanto el factor de interpolacion por division y el factor de decimacion para determinar el cambio fraccional de la rata en la señal. Se deben seleccionar los valores para la interpolacion y la decimacion que son relativamente numeros primos. Cuando estos factores no son primos el <i>FDATool</i> reduce la fraccion de rata de interpolacion/decimacion al minimo comun denominador y emite un mensaje en la barra de estado del <i>FDATool</i>. Por ejemplo, si el factor de interpolacion es y el factor de decimacion es 3, el <i>FDATool</i> reduce 2/1. cuando en el diseño de la rata cambia. Pero si el factor de interpolacion es 8 y el factor de decimacion es 3, el <i>FDATool</i> diseña el filtro sin cambios.
Factor de interpolacion	Utilice el control de flechas de arriba/abajo para especificar la antidad de interpolacion para ampliarla a la señal. El rango del factor empieza desde 2.
Factor de decimacion	Usa el control de flechas de arriba/abajo, para especificar la cantidad de decimacion para aplicarla a la señal. El rango del factor empieza en 2.
Unidades	Especifica si Fs, esta en Hz, KHz, MHz, o GHz, o normalizada (0,1).
Fs	Establece la escala maxima de la frecuencia de muestreo, en la unidad de frecuencia se especifica las unidades. Al seleccionar la unidad normalizado, esta no cuenta como un valor de Fs.

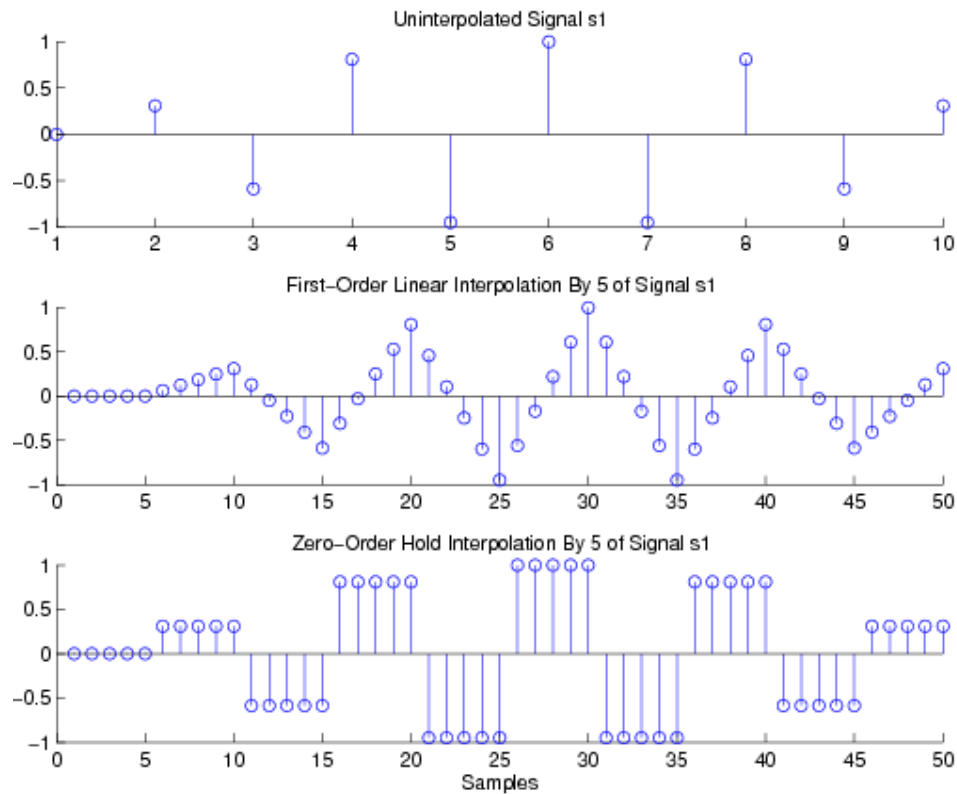
B.18.5 DESIGNACION DEL FILTRO

OPCION	DESCRIPCION
Uso de un filtro FIR	El <i>FDATool</i> directamente usa un filtro FIR para diseñar un filtro multirata. Si el filtro actual es de la forma FIR no se puede seleccionar esta opcion. No se puede diseñar un filtro multirar con las estructras de un filtro IIR.
Uso del filtro	El <i>FDATool</i> indica por defecto el uso del metodo de diseño de

Nyquist por defecto	Nyquist cuando el filtro actual en el <i>FDATool</i> no es un filtro FIR.
Integrador en cascada comb. (CIC)	Diseña filtros CIC utilizando las opciones previas en la parte derecha del área del panel del diseño multirata.
Interpolación HOLA (orden cero)	Cuando se diseña una interpolación, se puede especificar como se establecen los valores de interpolación entre los valores de la señal. Cuando se selecciona esta opción, la interpolación se aplica al valor de la señal más reciente, por cada valor interpolado hasta el máximo valor de la señal. Esto es similar a la técnica de mantener el muestreo.
Interpolación lineal (primer orden)	Cuando se diseña un interpolador se puede especificar como el filtro establece los valores interpolados en los valores de la señal. Al seleccionar esta opción, el interpolador aplica interpolación lineal entre la señal de valor par afijar el valor interpolado hasta el próximo valor de la señal.

Para ver la diferencia entre la interpolación HOLA y la interpolación lineal, la siguiente figura muestra una señal senoidal en 3 formas.

- La primera figura muestra la señal sin interpolar (S1)
- La segunda figura muestra la señal S1 interpolada, con una interpolación lineal/ de factor 5.
- La última gráfica muestra la señal S1 interpolada, con una interpolación HOLA de factor 5.



Se usa el FDATOOL para crear interpoladores similares, el siguiente código es el de figuras:

```
>>hm = mfilt. Linearinterp (S) % interpolador lineal.
>>hm = mfilt. Holdinterp (S) % interpolación hola
```

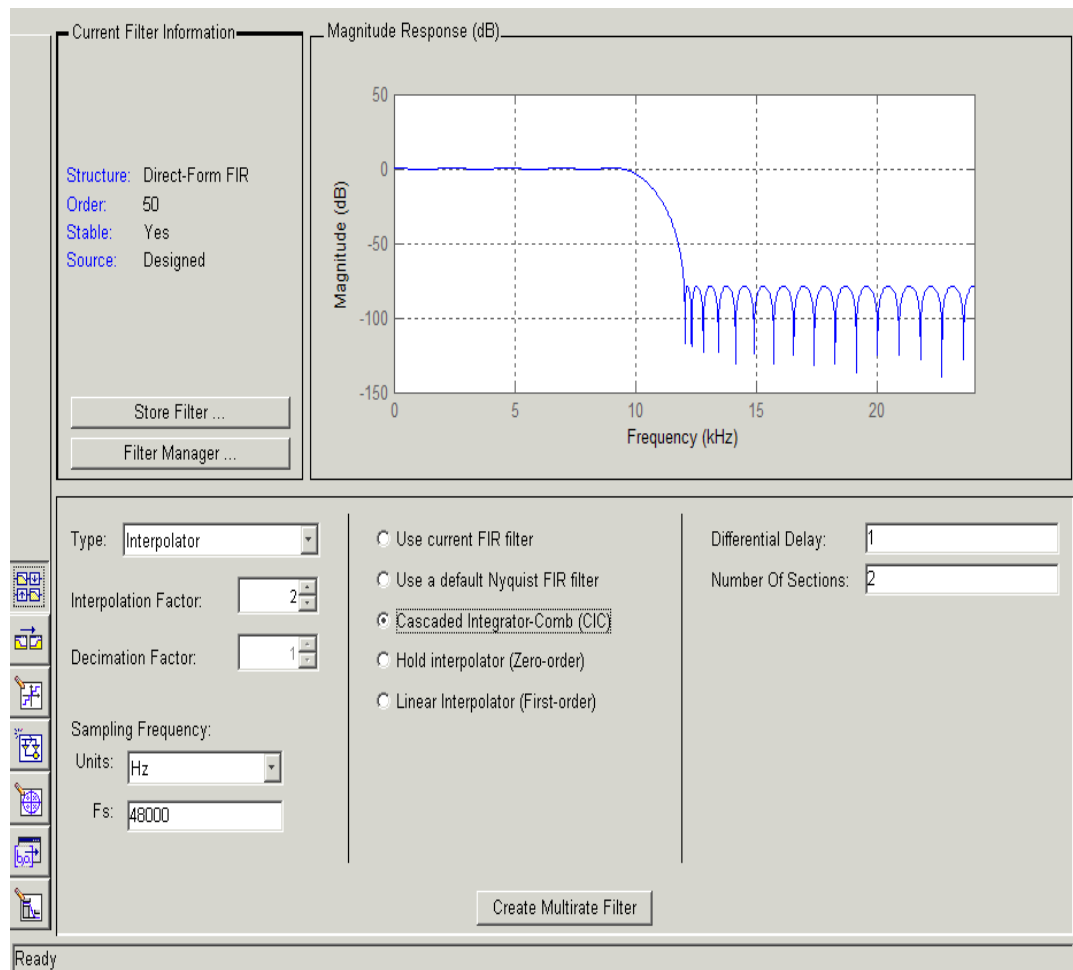
CONFIGURACIÓN PARA DISEÑAR FILTROS CIC	DESCRIPCION
Retardo diferencial	Establece la diferencia de retardo para el filtro . por lo general un valor de 1 o 2 es apropiado.
Numero de secciones	Especifica elm numero de secciones de un filtro CIC decimador. El valro por defecto de las secciones es 2 y el rango es cualquier numero entero positivo.

