

## **CAPÍTULO 2 -DECIMACIÓN-**

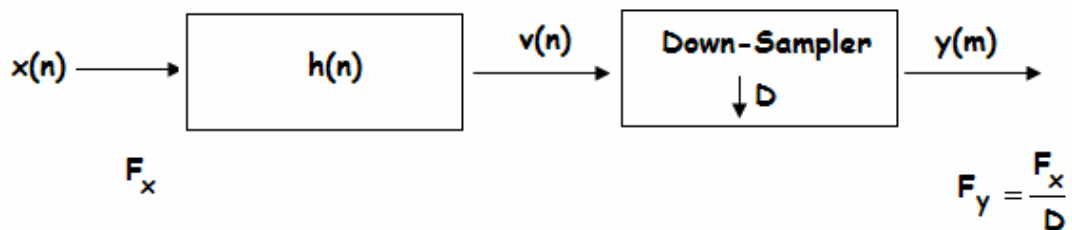
## 2.1 INTRODUCCIÓN

El proceso de convertir la frecuencia de una señal dada, a una frecuencia diferente, se le llama *conversión de la frecuencia de muestreo*. Los sistemas que emplean múltiples frecuencias de muestreo en el procesamiento de señales digitales son llamados *Sistemas de Procesamiento Digital de Señales Multifrecuencia*.

Este capítulo trata una de las operaciones básicas en el estudio de los Sistemas Multifrecuencia, la *decimación*. Se analizará en el dominio del tiempo y la frecuencia y se obtendrán relaciones de entrada y salida en el sistema, y finalmente se mostrarán dos simulaciones en Simulink.

## 2.2 DECIMACIÓN

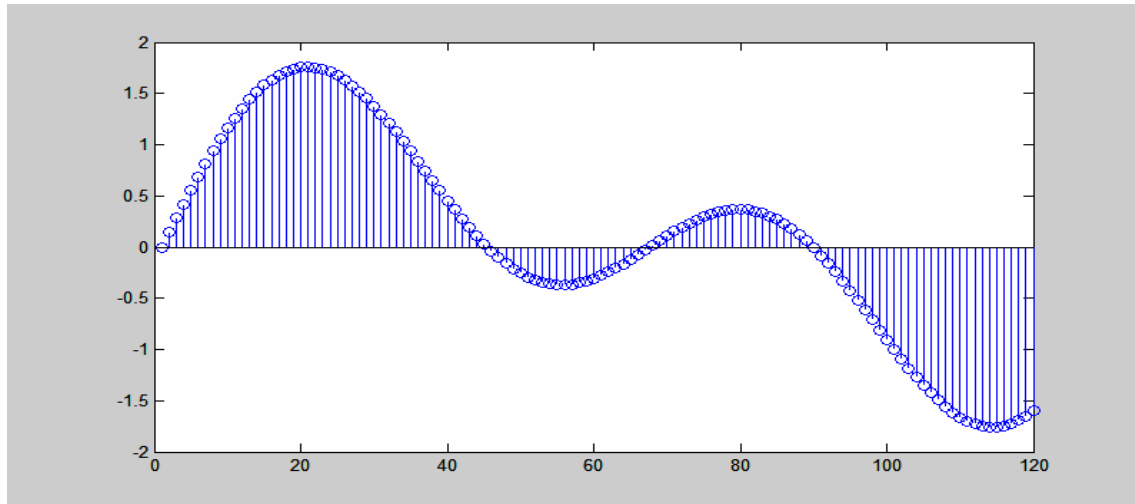
A la reducción de la frecuencia de muestreo se le llama *decimación*. La decimación consiste en dos etapas: filtrado y *downsampling* (muestreo hacia abajo) como se muestra en la Figura 2.1.



