

CAPÍTULO 4

-ESTRUCTURAS PARA LOS SISTEMAS MULTIFRECUENCIA-

4.1 INTRODUCCIÓN

En los capítulos anteriores presentamos un tratado de los dispositivos involucrados en sistemas para la conversión de la frecuencia de muestreo. Es fácil entender la necesidad del estudio de estructuras para realizar la conversión de la frecuencia de muestreo, como en el caso de la Figura 4.1 donde esta estructura puede convertir la frecuencia de muestreo por un factor de L/D , donde L y D representan los factores de *upsampling* y *downsampling* respectivamente.

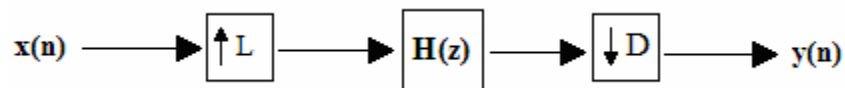


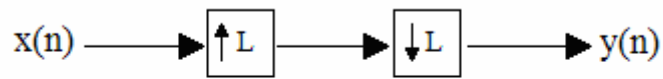
Figura 4.1: Conversión de frecuencia racional

Como se discutió en el Capítulo 3, el modelo teórico para este sistema es que incrementa la frecuencia de muestreo de la señal por un factor L (al llenar $L-1$ muestras de valor cero entre cada muestra de $x(n)$), filtra la señal resultante para eliminar las réplicas no deseadas y comprime la señal por un factor D (al retener una de cada D muestras).

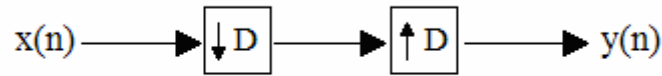
Una implementación directa del sistema mostrado en la Figura 4.1 es ineficiente, ya que el filtro pasabajos está operando a una frecuencia de muestreo alta en una señal para la cual $L-1$ muestras son de valor cero, y los valores de la salida filtrada son requeridos solamente una vez cada D muestras. Para este ejemplo se puede aplicar el conocimiento ya adquirido en los capítulos anteriores para implementar el sistema de la Figura 4.1 de una forma más eficiente (en términos de multiplicaciones por unidad de tiempo).

4.2 INTERCONEXIÓN EN CASCADA

Primero consideremos la conexión en cascada de un upsampler seguido de un downsampler, y supongamos que los factores D y L son iguales es decir $D=L$ como se muestra en la Figura 4.2. Sería demasiado rápido asegurar que la salida es igual a la entrada.



(a) Upsampler-Downsampler



(b) Downsampler-Upsampler

Figura 4.2: Conexiones cascada de los Convertidores

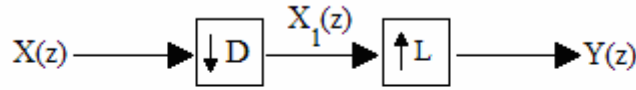
El proceso de *upsampling* con factor L introduce $L-1$ ceros después de cada muestra de la señal de entrada, y el *downsampling* por el mismo factor L remueve esos ceros y el resultado es la señal original. Si el *upsampling* precede al *downsampling* con el mismo factor, la señal no cambia, por lo cual en la práctica esta conexión no tiene sentido [4].

La Figura 4.2(b) muestra una conexión en cascada frecuentemente encontrada en bancos de filtros. Sabemos que en el *downsampling*, las muestras que no son múltiplos enteros de D , se convierten en cero e inmediatamente después se descartan esos valores para que la señal finalmente quede comprimida. El *upsampling* por D introduce nuevamente esos ceros, y por lo tanto la señal resultante queda como el primer paso que se realiza en el *downsampling*, es decir, el resultado de la señal original multiplicada por el tren de impulsos con la frecuencia D [4]. La señal $y(n)$ es igual a la señal $x(n)$ cuando n es múltiplo de D , y cero para otros valores. Si el *downsampling* lo realizamos antes que el *upsampling*, con el mismo factor, la señal de salida será diferente a la señal de entrada.

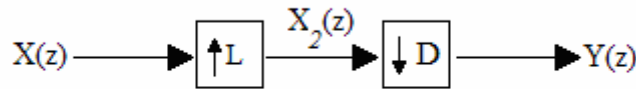
4.3 INTERCAMBIO ENTRE LOS CONVERTIDORES DE LA FRECUENCIA DE MUESTREO

Vamos a analizar si las operaciones de *downsampling* y *upsampling* pueden ser intercambiadas, es decir, si son conmutativas o no. La estructura (a) en la Figura 4.3

muestra la operación de *downsampling* por un factor D , seguido del *upsampling* por un factor L .



(a) *Downsampler-Upsampler*



(b) *Upsampler-Downsampler*

Figura 4.3: Intercambio de los convertidores de la Frecuencia de Muestra

La pregunta es cuándo podemos realizar el *upsampling* por un factor L antes que el *downsampling* por un factor M (Figura 4.3(b)) y obtener el mismo resultado. La respuesta se obtiene al analizar las estructuras mostradas en la Figura 4.3.

Para la Estructura (a):

$$\begin{aligned}
 X_1(z) &= \frac{1}{M} \sum_{k=0}^{M-1} X(z^{1/M} W_M^k) \\
 Y_1(z) &= X_1(z^L) = \frac{1}{M} \sum_{k=0}^{M-1} X(z^{L/M} W_M^k)
 \end{aligned} \tag{1}$$

Para la estructura (b):

$$\begin{aligned}
 X_2(z) &= X(z^L) \\
 Y_2(z^M) &= \frac{1}{M} \sum_{k=0}^{M-1} X_2(z W_M^k) = \frac{1}{M} \sum_{k=0}^{M-1} X(z^L W_M^{kL}) \\
 Y_2(z) &= \frac{1}{M} \sum_{k=0}^{M-1} X(z^{L/M} W_M^{kL})
 \end{aligned} \tag{2}$$

Como se puede observar al comparar las ecuaciones (1) y (2), un *downsampler* puede ser intercambiado con un *upsampler* si y sólo si $Y_1(z)=Y_2(z)$ y esto sólo ocurre cuando [4]:

$$W_M^k = W_M^{kL} \tag{3}$$

Esta relación se mantiene si y sólo si M y L no son mutuamente múltiplos, es decir, que M y L no tienen ningún factor común entero más que 1 [2].