B.18.6 EJEMPLO: DISEÑO DE UN CONVERTOR DE RATA FRACCIONAL

Para representar el proceso se usa el diseño de un filtro multirata en el FDATOOL, este ejemplo usa la operación para diseñar un convertor de rata

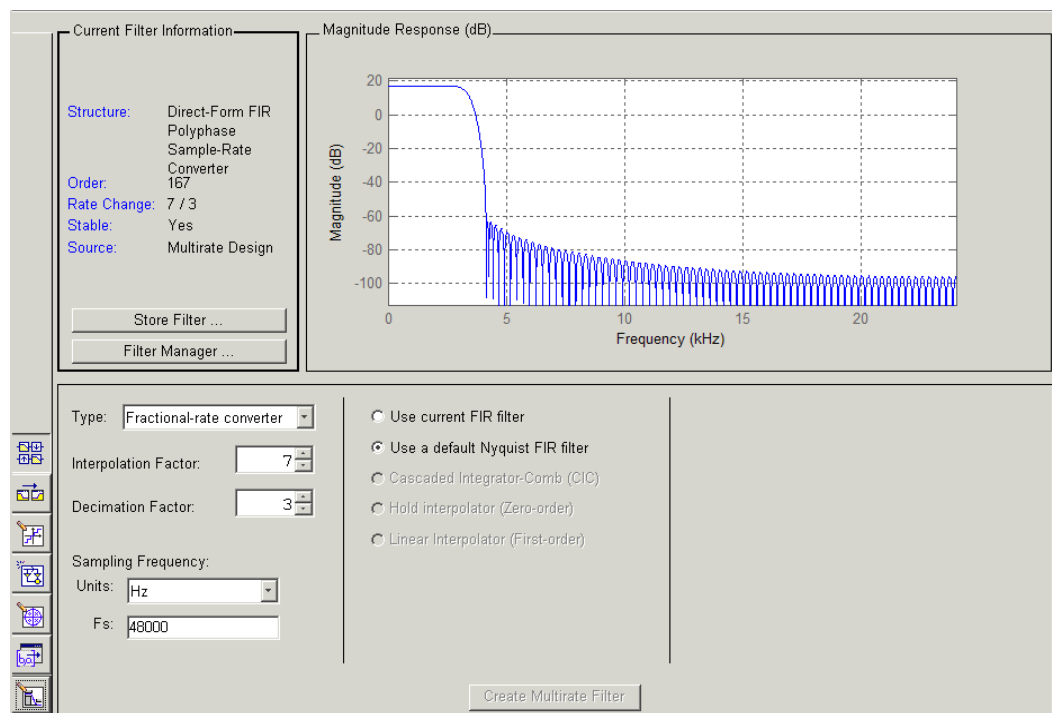
fraccional que utiliza $7/3$ como la rata fraccional. Comenzar el diseño mediante la creación de un filtro FIR lowpass en el FDATAOOL.

1. Abrir el FDATAOOL.
2. Seleccione las labores mínimas para un filtro FIR de orden mínimo, utilizando el diseño de equiripple.
3. El FDATAOOL muestra la respuesta en magnitud del filtro, haga clic en la barra lateral a modo de diseño y que muestra el panel de diseño multirata como se muestra en le siguiente figura:



4. Para diseñar un filtro de rata fraccional, seleccione de la lista el tipo de conversión fraccional. El factor de interpolación y el factor de decimación esta disponible.
5. En el factor de interpolación utilice las flechas para establecer el factor de interpolación a 7.
6. En el factor de deciminación utilice las flechas para establecer el factor de decimación a 3.

7. Seleccione por defecto el tipo nyquist. Usted puede diseñar el tipo de convertidor actual de filtro FIR.
8. Introduzca 2400 a FS.
9. Haga click en crear filtro multirata



B.18.7 EJEMPLO: DISEÑAR UN DECIMADOR CLICK PARA DATOS DE ENTRADA/ SALIDADE 8 BITS.

1. Abrir el FADTOOLy diseñe un filtro FIR lowpass
2. Cambie el FADTOOL como modo de diseño multirata.
3. El factor de decimaciòn es 3
4. Para utilizar el decimador click, seleccione “ cascaded - integrador con (CIC)”
5. Retardo diferencial como 2. En general, 1ò 2 son buenos valores de uso
6. Introduzca 2 para el número de selecciones.

Type:

Interpolation Factor:

Decimation Factor:

Sampling Frequency:

Units:

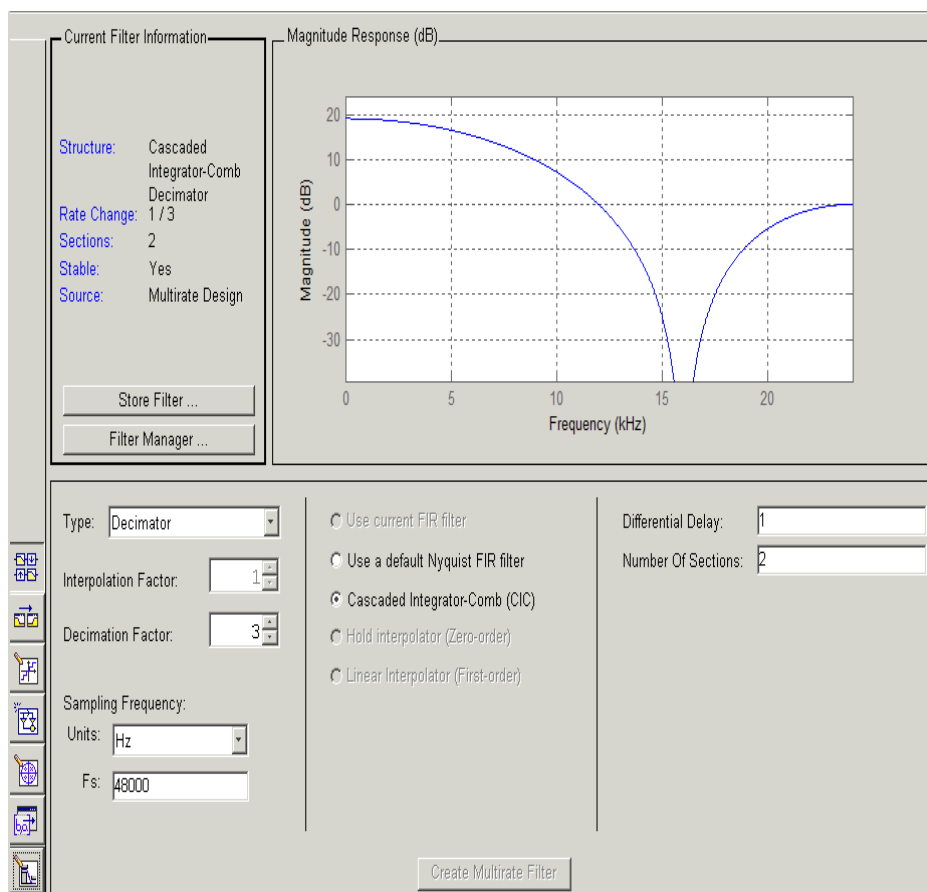
Fs:

☐ Use current FIR filter
☐ Use a default Nyquist FIR filter
☒ Cascaded Integrator-Comb (CIC)
☐ Hold interpolator (Zero-order)
☐ Linear Interpolator (First-order)

Differential Delay:

Number Of Sections:

7. Haga click en crear filtro multirata



El diseño de otros filtros multirata sigue el mismo patrón. Para diseñar un interpolador, seleccione una de estas opciones:

- Filtro FIR Nyquist casad- integrador como (CIC).
- Interpolador hold (orden zero)
- Interpolador lineal (primer orden)

Para diseñar un desimador, seleccione una de estas opciones:

- Filtro FIR Nyquist casad- integrador como (CIC)

Para diseñar un conversor de rata fraccional use por defecto:

- Filtro FIR Nyquist

B.18.8 CUANTIZADOR DE UN FILTRO MULTIRATA

Después de diseñar un filtro multirata en FDATool, la característica de un cuantizador permite convertir un filtro multirata de punto flotante en un filtro con aritmética punto fijo

NOTA: los filtros CIC son siempre punto fijo.

B.18.9 CUANTIZACION Y CONFIGURACION DE UN FILTRO MULTIRATA

Siga los siguientes pasos para convertir el filtro multirata de aritmética punto fijo y fijar las opciones de punto fijo.

1. Haga click en le botón de parámetros de cuantización
2. Seleccione el punto fijo desde la lista “filter Arithmetic” si su filtro es un filtro CIC, esta opción esta habilitada por defectos.
3. En le panel de cuantización establezca opciones de cuantización, entrada/salida y filtros internos.
4. Haga click en aplicar.

Entrada / salida: las siguientes opciones especifican la forma en que su filtro CIC utiliza los valores de entrada/salida.

NOMBRE DE OPCION	DESCRIPCION
Entrada de longitud de palabra	Establece la longitud de palabra usada para representar la entrada del filtro.
Entrada de longitud de fraccion	Establece la longitud de fraccion usada para representar la entrada del filtro.
Rango de entrada (+/-)	Permite definir el rango de entrada. Puede utilizar esta entrada en lugar de la opcion de longitud de fraccion. Al introducir un valor x, el

	rango es de $-x$ a x . El rango debe ser un numero positivo.
Salida de longitud de palabra	Establece la longitud de palabra usada para representar la salida del filtro.
<i>Avoid overflow</i>	Establece la longitud de fraccion de entrada para establecer la precision de valores de salida que excede el rango definido de la longitud de palabra.
Salida de longitud de fraccion	Establece la longitud de fraccion utilizada para representar los valores de salida de un filtro.
Rango de salida (+/-)	Permite definir el rango de salida. Al introducir un valor x , el rango es de $-x$ a x . El rango debe ser un entero positivo.

Filtros internos: con un filtro CIC, como filtro actual la opción de precisión del filtro incluye filtros internos, modos de control para la longitud de palabra y de fracción

- **Full:** todas las longitudes de palabra y de fracción establece $B_{max} + 1$. La precisión Full esta la configuración por defecto.
- **selección mínima de longitud de palabra:** coloca la selección de longitud de palabra en valores mínimos de redondeo de ruido y salida.
- **longitud de palabra especificas:** habilita la opción de longitud de palabras. Introduzca ya sea un valor escalar para el mismo valor de la selección.
- **Todas especificaciones:** proporciona la longitud de palabra y de fracción de cada sección.

B.19 REALIZACIÓN DE UN FILTRO CONO SUBSISTEMA DE BLOQUE EN SIMULINK

B.19.1 INTRODUCCIÓN

Después de diseñar o importar el filtro en FDATAOOL, el modelo característico permite crear un bloque de subsistema en simulink que implemente el filtro.

B.19.2 PANEL MODELO REALIZADO

El FDATAOOL conmuta el modo de realizar el modelo haciendo click en la barra lateral, de acceso para el panel modelo realizado y las opciones para cuantizar filtro como un bloque de subsistema de simulink.

B.19.3 OPCIONES DE MODELO:

- **bajo modelo:** establece opciones del FDATAOOL establece, donde se coloca el nuevo subsistema del bloque y como lo nombre.
- **destino:** abre o añade un bloque a la ventana del modelo de simulink seleccione modelo actual para añadir el bloque y selección del nuevo para crear un nuevo bloque.
- **nombre del modelo:** asigna el bloque un nombre.
- **sobre escribir bloque:** sobre escribe un bloque existen con el bloque, el resultado es que el bloque del filtro es sustituido con un nuevo bloque
- **crear bloque utilizado de elementos básicos:** crea un subsistema de bloque que implemente el filtro utilizado son, gain y los bloque de retardo

Nota: los filtros que usan elementos básicos aceptan muestras individuales basada en entrada no los vectores de entrada o marcos.

B.19.4 OPCIONES OPTIMIZACIÓN

Cuatro opciones habilitan la adaptación de la optimización.

- **Optimizar ganancias cero:** Especifica si elimina la ganancia con el bloque.
- **Optimizar ganancias uno:** Especifica si desea sustituir la unidad de ganancias del bloque con conexión directa en le sistema del filtro.
- **Optimizar ganancia -1:** Especificar si desea sustituir la unidad de ganancias negativas del bloque con un cambio de signo del bloque de suma mas cercano del filtro.
- **Optimizar cadena de retardo:** especifica si desea sustituir las cadenas de retardo en cascada a un solo bloque de retardo.

B.19.5 EJEMPLO: REALICE UN FILTRO USANDO FDATAOOL

Después de realizar el filtro cuantizado en el FDATAOOL con la respuesta en magnitud y fase deseada

Y con los coeficientes y forma, siga los siguientes pasos para convertir su filtro como un subsistema que se puede usar en simulink.

1. Haga click modelo realizado.
2. Seleccione el destino del modelo:
Modelo actual: Añade filtro actuales en el subsistema del modelo.
Modelo nuevo: Abre una nueva ventana del modelo de simulink y añade un subsistema del filtro ala nueva ventana
3. Proporcione un nombre para el nuevo subsistema.
4. Decida si ha de sobrescribir en le nuevo bloque y active o desactive la opción.
5. Seleccione el bloque como punto fijo.
6. Active o desactive la aplicación de optimización
7. Haga click en modelo realizado para terminar.

Se hace doble click en el bloque del subsistema del filtro se puede ver la aplicación de l filtro en el formulario de modelo de simulink.

B.19.6 SOPORTE DE LA ESTRUCTURA DE FILTRO

FDATool permite darse cuenta del tiempo discreto y los filtros multirata

ESTRUCTURA	DESCRIPCION
firdecim	Decimadores basados en los filtros FIR.
firtdecim	Decimadores basados en la transposicion de los filtros FIR.
linearinterp	Interpoladores lineales.
firinterp	Interpoladores basados en filtros FIR.
multirate polyphase	Filtros multirata.
holdinterp	Interpoladores que usa un algoritmo de interpolacion hold.
dfilt.allpass	Filtrso discretos con estructuras pasatodos.
dfilt.cascadeallpass	
dfilt.cascadewdfallpass	
mfilt.iirdecim	Decimadores basados en los filtros IIR.
mfilt.iirwdfdecim	
mfilt.iirinterp	Interpoladores basados en filtros IIR.
mfilt.iirwdfinterp	
dfilt.wdfallpass	

B.20 OBTENCIÓN DE AYUDA PARA EL FDATool

B.20.1 OPCIÓN ¿QUÉ ES ESTO?

Para encontrar información sobre una opción en particular del cuadro de diálogo.

1. Haga clic en el botón ¿Qué es esto?, y hay un cambio del cursor.
2. Haga clic en la región o la opción de interés.

También puede usar esto en el menú de ayuda.

B.20.2 AYUDA ADICIONAL PARA EL FDATAOL

Para obtener ayuda sobre la importancia de los filtros, o para obtener más información acerca del uso del FDATAOL para crear y analizar los filtros de doble precisión.

B.21 INTRODUCCIÓN A FILTROS ADAPTIVOS”

Los filtros adoptivos implican el cambio de parámetros de filtros (coeficientes) en el tiempo, para adaptarse a las características cambiantes de la señal.

Durante las 3 últimas décadas, los procesadores digitales de señales han realizado grandes avances en el aumento de la velocidad y complejidad, y la reducción del consumo de energía.

Como resultado de ello, el algoritmo del filtro adaptivo en tiempo real se convierte rápidamente en práctica y esencial para las comunicaciones de cable o inalámbricas.

- Descripción y adaptación de los filtros adaptivos
- Sistema de identificación
- Sistema inverso de identificación
- Ruido o interferencia de cancelación
- Predicción
- Caja de herramientas del diseño de filtros adaptivos.
- Ejemplos de algoritmo de filtros adaptivos usando LMS.
- Ejemplo de algoritmos de filtros adaptivos usando RLS

B.22 FILTROS ADAPTIVOS EN TOOLBOX DE DISEÑO DE FILTROS

B.22.1 DESCRIPCIÓN DE UN FILTRO ADAPTIVO EN EL TOOLBOX DE DISEÑO DE FILTROS

El toolbox de diseño de filtros contiene muchos objetos para el diseño, la construcción y aplicación de filtros adaptivos para datos.

Si bien los algoritmos difieren en sus detalles de implementación, el LMS y RMS comparten un enfoque operacional y es el de minimizar el error entre el filtro de salida la señal deseada.

B.22.2 ALGORITMO

Para objetos (adaptfilt) de filtros adaptivos. Cada algoritmo disponible aparece la entrada en una de las tablas junto con una breve descripción del algoritmo.

Haga clic en el algoritmo de la primera columna para obtener más información acerca de los asociados de adaptación técnica del filtro:

- LMS basados en filtros adaptivos.
- RLS basados en filtros adaptivos.
- Filtros adaptos de proyección afin.
- Filtros adaptivos en el dominio de la frecuencia.
- Filtros adaptivos basados en el Lattice

B.22.3 FILTROS ADAPTIVOS FIR BASADOS EN LMS (LEAST Mean Squares)

MÉTODO DE FILTROS ADAPTIVOS	ADAPTACIÓN DE ALGORITMO USANDO LA ADAPTACIÓN DE GENERACIÓN DE COEFICIENTE DE FILTROS
adaptfilt.adjlims	Algoritmo adaptivo filtro FIR LMS
adaptfilt.blms	Algoritmo adaptivo filtro FIR LMS BLACK
adaptfilt.blmsfft	Algoritmo adaptivo filtro FIR con black LMS basado en FFT
adaptfilt.dlims	Algoritmo adaptivo filtro FIR, LMS, tardía.
adaptfilt.filtxlims	Algoritmo adaptivo filtro FIR LMS con filtrado en X
adaptfilt.lms	Algoritmo adaptivo filtro FIR LMS
adaptfilt.nlms	Algoritmo adaptivo filtro FIR LMS normalizados
adaptfilt.sa	Algoritmo adaptivo filtro FIR LMS registro de datos
adaptfilt.se	Algoritmo adaptivo filtro FIR LMS registro de error
adaptfilt.ss	Algoritmo adaptivo filtro FIR LMS registro signo

B.22.4 FILTROS ADAPTIVOS FIR BASADOS EN RLS (Recursive least square)

MÉTODO DEL FILTRO ADAPTIVOS	ADAPTACIÓN DE ALGORITMO UTILIZADO PARA GENERAR FILTROS CUANTIZADOS MIENTRAS LA ADAPTACIÓN.
adaptfilt.ftf	Algoritmo de adaptación fase transversal mínimos cuadrados
adaptfilt.qrdrls	Algoritmo de adaptación descomprimiendo QR – RLS
adaptfilt.hrls	Algoritmo de adaptación propietario

	RLS
adaptfilt.hswrls	Algoritmo de adaptación familia SWRLS
adaptfilt.rls	Algoritmo de adaptación RLS de mínimos cuadrados recursivos.
adaptfilt.swils	Algoritmo de adaptación RLS ventana deslizante (sw)
adaptfilt.swftf	Algoritmo de adaptación ventana deslizante FTF.

B.22 5 FILTRO DE ADAPTACIÓN FIR, CON PROYECCIÓN AFIN (AP)

MÉTODO DEL FILTRO ADAPTIVOS	ADAPTAR EL ALGORITMO UTILIZANDO PARA GENERAR LOS COEFICIENTES DE FILTROS DURANTE LA ADAPTACIÓN.
adaptfilt.Ap	Algoritmo que utiliza inversión directa de matriz.
adaptfilt.Apn	Algoritmo que utiliza actualización de matriz recursiva
adaptfilt. bap	Bloque de algoritmo de adaptación

B.22.6 FILTROS ADAPTIVOS FIR EN EL DOMINIO DE LA FRECUENCIA

MÉTODO DE FILTRO ADAPTIVO	DESCRIPCIÓN DE LA ADAPTACIÓN DEL ALGORITMO UTILIZADO PARA GENERAR LOS COEFICIENTES DEL FILTRO DURANTE LA ADAPTACIÓN
adaptfilt. Fdaf	Algoritmo de adaptación en el dominio de la frecuencia.
adaptfilt.pbfdaf	Algoritmo de versión de bloque FDAF con partición.
adaptfilt.pbsfdaf	Algoritmo de versión de bloque FDAF sin partición.
adaptfilt.tdafdct	Algoritmo de adaptación transformado el dominio DCT
adaptfilt.tdafdft	Algoritmo de adaptación transformado el dominio DFT.
adaptfilt.ufdaf	Algoritmo para la adaptación si FDAF

B.22. 7 FILTROS ADAPTIVOS FIR (L) LATTICE – BASED

MÉTODO DE FILTRO ADAPTIVO	DESCRIPCIÓN DE LA ADAPTACIÓN DEL ALGORITMO UTILIZADO PARA GENERAR LOS COEFICIENTES DEL
---------------------------	--

	FILTRO DURANTE LA ADAPTACIÓN
adaptfilt.gal	Algoritmo de adaptación para filtros adaptivos con lattice gradiente
adaptfilt./s/	Algoritmo de adaptación lattice mínimos cuadrados
adaptfilt.qrd/s/	Algoritmo de adaptación RLS con descomposición QR.