Ejemplo 4.1

Ilustraremos el resultado formal al decir que $L=2$ y $M=3$, ya que no son números mutuamente múltiplos.

$$\begin{aligned} \left\{ W_3^k = e^{\frac{-j2\pi k}{3}} ; k = 0,1,2 \right\} &= \left\{ e^{-j0} ; e^{\frac{-j2\pi}{3}} ; e^{\frac{-j4\pi}{3}} \right\} \\ \left\{ W_3^{2k} = e^{\frac{-j2\pi k 2}{3}} ; k = 0,1,2 \right\} &= \left\{ e^{-j0} ; e^{\frac{-j4\pi}{3}} ; e^{\frac{-j8\pi}{3}} = e^{\frac{-j2\pi}{3}} \right\} \end{aligned} \quad (4)$$

Como era de esperarse, las ecuaciones en (4) son equivalentes. Ahora sea $L=2$ y $M=4$, en este caso, M y L son mutuamente múltiplos ya que $M=2L$. Los pares correspondientes son:

$$\begin{aligned} \left\{ W_4^k = e^{\frac{-j\pi k}{2}} ; k = 0,1,2,3 \right\} &= \left\{ e^{-j0} ; e^{\frac{-j\pi}{2}} ; e^{-j\pi} , e^{\frac{-j3\pi}{2}} \right\} \\ \left\{ W_4^{2k} = e^{-j\pi k} ; k = 0,1,2,3 \right\} &= \left\{ e^{-j0} ; e^{-j\pi} ; e^{-j2\pi} ; e^{-j3\pi} \right\} \end{aligned} \quad (5)$$

Como se puede observar en (5) las ecuaciones no son equivalentes. En resumen podemos decir que el *upsampler* y el *downsampler* pueden ser intercambiados si y sólo si M y L no son múltiplo uno del otro.

4.4 IDENTIDADES NOBLES

Hasta el momento hemos visto conexiones cascada de *downsamplers* y *upsamplers* con sistemas LIT (Lineales e Invariantes en el Tiempo). Una diferente conexión en cascada se muestra en la Figura 4.4(a) donde un filtro $H(z)$ va después de un *downsampler*, o en la Figura 4.4(d) donde un filtro $H(z)$ precede un *upsampler*. Éstas interconexiones se presentan cuando usamos las *representaciones polifase* (sección 4.5)

para filtros de decimación e interpolación. Si la función $H(z)$ es racional, podemos redibujar la Figura 4.4(a) como la Figura 4.4(b) y también podemos redibujar la Figura 4.4(c) como la Figura 4.4(d). A estas estructuras se les conoce como *identidades nobles* y son muy útiles en la teoría e implementación de los sistemas multifrecuencia [3].

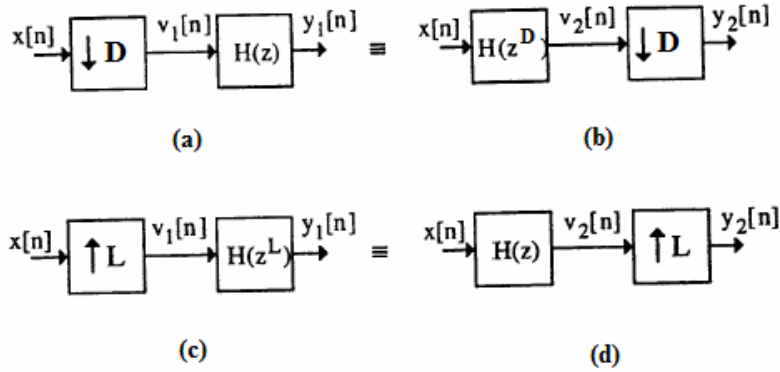


Figura 4.4: Las identidades nobles para los sistemas multifrecuencia

A continuación verificaremos las identidades.

Consideremos la estructura de la Figura 4.4(a), podemos ver que [3]:

$$V_1(z) = \frac{1}{D} \sum_{k=0}^{D-1} X(z \frac{1}{D} W_D^k); \quad Y_1(z) = V_1(z)H(z) = \frac{1}{D} \sum_{k=0}^{D-1} X(z \frac{1}{D} W_D^k)H(z) \quad (6)$$

Si observamos a la Figura 4.4(b) podemos darnos cuenta que:

$$V_2(z) = X(z)H(z^D); \quad Y_2(z) = \frac{1}{D} \sum_{k=0}^{D-1} X(z \frac{1}{D} W_D^k)G((z \frac{1}{D} W_D^k)^D) = \frac{1}{D} \sum_{k=0}^{D-1} X(z \frac{1}{D} W_D^k)G(z W_D^{kD}); \quad (7)$$

Por otro lado sabemos que [3]:

$$W_D^{kD} = e^{\frac{-j2\pi kD}{D}} = 1; \quad \text{por lo que la ecuación (7) queda finalmente como:}$$

$$Y_2(z) = \frac{1}{D} \sum_{k=0}^{D-1} X(z \frac{1}{D} W_D^k)G(z) \quad (8)$$

Y podemos ver que las ecuaciones (6) y (8) son iguales por lo que las estructuras de la Figura 4.4(a) y 4.4(b) son equivalentes. De igual manera podemos ver que las estructuras mostradas en la Figura 4.4(c) y 4.4(d) son equivalentes, ya que podemos observar que en la Figura 4.4(c):

$$Y_1(z) = H(z^L)V_1(Z) = H(z^L)X(z^L) \quad (9)$$

Mientras que en la Figura 4(d):

$$V_2(z) = X(z)H(z) \therefore Y_2(z) = X(z^L)H(z^L) \quad (10)$$

Podemos observar que las ecuaciones (9) y (10) son iguales por lo que los sistemas mostrados en la Figura 4(c) y 4(d) son equivalentes.