B.23 PROPIEDADES DE UN FILTRO PUNTO FIJO

B.23.1 DESCRIPCIÓN DE UN FILTRO PUNTO FIJO

Existe una diferencia entre los filtros punto fijo y los filtros cuantizados; los filtros cuantizados representan un superconjunto que incluye filtros de punto fijo. Los objetos dfilt tienen una propiedad aritmética única o fija que los filtros de cuantificación. Sin embargo después de configurar la propiedad aritmética “fija” el resultado del filtro es cuantizado y de punto fijo.

B.23.2 OBJETOS Y FILTROS PUNTO FIJO

Los filtros punto fijo dependen de los objetos punto fijo de la caja de herramientas de punto fijo. Se puede ver en esto cuando se muestra un filtro de punto fijo en el prompt de matlab.

```
>>hd=dfilt.df2
hd =
    FilterStructure: 'Direct-Form II'
      Arithmetic: 'double'
      Numerator: 1
      Denominator: 1
 PersistentMemory: false
        States: [0x1 double]
>>hd.arithmetic='fixed'
hd =
    FilterStructure: 'Direct-Form II'
      Arithmetic: 'fixed'
      Numerator: 1
      Denominator: 1
 PersistentMemory: false
        States: [1x1 embedded.fi]
  CoeffWordLength: 16
    CoeffAutoScale: true
        Signed: true
 InputWordLength: 16
 InputFracLength: 15
 OutputWordLength: 16
   OutputMode: 'AvoidOverflow'
 StateWordLength: 16
 StateFracLength: 15
   ProductMode: 'FullPrecision'
 AccumWordLength: 40
   CastBeforeSum: true
```

```
RoundMode: 'convergent'
OverflowMode: 'wrap'
```

La notificación `embedded.fi` indica que el estado está representado por un objeto punto fijo, usualmente llamado objeto `fi`. Como se ve a continuación las propiedades del objeto `fi` representan valores de los estados del filtro.

B.23.3 PROPIEDADES DE UN FILTRO PUNTO FIJO

Los filtros en tiempo discreto usan objetos del toolbox que realizan el filtrado y configuración del filtro.

En los filtros de tiempo discreto o `dfilt` muchas de las propiedades son dinámicas, en el sentido de que se disponga de ellos en función de la configuración de otras propiedades del `dfilt`.

B.23.4 PROPIEDADES DINÁMICAS

Al utilizar una función `dfilt.structure` para crear un filtro. Matlab muestra las propiedades del filtro en la ventana de comandos de retorno. En general, se muestran seis o siete propiedades, que van desde `filterstructure` hasta `persistent memory`. Estas primeras propiedades están siempre presentes en el filtro.

Una de las propiedades más importante es la de aritmética, donde controla las propiedades dinámicas de un filtro.

`>>Productmode = "full precisión".`

La siguiente tabla resume las propiedades estáticas y dinámicas, de los filtros punto fijo y representa una breve descripción de cada uno.

Nombre de Propiedad	Valor Válido	Descripción
Accum Frac Length	Número de bit positivos o negativos	Especifica la longitud de fracción utilizada para interpretar los datos de salida del acumulador. Esta es una propiedad de la FIR y Lattice. Los filtros IIR tiene dos propiedades similares Den Acum Fraclength que permiten establecer la precisión del numerador y las operaciones del denominador por separado.
Accum Word Length	Número de bit entero positivo	Establece la longitud de palabra usado para almacenar los datos en el buffer acomodador.
Aritmético	[Double] Single fijo	Define la aritmética que usa el filtro. Da opciones individuales, dobles y fijos. Define el modo de funcionamiento del filtro.

Cast Sum	Befote	Verdadero o Falso	Especifica si los _____ numéricos a emitir el formato del acumulador es adecuado antes de llevar a cabo la operación de suma.
Copff Seale	Auto	Verdadero o falso	Especifica si el filtro escoge automáticamente la función de longitud adecuada para representar los coeficientes del filtro sin overflow.
Copff Fraclength		Cualquier positivo o negativo de bits	Establece la longitud de fracción del filtro que se usa para interpretar los coeficientes.
Copf length	Word	Cualquier número positivo de bits	Especifica la longitud de palabra para aplicar los coeficientes del filtro.
Den Accum Fraclength		Cualquier número positivo o negativo de bits	Especifican cómo se interpreta el algoritmo del filtro en el sentido del resultado de operaciones de suma donde se involucren los coeficientes del denominador.
Den Fraclength		Cualquier número positivo o negativo de bits	Establece la longitud de fracción del filtro para interpretar coeficientes.
Denominador		Sólo valores del coeficiente del filtro,	Posee el denominador de los coeficientes del filtro IIR.
Den Prod Fraclength		Solo valores positivos o negativos del bit.	Especifica cómo el filtro interpreta los resultados de las operaciones de multiplicación den los coeficientes del denominador.
Den shale fraclength		Sólo valores positivos o negativos de bits.	Especifica la longitud de fracción usado para interpretas los estados asociados con los coeficientes del denominador del filtro.
Frac Delay		Sólo valores decimales 0 ó 1	Especifica el retardo fraccional proveniente del filtro en fracción decimal de una muestra.
FDautoscale		Verdadero o falso	Especifica cómo el algoritmo del filtro interpreta los resultados de las operaciones de los coeficientes del denominador.
FDFraclength		Sólo valores positivos o negativos de bits.	Especifica la longitud de fracción para representar el retardo fraccional.
FDProd Fraclength		Sólo valores positivos o negativos de bits.	Especifica la longitud de fracción para representar el resultado de la multiplicación de los coeficientes con fracaso de retardo.
FD Prod Word		Sólo valores	Especifica la longitud de palabra para

length	enteros o positivos de bits	representar el resultado de multiplicación de los coeficientes con un retardo fraccional.
FD Word length	Sólo valores positivos o negativos de bits	Especifica la longitud de palabra para representar la fracción de retardo.
Den State Word length	Cualquier número positivo de bits	Especifica la longitud de palabra usada para representar los estados asociados con los coeficientes del denominador del filtro.
Filter Internals	[Full precisión] Specify Precisión	Controla si el filtro establece salidas de longitud de palabra y fracción y si el acumulador de longitud de fracción y de palabra mantiene automáticamente la mejor precisión de los resultados durante el filtrado. El valor por defecto Full Precisión, establece automáticamente la longitud de palabra y de fracción determinado por el filtro. Specify Precisión expone la salida y las propiedades relacionando al acumulador de modo que se pueda fijar su propia palabra y longitud de fracción.
Structura del Filtro	No aplicable	Describe el flujo de la señal para el objeto del filtro, incluyendo todos los objetos (elementos) activos que realiza operaciones durante el filtrado (ganancias, retardos, sumas, productos y entradas, salida).
Input Frac length	Sólo números positivos o negativos de bits	Especifica la longitud de fracción que el filtro usa para interpretar los datos que son procesados por el filtro.
Input Word length	Sólo números positivos o negativos de bits	Especifica la longitud de palabra aplicada para representar la salida de datos.
Ladder	Cualquier coeficiente ladder de doble precisión en el tipo de datos.	Los filtros lattice incluyen esta propiedad para almacenar los coeficientes ladder.
Ladder Accum Frac Length	Sólo números positivos o negativos de bits	Define la duración aplicada a los valores de salida para el acumulador que almacena los valores de los cálculos ladder.
Ladder Frac Length	Sólo números positivos o	Determina la longitud de fracción utilizada para interpretar los coeficientes.

	negativos de bits.	
Lattice	Cualquier estructura de coeficientes Vu hoy valores.	Almacena los coeficientes lattice para filtros basados en lattice.
Lattice Accum Frac Length	Sólo números positivos o negativos de bits.	Especifica la forma en que el acumulador de salidas de los resultados de operaciones en los coeficientes de lattice.
Lattice Fra Length	Sólo números positivos o negativos de bits	Especifica la longitud de fracción aplicado a los coeficientes lattice.
Multiplicand Fra Length	Sólo números positivos o negativos de bits	Establece los valores de la longitud de fracción usados en operaciones de productos en el filtro. Esta propiedad incluye filtros de estructura de forma directa ____ transpuesto.
Multiplicand Word Lenght	Todo número positivo de bits	Establece la longitud de palabra que se aplica a la entrada de la operación de multiplicación. El filtro dfit incluye esta propiedad.
Vum Accum Frac Length	Sólo números positivos y negativos de bits.	Especifica cómo el algoritmo del filtro interpreta los resultados de las operaciones además de los coeficientes del numerador.
Numerador	Sólo para coeficientes de filtros de doble precisión	Posee los valores de coeficientes del numerador para el filtro.
Num fracc length	Sólo para números positivos o negativos de bits	Establece la longitud de fracción usado para interpretar los coeficientes del filtro.
Num Prod Length	Sólo números positivos o negativos de bits.	Especifica cómo el algoritmo del filtro interpreta los resultados de las operaciones del producto de los coeficientes del numerador.
Num Stale Frac Length	Sólo números positivos o negativos de bits.	Para filtros IIR, define la longitud de fracción del numerador que se aplica a los estados del filtro. Especifica la longitud de fracción utilizada para interpretar los estados asociados a los coeficientes del numerador del filtro.