4.5 FILTROS POLIFASE

4.5.1 DECIMACIÓN POLIFASE

Un importante avance en el procesamiento de señales multifrecuencia es la invención de la *representación polifase*. Esto permite una gran simplificación de los resultados teóricos y esto lleva a eficientes implementaciones de filtros para la decimación/interpolación; así como para los bancos de filtros.

Supongamos un filtro FIR con N coeficientes, donde N es un múltiplo entero de M , podemos obtener M diferentes componentes de muestras discretas de la respuesta al impulso. Eso significa que las correspondientes *transformadas* z pueden ser particionadas en M sub-señales. Por ejemplo, para $N=12$ y $D=4$ podemos escribir:

$$\begin{aligned} H(z) = \sum_{n=0}^{N-1} h(n)z^{-n} &= h(0)z^0 + h(1)z^{-1} + h(2)z^{-2} + h(3)z^{-3} + h(4)z^{-4} + h(5)z^{-5} + h(6)z^{-6} + \\ &h(7)z^{-7} + h(8)z^{-8} + h(9)z^{-9} + h(10)z^{-10} + h(11)z^{-11} = \\ &z^{-0} [h(0)z^0 + h(4)z^{-4} + h(8)z^{-8}] + z^{-1} [h(1)z^{-0} + h(5)z^{-4} + h(9)z^{-8}] + \\ &z^{-2} [h(2)z^{-0} + h(6)z^{-4} + h(10)z^{-8}] + z^{-3} [h(3)z^{-0} + h(7)z^{-4} + h(11)z^{-8}] \end{aligned} \quad (11)$$

Esta ecuación se puede reescribir como:

$$H(z) = \sum_{k=0}^3 z^{-k} H_k(z^4), \quad (12)$$

donde

$$H_k(z^4) = \sum_{n=0}^2 h(4n+k)(z^4)^{-n}; \quad k=0,1,2,3\dots \quad (13)$$

Podemos generalizar la expresión (12) para obtener:

$$H(z) = \sum_{k=0}^{D-1} z^{-k} H_k(z^D), \quad (14)$$

donde

$$H_k(z^D) = \sum_{n=0}^{N/D-1} h(nD+k)(z^D)^{-n}; \quad k=0,\dots,D-1 \quad (15)$$

Y

$$H_k(z) = \sum_{n=0}^{N/D-1} h(nD+k)(z)^{-n}; \quad k=0,\dots,D-1 \quad (16)$$

Se llama *realización polifase* cuando se hace una realización de $H(z)$ basada en la descomposición de (14) y señala las bases para la *decimación polifase* [1].

La ecuación (15) representa los *componentes polifase* del filtro. Usando la relación (14), la estructura de decimación de la Figura 1.1 del Capítulo 2, puede ser redibujada como se muestra en la Figura 4.5(a). Usando el principio de superposición y las *identidades nobles* podemos obtener una versión más eficiente que si se implementara de la forma tradicional, en la cual tanto el número de operaciones de filtrado como la cantidad de memoria requerida son reducidos. La estructura mencionada se muestra en la Figura 4.5(b), como podemos observar, los sub-filtros son los componentes polifase (16) del filtro de decimación.

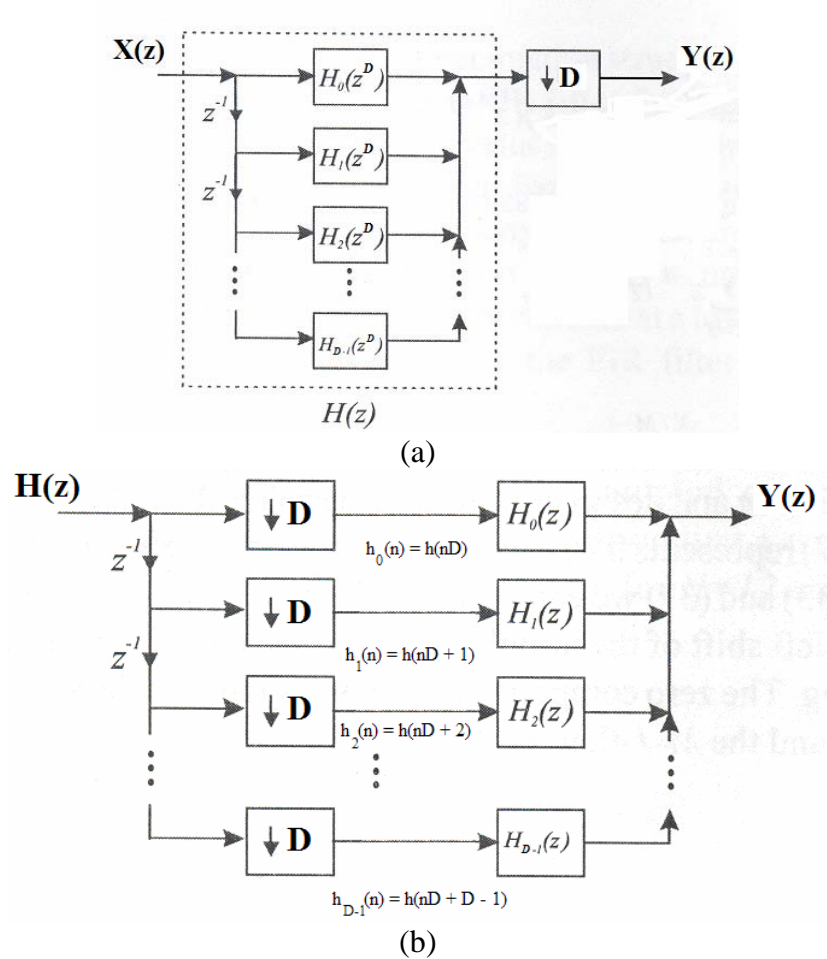


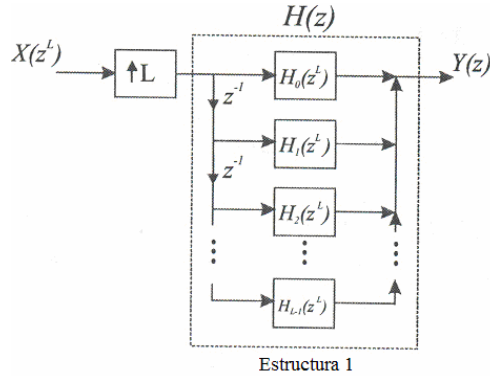
Figura 4.5: Decimación Polifase

4.5.2 INTERPOLACIÓN POLIFASE

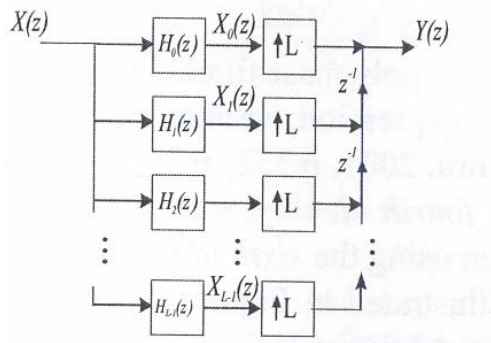
Consideremos un filtro de interpolación como el que se mostró en la Figura 3.1 del Capítulo 3, una implementación directa de $H(z)$ sería ineficiente ya que, no más del 50% de las muestras de entrada a $H(z)$ son valores diferentes de cero, en el proceso de convolución hay muchas multiplicaciones innecesarias con valores cero, la convolución que se realiza a una frecuencia mayor a la original, puede ser reemplazada por convoluciones independientes, a una frecuencia de muestreo menor usando la descomposición polifase (14), podemos reemplazar los coeficientes M por L para obtener [3]:

$$H(z) = \sum_{k=0}^{L-1} z^{-k} H_k(z^L) \quad (17)$$

Si expresamos el filtro polifase en la estructura de interpolación de la Figura 2.1 del Capítulo 2, usando la expresión (15), obtenemos la estructura mostrada en la Figura 4.6(a) [4]



(a)



(b)

Figura 4.6: Interpolación Polifase

De acuerdo con la identidad vista en el Capítulo 3 para el *upsampler*, y mediante la identidad noble para el *upsampler*, podemos redibujar la estructura de la Figura 4.6(a) como se muestra en la Figura 4.6(b).

Cada muestra de entrada es simultáneamente enviada a los subfiltros $H_0(z), \dots, H_{L-1}(z)$, donde el filtrado se desarrolla a una frecuencia de muestreo menor, y da como resultado los componentes $X_0(z), \dots, X_{L-1}(z)$. Estos componentes pasan por el proceso de *upsampling* (se agregan ceros), y son intercalados con retrasos, y finalmente son sumados para formar la señal de salida.

Ejemplo 4.2

Una estructura más eficiente para el interpolador y decimador, puede ser realizada cuando se diseña un filtro con coeficientes simétricos, es decir que los coeficientes presenten la forma:

$$H(z) = h(0) + h(1)z^{-1} + h(2)z^{-2} + \dots + h(M-2)z^{-(M-2)} + h(M-1)z^{-(M-1)}$$

Una realización eficiente para un decimador con un factor D=4, y un filtro FIR de longitud 15, puede ser obtenido al realizar una descomposición polifase de 4 ramas de H(z), de la siguiente manera [5]:

$$H(z) = H_0(z^4) + z^{-1}H_1(z^4) + z^{-2}H_2(z^4) + z^{-3}H_3(z^4) \quad (18)$$

Si aplicamos las propiedades antes vistas para el downsampler y lo desplazamos a través de los filtros polifases, la estructura se puede ver como:

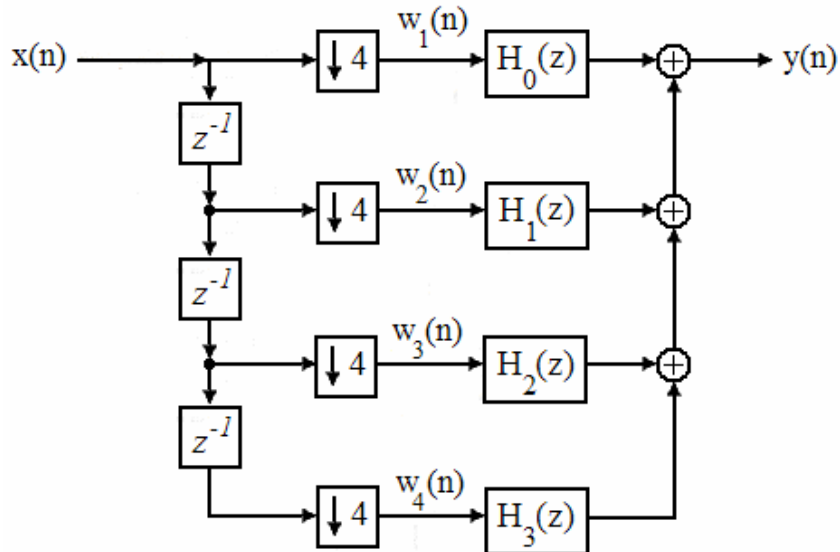


Figura 4.7: Estructura para Decimación

Por ejemplo si diseñamos la función de transferencia para un filtro FIR de longitud 15 usando coeficientes simétricos y haciendo uso de la descomposición polifase, queda de la siguiente manera:

$$H(z) = h(0) + h(1)z^{-1} + h(2)z^{-2} + h(3)z^{-3} + h(4)z^{-4} + h(5)z^{-5} + h(6)z^{-6} + h(7)z^{-7} + h(7)z^{-8} \\ + h(6)z^{-9} + h(5)z^{-10} + h(4)z^{-11} + h(3)z^{-12} + h(2)z^{-13} + h(1)z^{-14} + h(0)z^{-15} \quad (19)$$

Al usar las ecuaciones (14) y (16) tenemos que:

$$H_0(z) = h(0) + h(4)z^{-1} + h(7)z^{-2} + h(3)z^{-3} \\ H_1(z) = h(1) + h(5)z^{-1} + h(6)z^{-2} + h(2)z^{-3} \\ H_2(z) = h(2) + h(6)z^{-1} + h(5)z^{-2} + h(1)z^{-3} \\ H_3(z) = h(3) + h(7)z^{-1} + h(4)z^{-2} + h(0)z^{-3} \quad (20)$$

De la Figura 4.7 podemos observar que:

$$Y(z) = H_0(z)W_1(z) + H_1(z)W_2(z) + H_2(z)W_3(z) + H_3(z)W_4(z) \\ = h(0)(W_1(z) + z^{-3}W_4(z)) + h(4)(z^{-1}W_1(z) + z^{-2}W_4(z)) + h(7)(z^{-2}W_1(z) + z^{-1}W_4(z)) \\ + h(3)(z^{-3}W_1(z) + W_4(z)) + h(1)(W_2(z) + z^{-3}W_3(z)) + h(5)(z^{-1}W_2(z) + z^{-2}W_3(z)) \\ + h(6)(z^{-2}W_2(z) + z^{-1}W_3(z)) + h(2)(z^{-3}W_2(z) + W_3(z)) \quad (21)$$

Una estructura para un decimador computacionalmente eficiente con un factor $D = 4$ basado en la ecuación de arriba se muestra en la Figura 4.8.