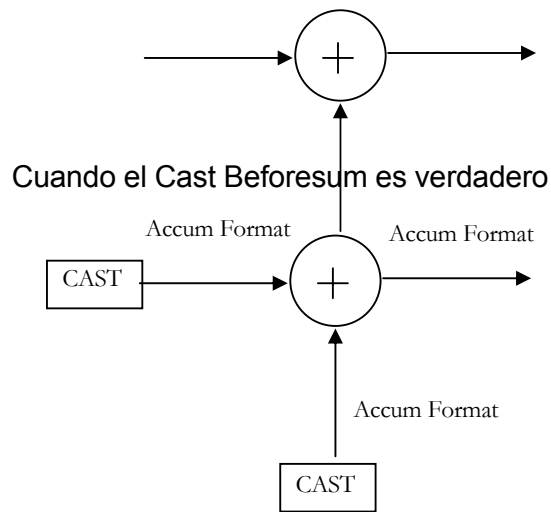
Num Stale Word Length	Sólo números positivos de bits.	Para filtros IIR, define la longitud de palabras que se aplica a los estados del numerador del filtro. Especifica la longitud de palabra usada para interpretar los estados asociados a los coeficientes del numerador del filtro.
Out put fraclength	Sólo números positivos o negativos de bits dependiendo de la estructura del filtro.	Determina el filtro interpreta los datos filtrados.
Out mode	[Avoid Overflow] Best precisión specify precisión	Establece el modo que el filtro usa para escalar el filtrado en la entrada de datos. Se tiene las siguientes opciones: Avord Querflow: dirige el filtro para establecer la salida de datos de la longitud de fracción para evitar el overflow de datos. Best Precisión: dirige el filtro para establecer la salida de datos de la longitud de fracción para aumentar al máximo la precisión de salida de datos. Specify Precisión: establece la longitud de fracción usada por los datos filtrados.
Out Put Wordlenght	Sólo números positivos o bits.	Determina la longitud de palabra usada para los autos filtrados.
Overflor moo	Saturado o wip	Establece el modo a usar para responder a las condiciones de overflow en la aritmética de punto fijo. Saturate: limita la salida en el valor representable mayor o menor. Wnp: desborda los valores establecidos al valor más cercano usando aritmética modular.
Product Frac Length	Sólo números positivos o negativos de bits	Para la salida de la operación de un producto, esto establece la longitud de fracción a usar para interpretar los datos numéricos.
Product Frac Lenght	Sólo positivos o negativos de bits	Establece la longitud de fracción para interpretar los datos numéricos del producto.
Product Mode	Full precisión keepiso	Determina la manera en que el filtro se encarga de las operaciones del producto de

	Keepmsb Specify Precisión	salida.
Product Word Length	Sólo números positivos de bits. Por defecto es 16 o 32 dependiendo de la estructura del filtro.	Especifica la longitud de palabras a utilizar para resultado de las multiplicaciones.
Persistent Memory	Verdadero o falsa	Especifica si el filtro reestablece los estados y la memoria antes de cada operación de filtrado. Permite definir si el filtro retiene los estados de filtrados de estados anteriores.
Roun Mode	Convergente cel Fix flor Round	Establece el modo que el filtro usa para cuantizar valores numéricos.
Scale valve fra length	Sólo números positivos o negativos de bits	Escala de valores que trabajen los filtros sos.
Cale valves	Matriz (2 x 1) con valores de 1.	Almacena el valor escalizado de las secciones del filtro sos.
Signed	Verdadero o falso	Especifica si el filtro utiliza o no signo en los coeficientes del punto fijo.
SOS Matrix	[100100]	Muestra la matrix en el formato (secciones coeficientes x sección dato).
Section Input Autoscala	Verdadero o falso	Especifica si el filtro escoge automáticamente la longitud de fracción adecuada para evitar el overflow de los datos que entran por una sección del filtro sos.
Section Input Fraclength	Sólo números positivos o negativos de bits	Controla la forma en que el filtro interpreta los valores de la sección del filtro mediante el establecimiento de la longitud de fracción.
Section Input Word Length	Sólo números positivos o negativos de bits	Establece la longitud de palabra utilizada para representar el movimiento de los datos en una sección del filtro sos.
Section Output Autoscala	Verdadero o falso	Especifica si el filtro escoge automáticamente la longitud de fracción adecuada para evitar el overflow de los

		datos de salida de la sección de un filtro sos.
Section Output Fracc Lenght	Sólo números positivos o negativos de bits	Controla la forma en que el filtro interpreta las valores entre secciones del filtro mediante el establecimiento de la longitud de fracción.
Section Output Word Length	Sólo números positivos o negativos de bits	Establece la longitud de palabras utilizadas para representar los datos procedentes de una sección del filtro sos.
Stale Frac Length	Sólo números positivos o negativos de bits	Define la longitud de fracción que se aplica para interpretar los estados del filtro.
Status	[1x1] embebed fi	Contiene los estados del filtro, antes, durante y después de las operaciones del filtro.
State Word length	Sólo números positivos de bits	Establece la longitud de palabras utilizada para representar los estados del filtro.
Topsum fraclength	Sólo números positivos o negativos	Establece la longitud de fracción utilizada para representar los valores de top del filtro además de las operaciones de valores.
Tap Sam Mode	Full Precisión keep.LSB KeepMSB Specific / Precisión	Determina cómo el acumulador de salida almacena los top que involucren al filtro.
Tap Sum Word Length	Sólo números positivos de bits	Establece la longitud de palabra utilizada para representar el top del filtro durante la suma.

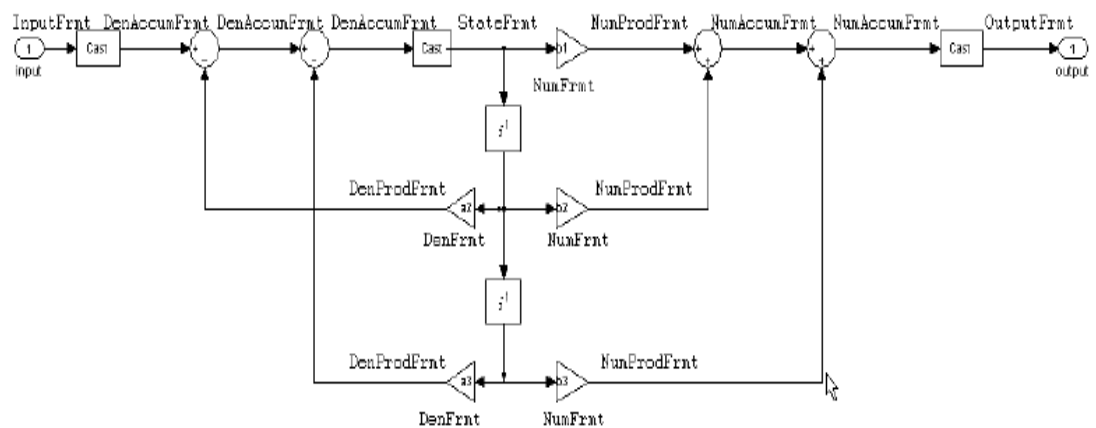
Cadena de Propiedades Aritméticas	Breve Descripción del Efecto en el Filtro
Doble (double)	Todas las operaciones del filtrado y los coeficientes usan punto flotante de doble precisión. Cuando se usa dfilter.structure para crear un objeto de filtro, doble es el valor por defecto de la aritmética.
Único (single)	Todas las operaciones del filtrado y los coeficientes usa precisión única y punto flotante en las representaciones matemáticas.
Fijo (fixed)	Se aplica para determinados valores por defecto en los filtros de punto fijo, como en los propiedades de coeficientes de longitud de palabra, longitud de fracciones y diversos modos de funcionamiento.

B.23.5 DIAGRAMAS DE CASTBEFORESUM:

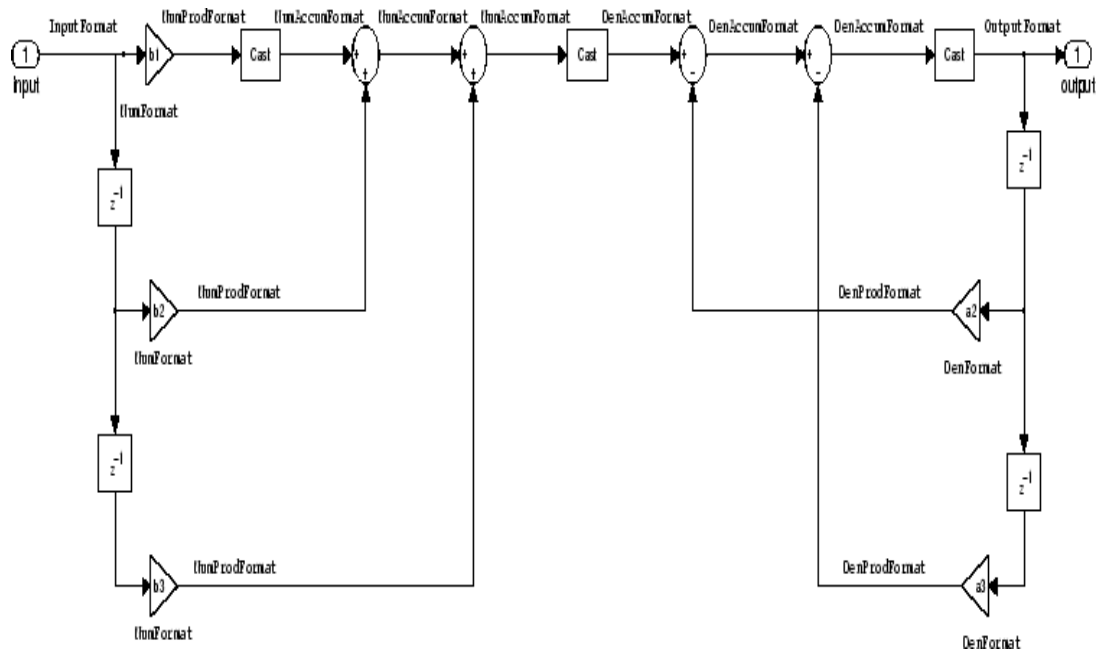


B.23.6 ESTRUCTURA DE FILTROS CON CUANTIZADORES:

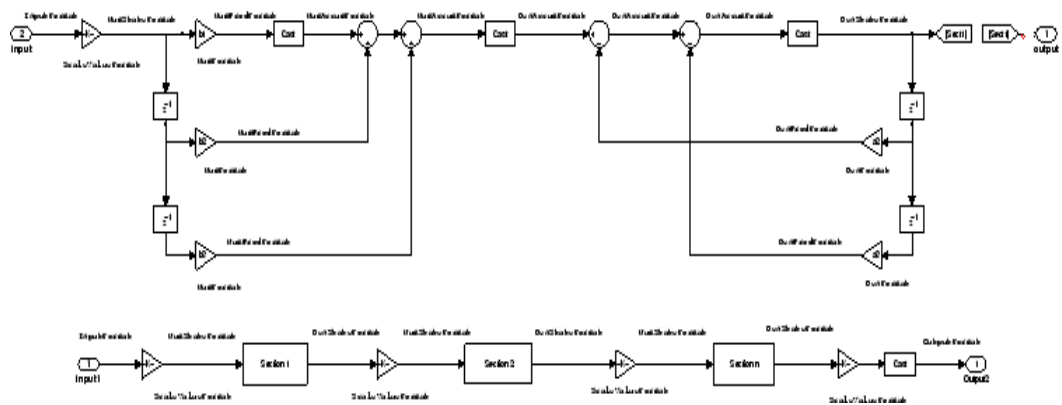
ESTRUCTURA DEL FILTRO IIR dfz (forma directa II):



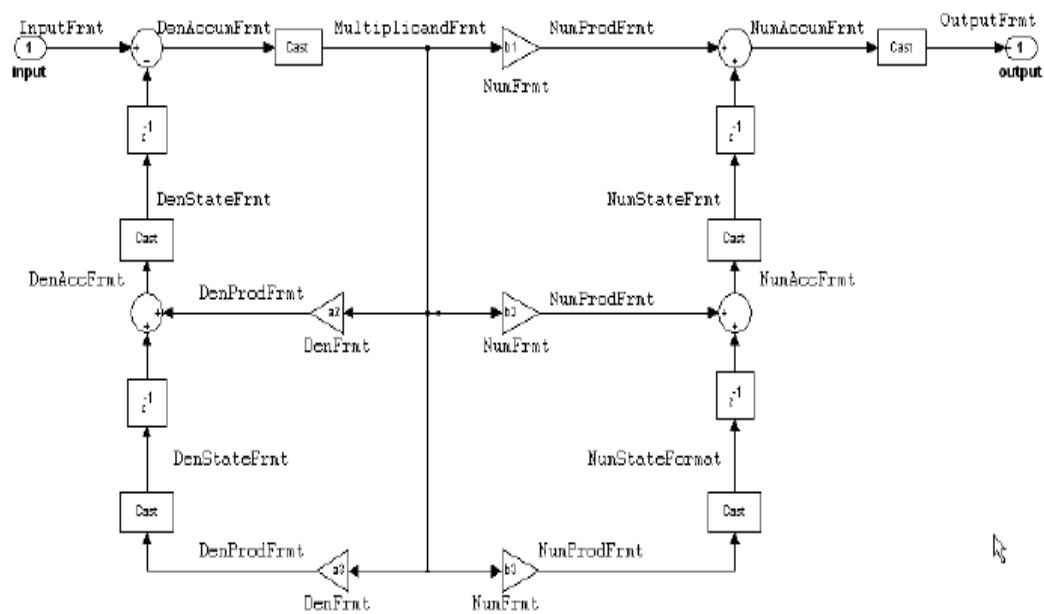
ESTRUCTURA DEL FILTRO FORMA DIRECTA I:



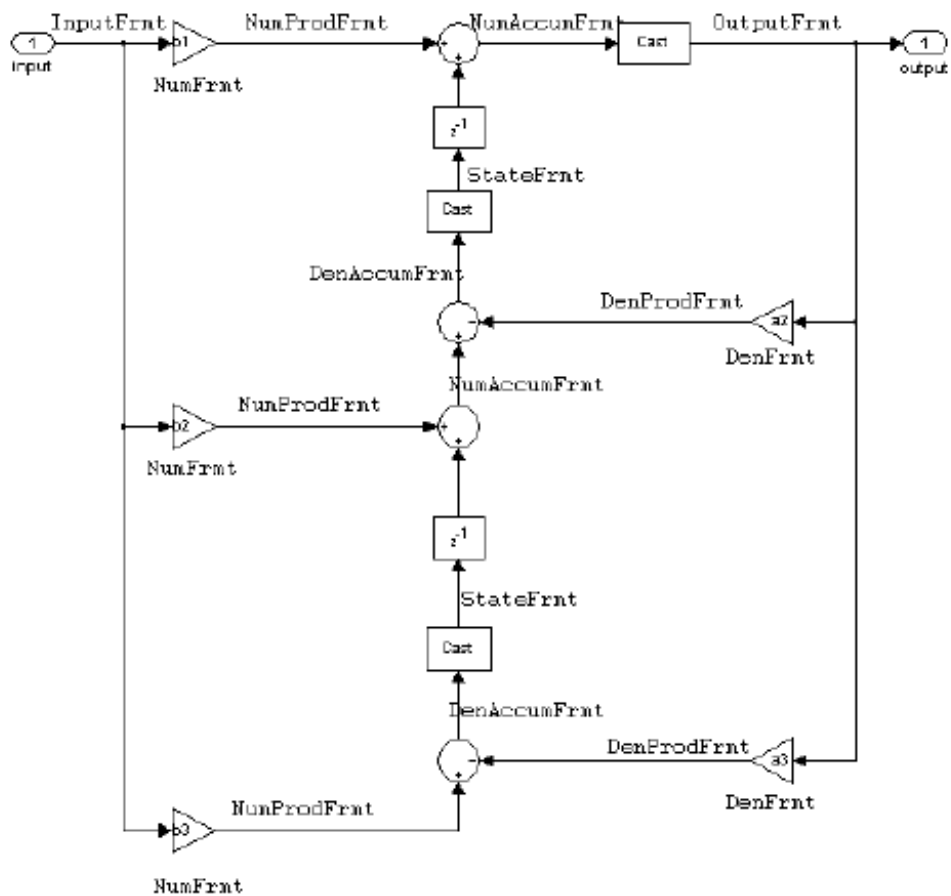
ESTRUCTURA DEL FILTRO FORMA DIRECTA I DE SEGUNDO ORDEN:



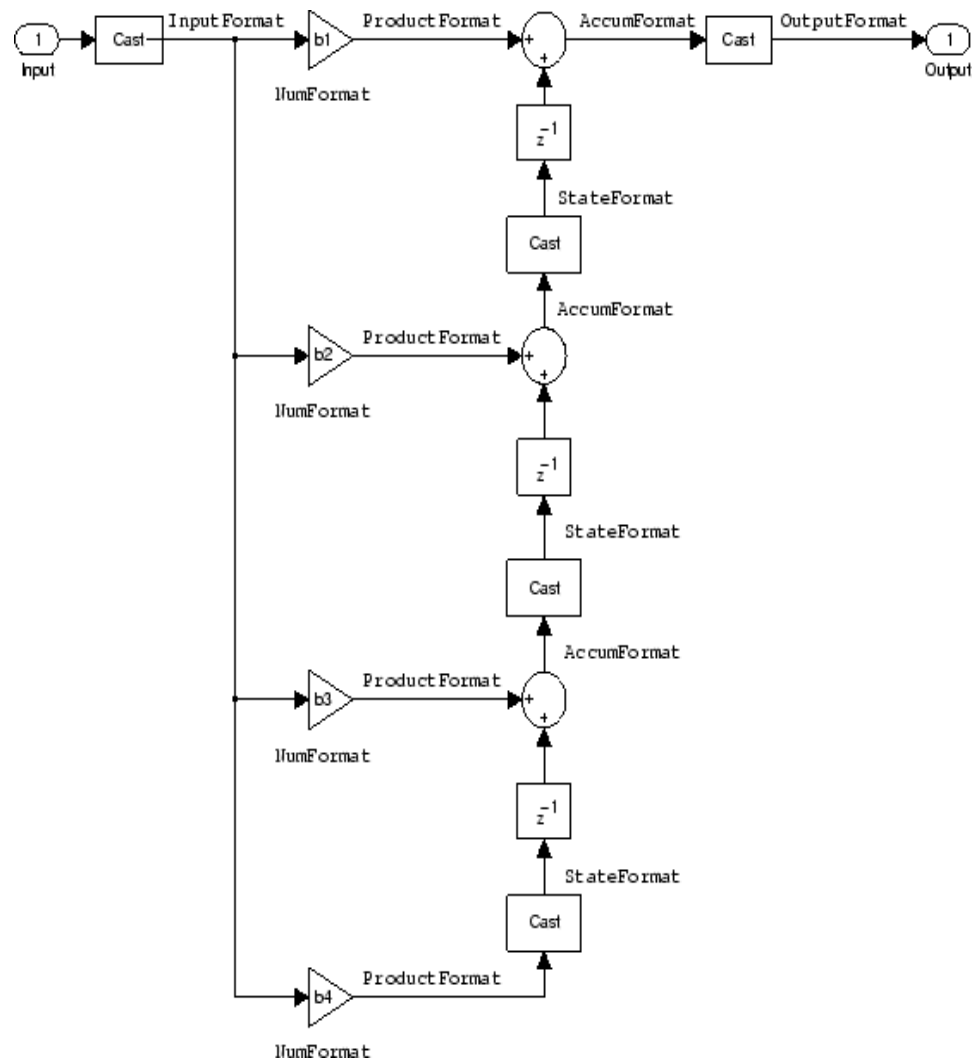
ESTRUCTURA DEL FILTRO 1D TRANSPUESTO:



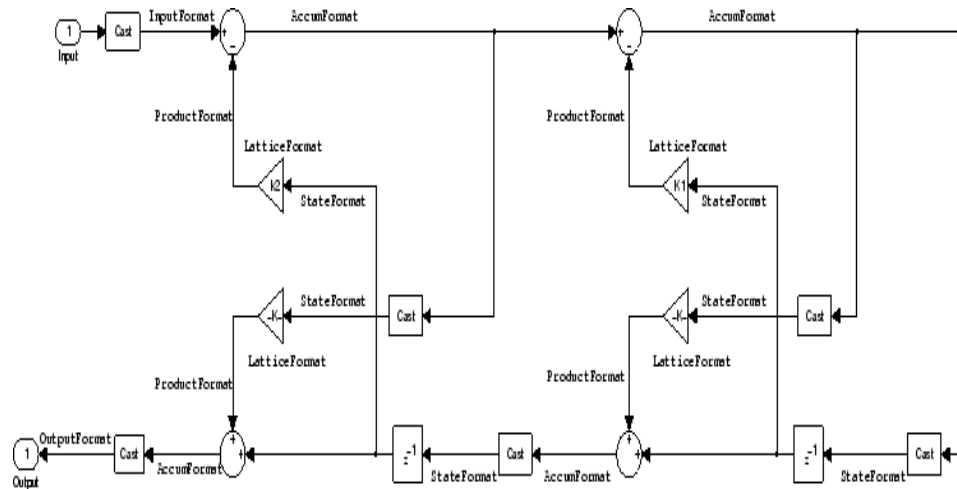
ESTRUCTURA DEL FILTRO 2D:



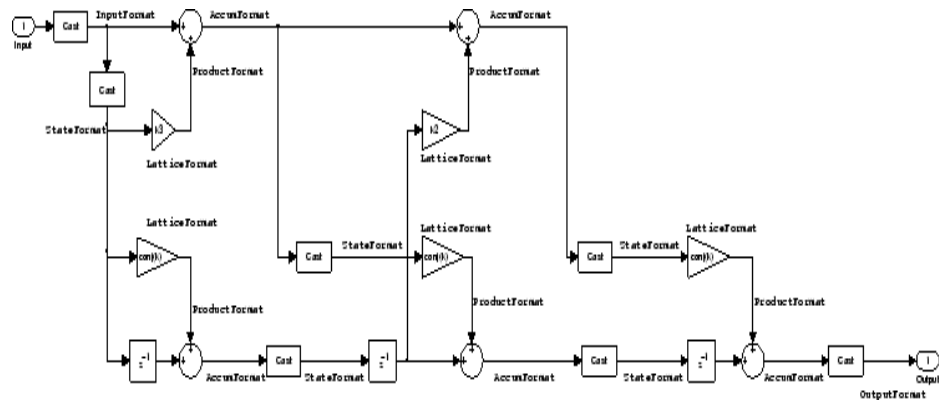
FILTRO FIR DE FORMA DIRECTA ASIMÉTRICA DE CUALQUIER ORDEN:



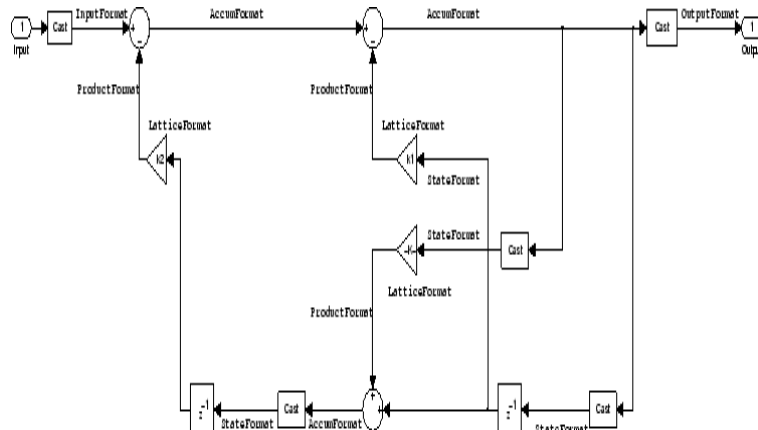
FILTRO LATTICE LOW PASS:



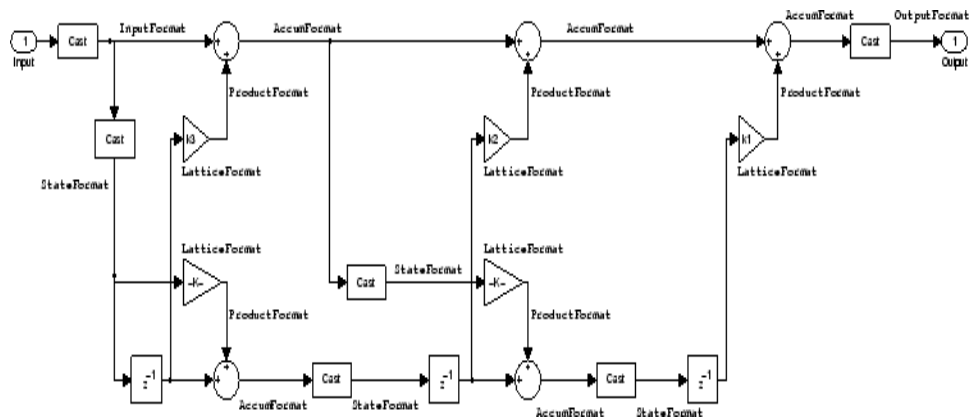
CONTROL DE LONGITUD DE PALABRA Y FRACCION DE UN FILTRO LATTICE:



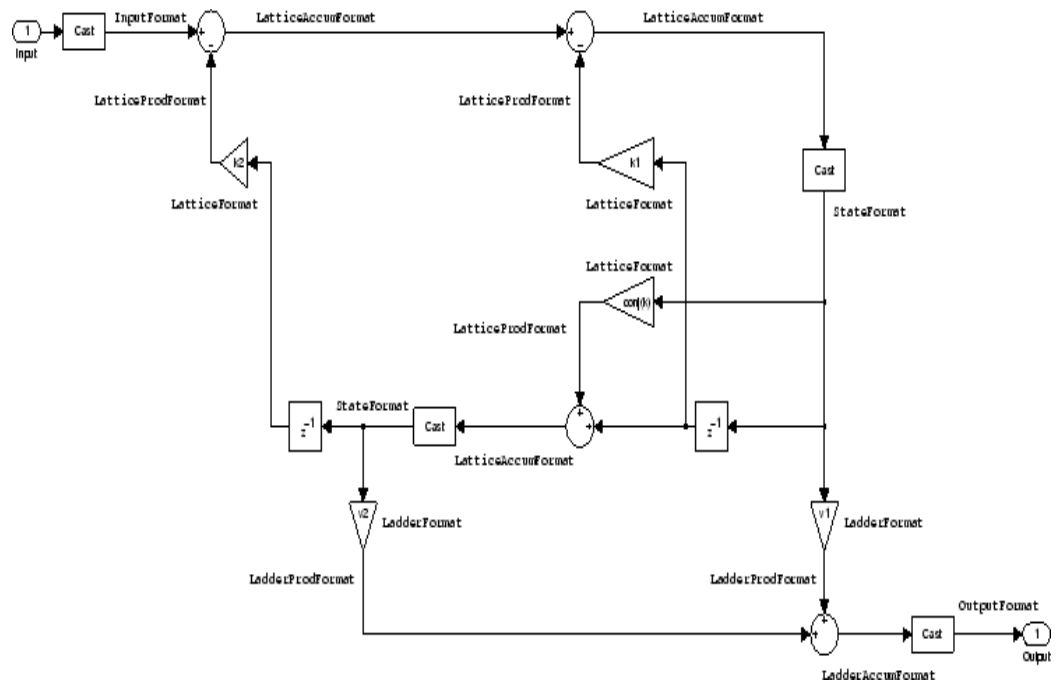
FILTRO LATTICE AUTOREGRESIVO (AR):



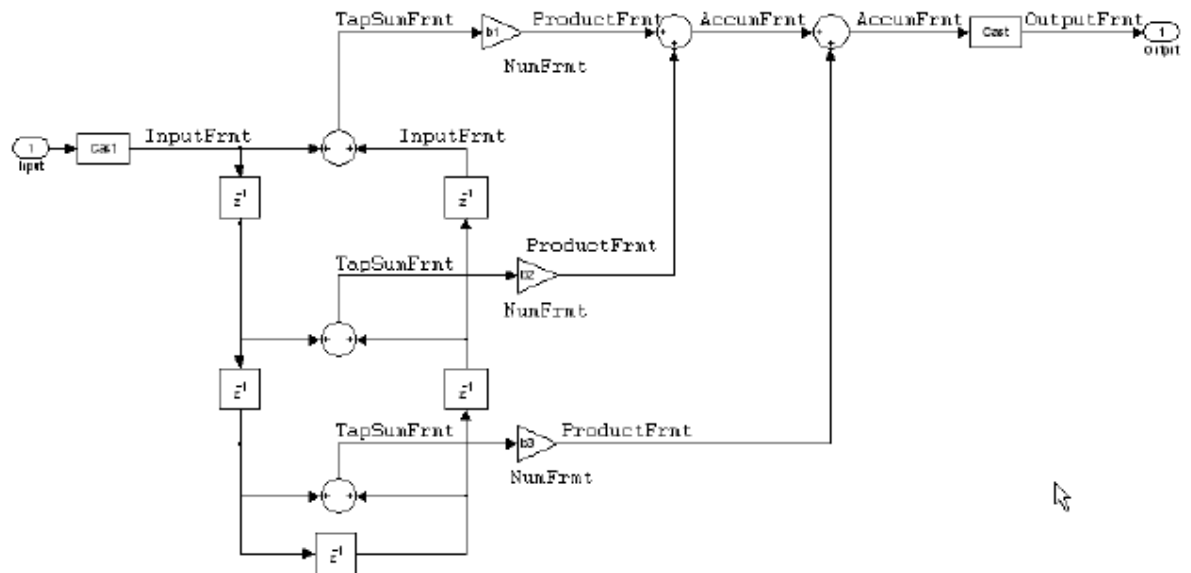
FILTRO LATTICE MOVING AVERAGE (MA) PARA FASE MÍNIMA:



FILTRO LATTICE MOVING AVERAGE AUTOREGRESIVO (ARMA):

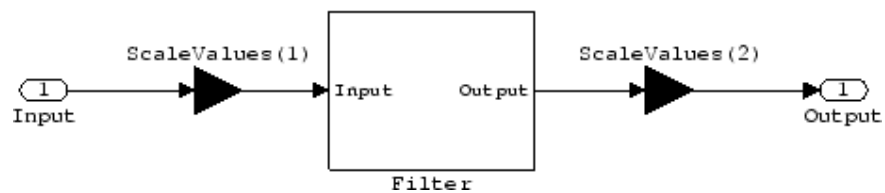


FILTRO FIR SIMÉTRICO DE CUALQUIER ORDEN:



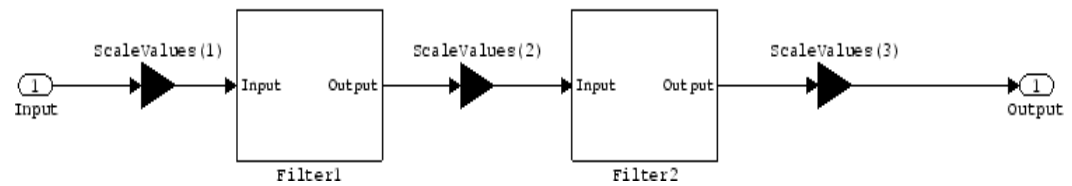
APLICACIÓN DE LA PROPIEDAD SCALE VALUES AL CUANTIZAR UN FILTRO CON UNA SOLA SECCIÓN:

Application of ScaleValues to a Single Section

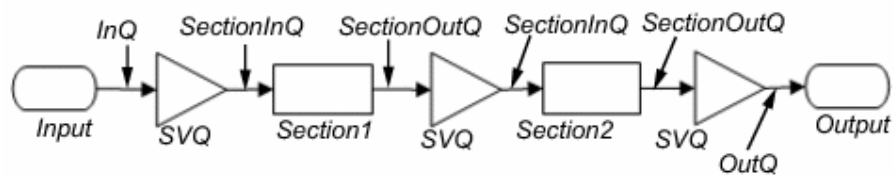
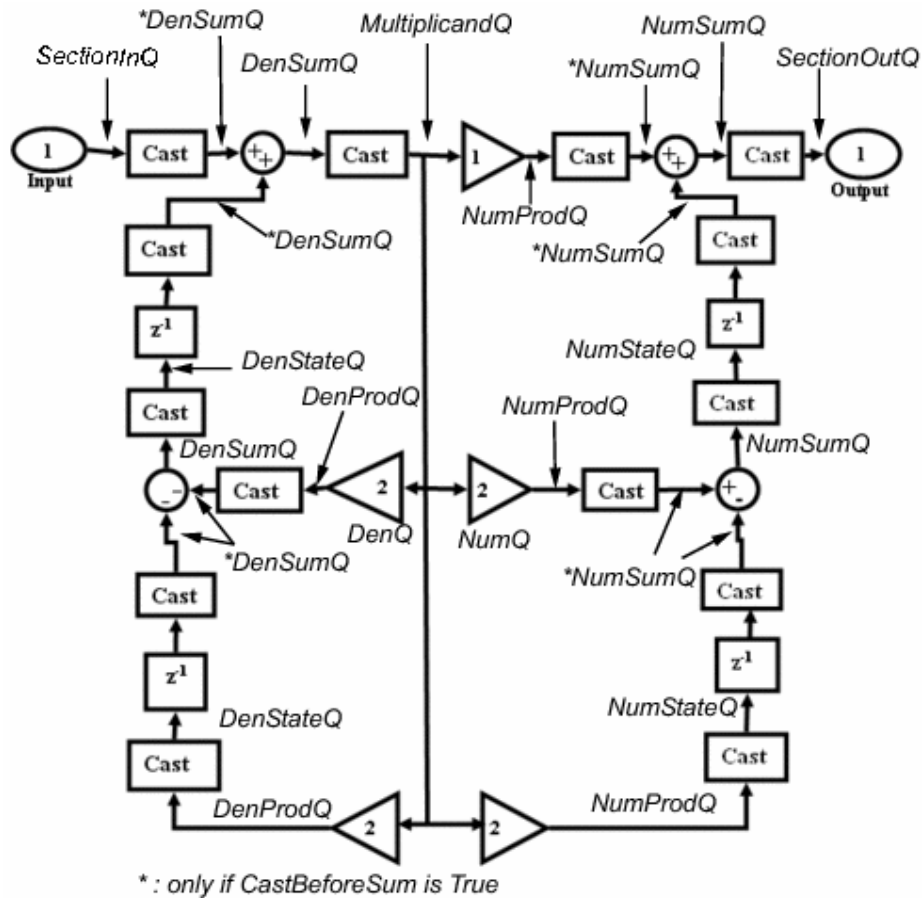


APLICACIÓN DE LA PROPIEDAD SCALE VALUES AL CUANTIZAR UN FILTRO CON DOS SECCIONES:

Application of ScaleValues to Multiple Sections



FILTRO SUS COMPUESTOS EN SECCIONES:



B.24 PROPIEDAD DE FILTROS ADAPTIVOS

B.24.1 SISTEMA DE LAS PROPIEDADES:

Propiedad	Descripción
Algoritmo	Reporta el algoritmo que el objeto usa para la adaptación. Cuando se construye un filtro adoptivo, esta propiedad de establecer automáticamente por el constructor como adaptfilt.nlms, la creación de un filtro adoptivo es utilizado en el algoritmo LMS normalizado. No se puede cambiar el valor sólo es de lectura.

Avg Factor	Factor promedio utilizado para calcular la estimación exponencial de la potencia en la señal bins transformado por las actualizaciones del coeficiente. Avg factor debe oscilar entre 0 y 1. Para filtrar objetos por defecto el avg factor debe ser igual a 1. Lambda (λ) es el argumento que representa a la entrada del Avg factor.
Bkwd Pred Error Power	Devuelve la media cuadrada mínima de la predicción del error.
Bkwd Prediction	Retorna las predicciones de las muestras generales durante la adaptación.
Blacklength	Longitud de bloque para el coeficiente de actualizaciones. Este debe ser un número entero positivo, tal que 1) blacklength, también sea un entero.
Coefficients	Vector que contiene los coeficientes del filtro inicial. Debe ser un vector de longitud L (e), donde e es el número de coeficientes del filtro.
Conversión factor	Factor de conversión, por defecto es una matriz [1x1]
Delay	Actualiza el demoramiento en tiempo de muestreo.
Desired Signal _____	Señal deseada de estados en la adaptación de filtros. El valor por defecto es 0 con un vector de longitud [blocken-1] ó (swbluckden-1) dependiendo del algoritmo.
Epsilon status	Vector de los valores de epsilon de los filtros adoptivos. Su valor por defecto es 0.
Error status	Vector de el error de estado de un filtro adoptivo, por defecto su valor es 0.
FFT Coefficients	Almacena los coeficientes del filtro que se encuentran de la forma de la transformada discreta de Fourier.
FFT status	Almacena los estados de la FFT de los coeficientes del filtro durante la adaptación.
Fi herd imput status	Vector de la entrada de estados del filtro con longitud igual de -1 a 1.
Filteclength	Contiene la longitud del filtro, no es el orden del filtro. La longitud del filtro menos uno es el orden del filtro. Por ejemplo, una longitud de 10 el filtro es de orden 9.
Forgetting factor	Determina la forma en que el filtro, adaptivo RLS utiliza datos en cada iteración.
Fiwdired error power	Devuelve la media cuadrada mínima del error de predicción en dirección de avance.
Fwd predicción	Contiene los valores previstos para las muestras durante la adaptación.
Init factor	Factor limitado.
Inucov	Raíz cuadrada de la matriz del factor de covarianza de entrada.

Kalmangain	Vacía el objeto construido.
Kalmangain states	Contiene los estados de la ganancia de Kalman actualizada durante la adaptación.
Leakage	Contiene el Leakage en el algoritmo del filtro adaptivo.
Offset cov	Contiene la matriz de la covarianza del offset
Offset	Especifica la opción de ofset para el denominador de los términos de los pasos normalizados.
Power	Vector de potencia de elementos 2 x 1
Projection orde	Orden de la proyección de los algoritmos afin, define el tamaño de la señal de entrada de la matriz de covarianza.
Reflection coeffs	Coefficientes determinados por la parte de la reflexión del filtro durante la adaptación.
Reflection coeffs sstep	Tamaño del paso usado para determinar los coeficientes reflexionados.
Persisten memory	Especifica si los estados del filtro se reestablecen y se activa la memoria antes de cada operación de filtrado (retiene o no los estados)
Secondary path coeffs	Un vector que contiene los valores del coeficiente de path secundarios desde la salida del activador hacia el sensor de error.
Secondary path estimate	Estimación del path secundario del modelo de filtro.
Secondary path state	Estados de path secundario del filtro.
Sqrt cov	Raíz cuadrada de la entrada de la matriz covariante.
Sqrtinvcov	Raíz cuadrada de la inversa de la señal de entrada de la matriz covariante.
States	Vector de los estados del filtro adaptivo.
Stepsize	Tamaño de las palabras de las iteracciones del proceso del filtro adaptivo.
Swblocklength	Bloque de longitud.

B.25 PROPIEDAD DE LOS FILTROS MULTIRATA

B.25.1 SÍNTESIS DE LAS PROPIEDADES:

Nombre	Valor	Default	Descripción
Blacklength	Enteros positivos	100	Longitud de cada bloque de datos de entrada del fft usando en el filtro
Decimation factor	Solo enteros positivos	2	Monto de reducir la entrada de rata del muestreo

Differential delay	Cualquier entero	1	Establece el retardo diferencial para el filtro. Usualmente es un valor de uno o dos es apropiado.
Filter intervals	Full precision min wordlength specific Word-length	Full precision	Controla si el filtro de salida establece la longitud de palabra y la longitud de fracción y el acumulador de longitud de palabra y de fracción mantiene automáticamente la mejor precisión de los resultados durante el filtrado.
Filter structure	Mfilt estructura de cadena	Ninguno	Describe el flujo de señal para el objeto del filtro, incluye todos los elementos activos para realizar durante las operaciones de filtrado.
Input offset	Enteros	0	Contiene el número de datos de entrada de muestras procesadas sin la generación de las muestras de salidas.
Interpolation factor	Enteros positivos	2	Factor de interpolación del filtro L especifica la cantidad de aumento de la tasa de muestra de entrada.
Number of sections	Enteros positivos	2	Número de

			secciones utilizadas es el decimador y en el integrador de los filtros CIE.
Numerator	Valor doble de array	No hay valores	Vector que contiene los coeficientes del filtro lowpass FIR, utilizado para la interpolación.
Overflow mode	Saturado wap	Wap	Establece el modo utilizado para responder a las condiciones del overflow en la aritmética del punto fijo.
Polyphase accum	Valores dependiendo del tiempo del filtro	0	Almacenar el valor restante en el acumulador del filtro después de los procesos de la última entrada de la muestra.
Persistent memory	Falso o verdadero	Falso	Determina si los filtros de los estados para obtener restaurado a partir de los valores para cada operación del filtrado.
Rate change factors	[L, m]	[2, 3] [3, 2]	Informa la decimación (m) e interpolación (l) como factor del objeto del filtro.
States	Cualquier matriz m+1 hasta n de valores dobles	[2 x 2] Int 32	Almacena condiciones para el filtro. Incluido los valores para las secciones del integrados. N = # del filtro m = # de

				secciones
Section length	Word length	Min Word length specify Word length	Minword length	Determina si el filtro (objeto) establece la longitud de palabra o de la sección que proporciona la longitud de palabra explícitamente.
Specify length	Word length	Vector de enteros [16 16 16 16] bits		
Word persection	length	Cualquier entero o vector de longitud $2n$	16	Define la longitud de palabra utilizando en cada sección, n es el número de secciones.