Ejemplo 4.3

Una realización eficiente para un interpolador con un factor $L=3$, y un filtro FIR de longitud 15, puede ser obtenido al realizar una descomposición polifase de 3 ramas de $H(z)$, de la siguiente manera:

$$H(z) = R_2(z^3) + z^{-1}R_1(z^3) + z^{-2}R_0(z^3) \quad (22)$$

Y al mover el *up-sampler* a través del filtro polifase queda como se muestra en la Figura 4.9:

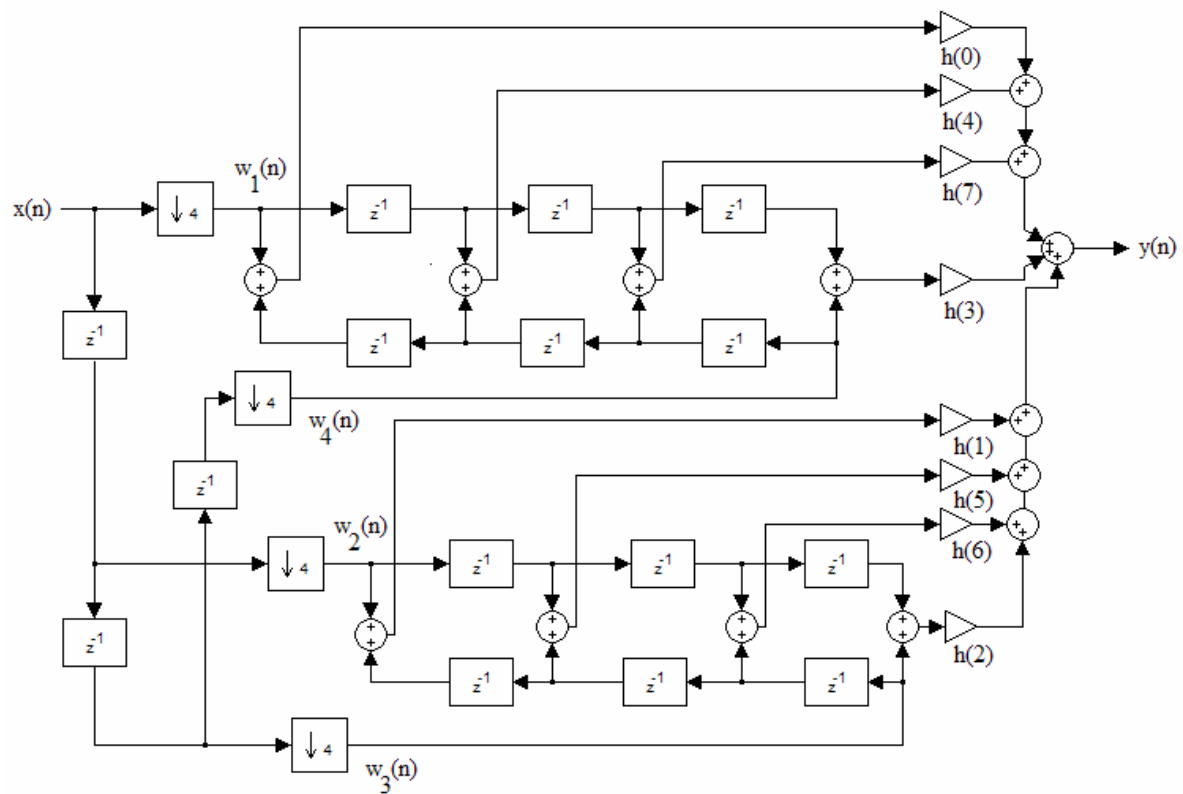


Figura 4.8: Realización Eficiente para Decimación

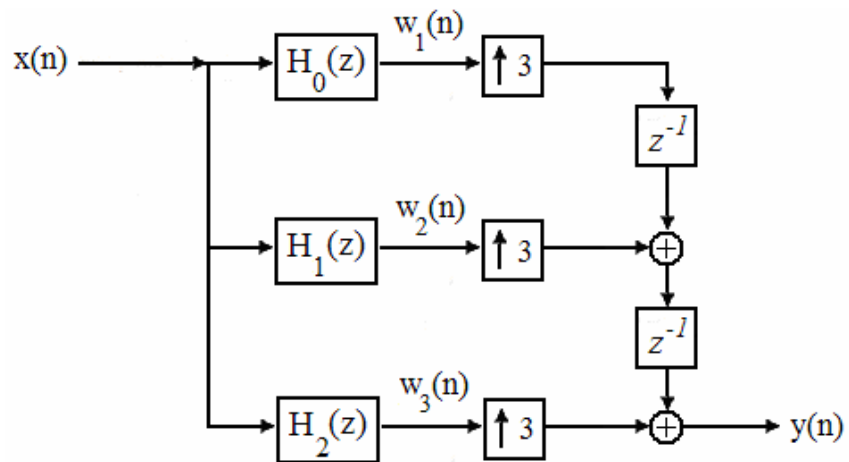


Figura 4.9: Estructura para Interpolación

Como en el decimador, podemos hacer un diseño más eficiente al diseñar un filtro con coeficientes simétricos, para éste ejemplo podemos obtener mediante las ecuaciones (14) y (16) las siguientes sumatorias:

$$H(z) = \sum_{k=0}^2 z^{-k} H_k(z^3) \quad (23)$$

$$H_k(z) = \sum_0^4 h(n3+k)(z)^{-n}; \quad k = 0,1,2. \quad (24)$$

Que nos llevan a los siguientes resultados:

$$\begin{aligned} W_3(z) &= h(0)X(z) + h(3)z^{-1}X(z) + h(6)z^{-2}X(z) + h(5)z^{-3}X(z) + h(2)z^{-4}X(z) \\ W_1(z) &= h(2)X(z) + h(5)z^{-1}X(z) + h(6)z^{-2}X(z) + h(3)z^{-3}X(z) + h(0)z^{-4}X(z) \\ W_2(z) &= h(1)(X(z) + z^{-4}X(z)) + h(4)(z^{-1}X(z) + z^{-3}X(z)) + h(7)z^{-2}X(z). \end{aligned} \quad (25)$$

Una estructura computacionalmente eficiente para un interpolador de factor 3, basándose en las ecuaciones anteriores se muestra a continuación:

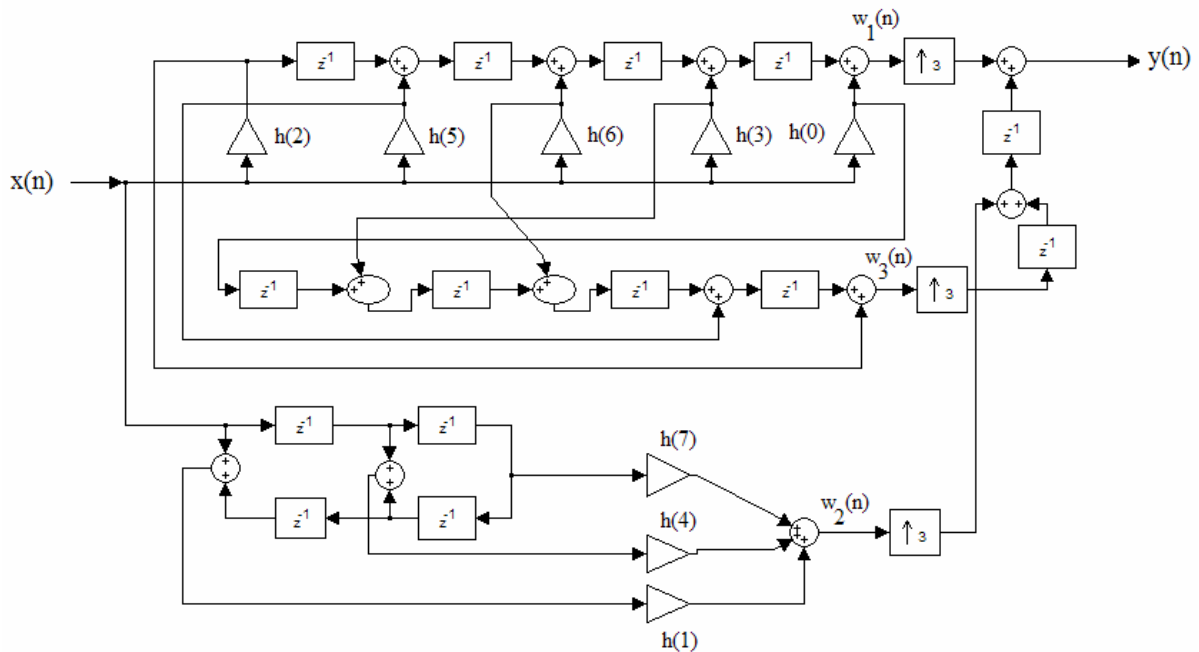


Figura 4.10: Realización Eficiente para Interpolación

4.6 SIMULACIÓN DE UN CAMBIO DE FRECUENCIA RACIONAL

A continuación se muestra la simulación de un sistema el cual hace un cambio de frecuencia racional mediante dos maneras diferentes, y ambos sistemas producen el mismo resultado.

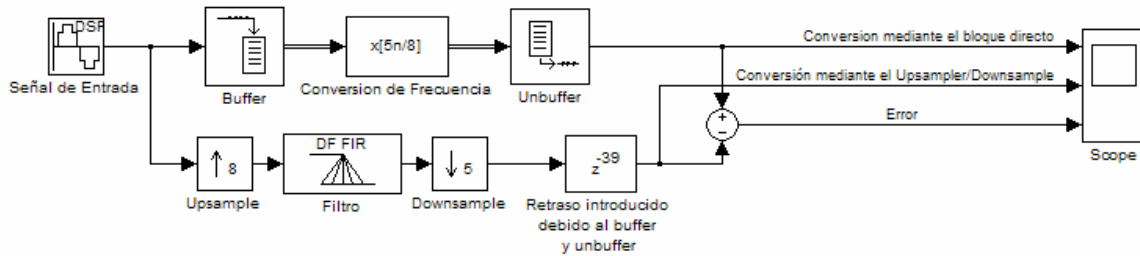


Figura 4.11: Realización Eficiente para Interpolación

Simulink tiene un bloque llamado “FIR Rate Conversion” que permite hacer la conversión racional especificando los factores “ L ” y “ M ” así como el filtro para dicha conversión. Dado que las muestras de la señal sinusoidal exceden la capacidad de entrada de datos al bloque “FIR Rate Conversion”, se tienen que almacenar en un bloque denominado “Buffer” el cual redistribuye las muestras de entrada a un nuevo tamaño de *frame*, mayor o menor al del *frame* de entrada [9].

Una vez que el tamaño del *frame* es el correcto, se pasa la salida al bloque “FIR Rate Conversion” el cual hace la conversión de frecuencia correspondiente y la salida la manda al bloque “Unbuffer” para que realice el proceso contrario al que hizo el “Buffer” y así tener la salida correcta.

Se puede observar que el proceso de conversión de frecuencia racional se puede llevar a cabo también mediante la conexión descrita en la Figura 4.1. El filtro para ambos casos fue utilizado para limitar la señal a $1/\max(M,L)$ donde M y L son los factores de *downsampling* y *upsampling* el orden del filtro fue elegido suficientemente grande (63) para evitar los problemas de *aliasing* e *imaging*, ya que un filtro de alto orden presenta

mayor calidad. En la segunda forma de implementar este cambio de frecuencia racional, se introdujo un retraso, esto debido a que a comparación de la primera forma, se tiene un retraso debido al proceso de *buffering* y al de *unbuffering*.

Los parámetros de simulación se muestran en la Tabla 2:

	Tipo	Frecuencia	Amplitud	Fs
Señal de entrada	Sinusoidal	0.1Hz	1	1 segundo

	Factor L	Factor M	Filtro
FIR Rate Conversion	8	5	<code>fir1(63,1/8)*8</code>

	Factor
Up-sampler	8
Down-sampler	5

	Tipo	Coeficientes del Numerador
Filtro	FIR	<code>fir1(63,1/8)*8</code>

	Tiempo de Simulación	Tipo	Modo
Parámetros de Simulación	0.0-99	Fixed-Step Discrete, auto	Single Tasking

Tabla 2: Parámetros de simulación para el sistema de conversión de frecuencia racional

A continuación se muestran las salidas correspondientes a este sistema:

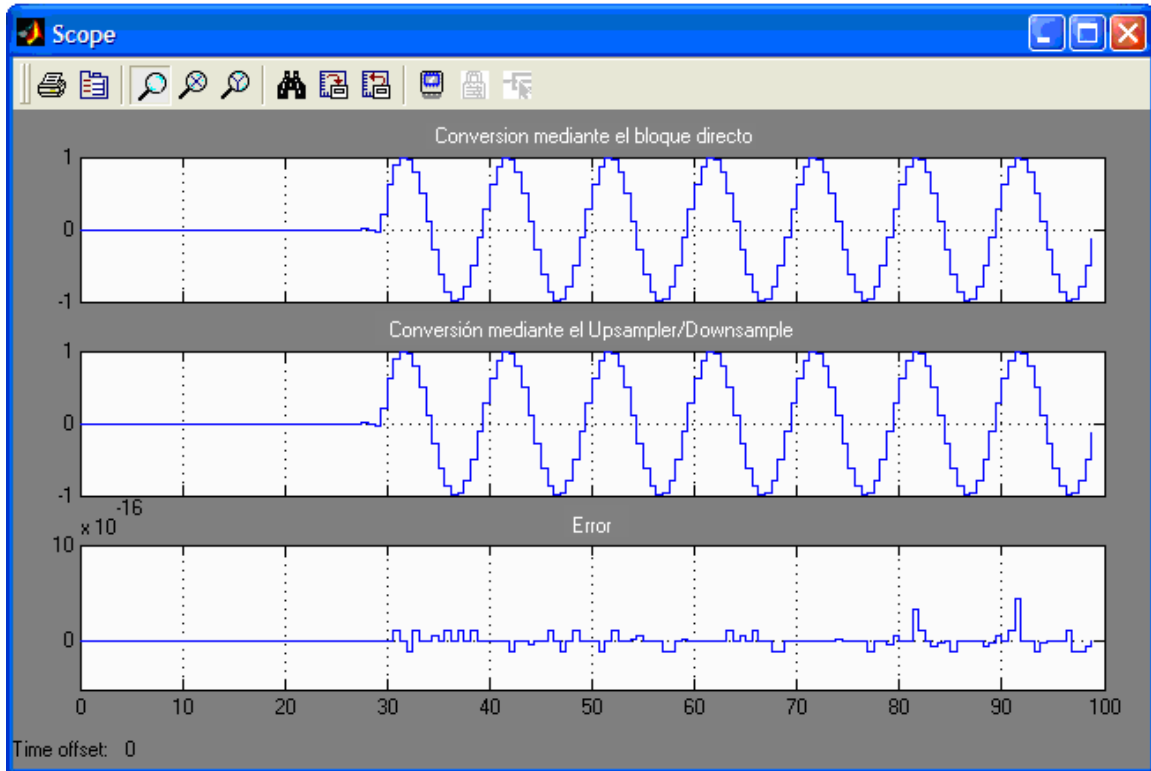


Figura 4.12: Salidas para el sistema de conversión de frecuencia racional

Como se puede observar, las salidas de ambos sistemas son casi idénticas, también se muestra la diferencia entre ambos.

Se puede observar que el bloque “FIR Rate Conversion” aunque tiene integrado el *upsampler* y el *downsampler*, es ineficaz, ya que introduce un gran retraso, por lo que la simulación con los sistemas de *upsampling* y *downsampling*, es mucho más eficaz.

SIMULACIÓN DE UN SISTEMA DE DECIMACIÓN EFICIENTE

En el ejemplo 4.2 se desarrolló y se presentó la estructura para un sistema eficiente de decimación con un factor $D=4$, el cual, fue diseñado para un filtro FIR simétrico de longitud 15.

A continuación se presenta la simulación del sistema descrito anteriormente, en la Tabla 3 se presentan los coeficientes del filtro pasabajas calculado, este filtro tiene una frecuencia de corte de 4.5Khz.

Coeficientes
$h(0) = h(14) = -0.014112893$
$h(1) = h(13) = -0.001945309$
$h(2) = h(12) = 0.040000004$
$h(3) = h(11) = 0.01223454$
$h(4) = h(10) = -0.09138802$
$h(5) = h(9) = -0.01808986$
$h(6) = h(8) = 0.3133176$
$h(7) = 0.52$

Tabla 3: Coeficientes de un filtro FIR de longitud 15

La señal de entrada es una señal sinusoidal con una frecuencia de 17Khz, una frecuencia de muestreo de $1.960784313725490e-005$, el sistema descrito se simuló como lo muestra la Figura 4.13:

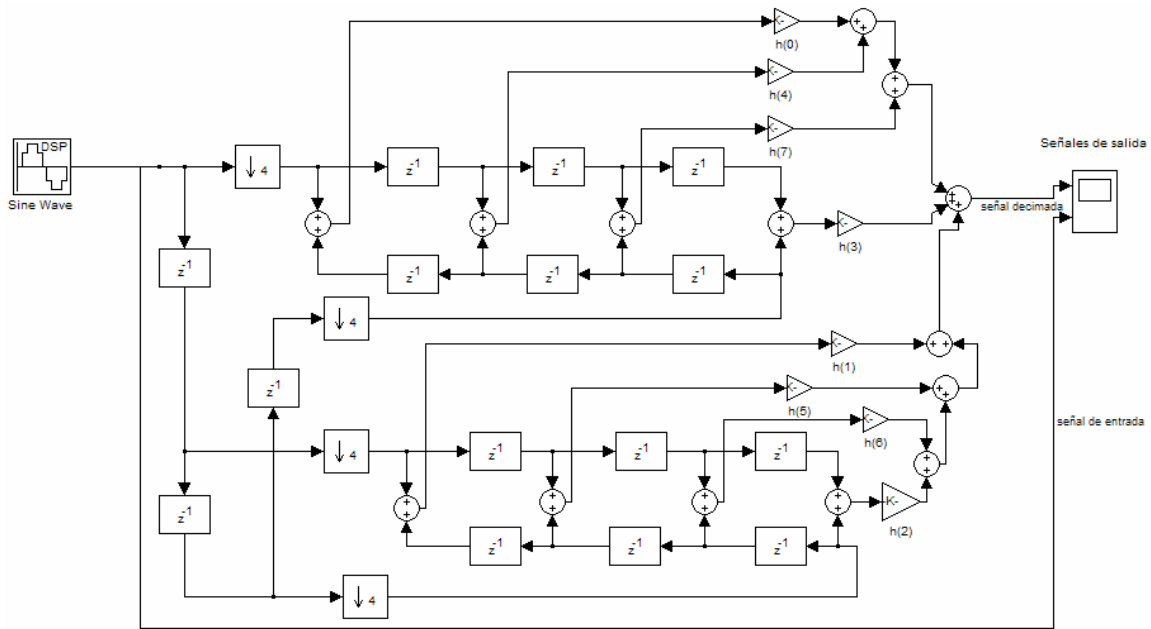
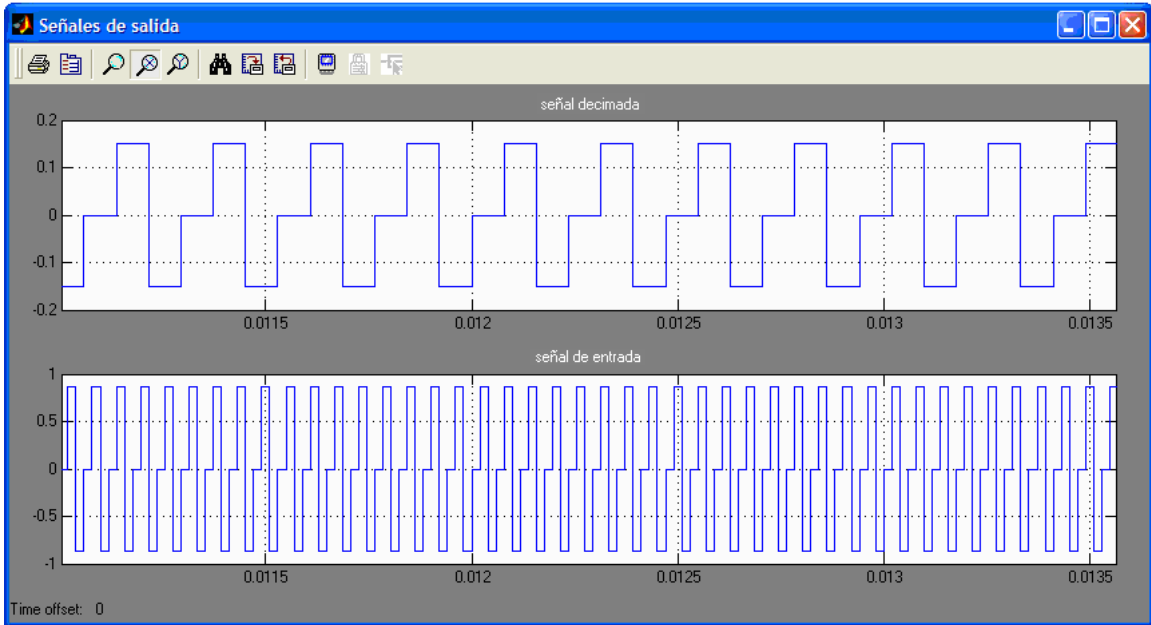


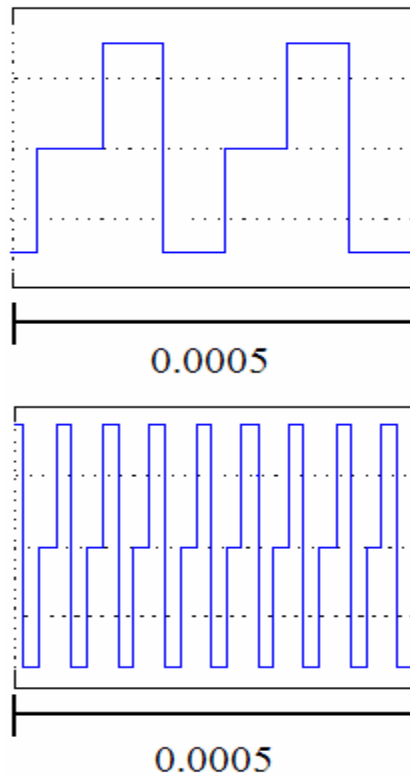
Figura 4.13: Simulación de un sistema de decimación eficiente

En la Figura 4.14(a) podemos observar las señal de salida que corresponden a la señal original y a la señal decimada, se puede observar que cada 0.0005 ocurren 8 muestras de la señal de entrada, mientras que para el mismo intervalo, en la señal decimada, solamente ocurren dos muestras como era de esperarse, ya que entre cada muestra tomada de la señal original, se descartan 3.

Así mismo se puede observar que la amplitud de la señal decimada es escalada por un factor de $\frac{1}{4}$ con respecto a la señal original.



(a) Salida del sistema



(b) Ampliación de la salida del sistema

Figura 4.14: Salidas para la simulación del sistema eficiente de decimación.