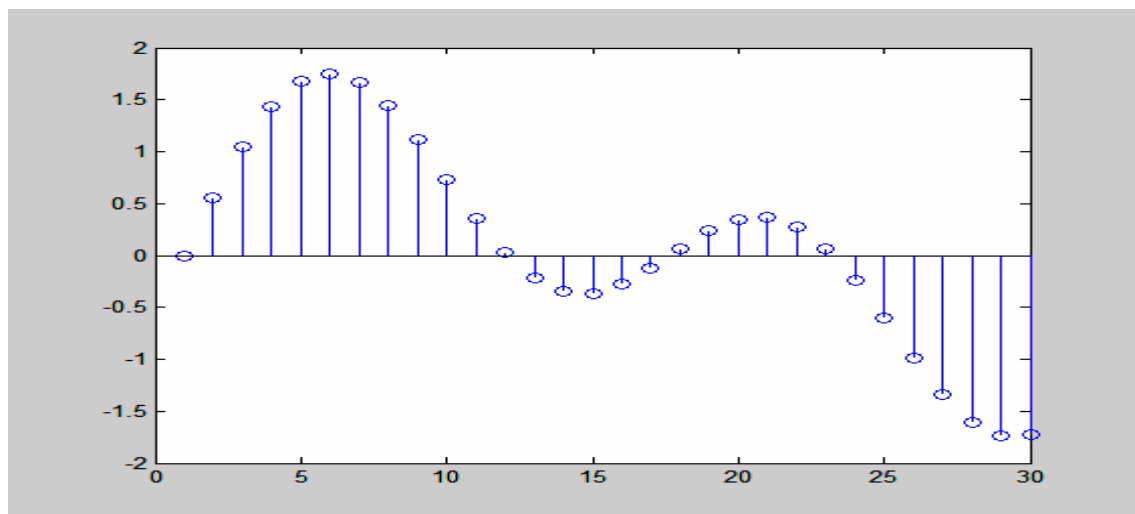
*Figura 2.1: Decimación por un factor  $D$*

Gráficamente lo que hace la decimación es tomar cada  $D$  muestras equidistantes de la señal original, y descarta las demás; dejando así una señal equivalente pero con diferente tasa de muestreo, como se puede ver en la Figura 2.2.

El proceso de downsampling se representa por lo general mediante una caja con una flecha apuntando hacia abajo, seguida del factor de downsampling como se puede observar en la Figura 2.3.



(a) Señal original



(b) Señal decimada por un factor  $D=4$

Figura 2.2: Decimación en tiempo



*Figura 2.3: Downsampling*

Primero consideraremos el proceso de downsampling y después de su descripción se explicará el porqué es necesario aplicar una etapa de filtrado antes de la de Downsampling.

## 2.3 DOWNSAMPLING

El proceso de downsampling reduce la frecuencia de muestreo de la entrada  $F_x$  por un factor entero  $D$ , el cual es conocido como factor de downsampling, como se puede observar en la Figura 2.1, este proceso se realiza después de la etapa de filtrado.

### 2.3.1 Análisis en el dominio del tiempo

Si denotamos a  $h(n)$  como la respuesta al impulso del filtro, entonces tenemos que la salida  $v(n)$  del filtro se puede obtener mediante la convolución de  $x(n)$  y  $h(n)$ , y está dada por [5]:

$$v(n) = \sum_{k=0}^{\infty} h(k)x(n-k) \quad (1)$$

La señal de salida del filtro entra al down-sampler y la frecuencia de muestreo será reducida por un factor  $D$ ; dando como resultado  $y(m)$ , con  $y(m)$  se puede observar que sucede un escalamiento en tiempo, entonces:

$$y(m) = v(mD) \quad (2)$$

Usando (1) y (2) podemos llegar a la conclusión de que [5]:

$$y(m) = \sum_{k=0}^{\infty} h(k)x(mD - k) = \sum_{k=0}^{\infty} x(k)h(mD - k) \quad (3)$$

### **2.3.2 Análisis en el dominio de la frecuencia**

Las características en el dominio de la frecuencia de la señal de salida  $y(m)$  pueden ser obtenidas relacionando el espectro de  $y(m)$  con el espectro de la señal de entrada  $x(n)$ . Para empezar definamos una secuencia  $v'(n)$  como [4]:

$$v'(n) = \begin{cases} v(n) & n = 0, \pm D, \pm 2D \dots \\ 0 & \text{cualquier otro valor} \end{cases} \quad (4)$$

Se puede observar que  $v'(n)$  puede ser vista como una secuencia obtenida de la multiplicación de  $v(n)$  con un tren de impulsos periódicos de periodo  $D$  y de amplitud unitaria, como se muestra en la Figura 2.4, quedando así [4]:

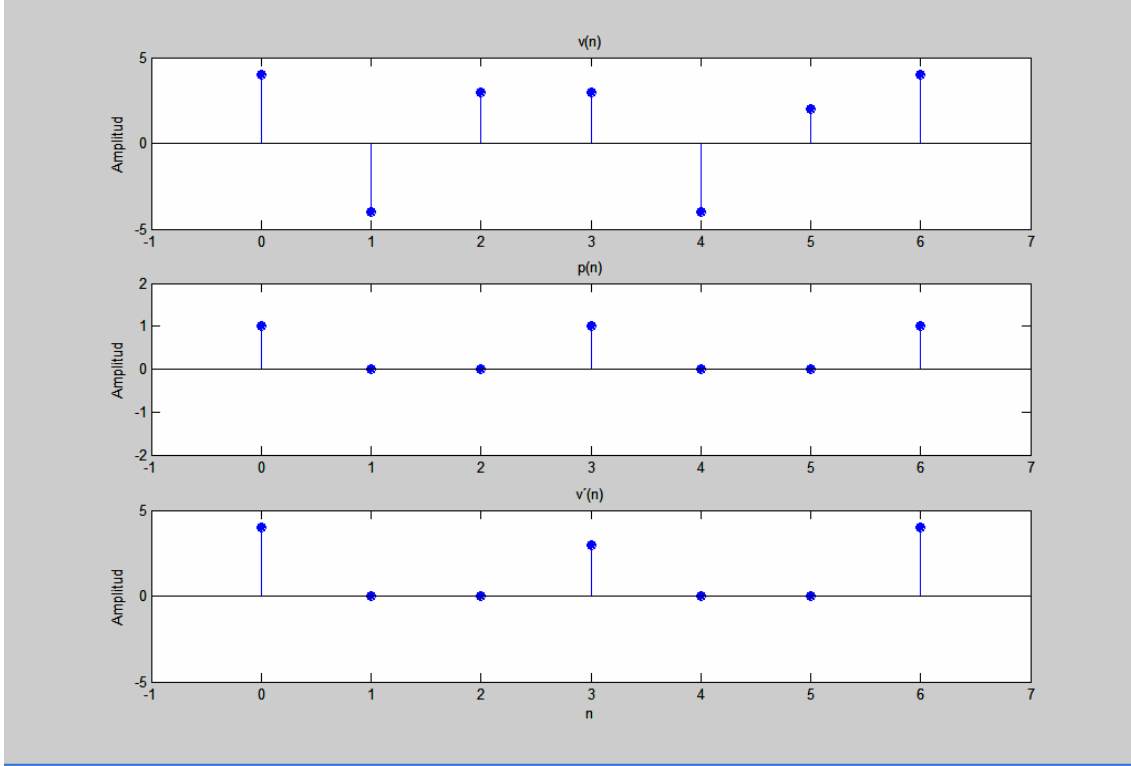
$$v'(n) = v(n)p(n) \quad (5)$$

La representación de la serie de Fourier discreta para un tren de impulsos  $p(n)$ , con periodo “ $D$ ” está dada por [4]:

$$p(n) = \frac{1}{D} \sum_{k=0}^{D-1} e^{j \frac{2\pi kn}{D}} \quad (6)$$

Obtenemos la Transformada de Fourier (FT) de la secuencia  $v'(n)$  usando (5):

$$V'(e^{j\omega}) = \sum_{n=-\infty}^{\infty} v'(n)e^{-j\omega n} = \sum_{n=-\infty}^{\infty} v(n)p(n)e^{-j\omega n} \quad (7)$$



*Figura 2.4: Multiplicación de  $v(n)$  con un tren de impulsos periódicos  $p(n)$  con período  $D=3$*

Sustituyendo la ecuación (6) en (7) se obtiene que:

$$V'(e^{jw}) = \sum_{n=-\infty}^{\infty} v(n) \left[ \frac{1}{D} \sum_{k=0}^{D-1} e^{\frac{j2\pi kn}{D}} \right] e^{-jwn} \quad (8)$$

Al intercambiar las sumatorias en (8), se obtiene la siguiente expresión para  $V'(e^{jw})$  [4]:

$$V'(e^{jw}) = \frac{1}{D} \sum_{k=0}^{D-1} \sum_{n=-\infty}^{\infty} v(n) e^{-jn(w - \frac{2\pi k}{D})} \quad (9)$$

Sabemos que la Transformada de Fourier para  $v(n)$  es:

$$V(e^{jw}) = \sum_{n=-\infty}^{\infty} v(n)e^{-jnw} \quad (10)$$

Aplicando la propiedad de cambio de frecuencia de la Transformada de Fourier, tenemos que [4]:

$$V(e^{j(w-\frac{2\pi k}{D})}) = \sum_{n=-\infty}^{\infty} v(n)e^{-jn(w-\frac{2\pi k}{D})} \quad (11)$$

Al observar el lado derecho de la ecuación (11) podemos darnos cuenta que es igual a una parte de (9), por lo que podemos concluir que:

$$V'(e^{jw}) = \frac{1}{D} \sum_{k=0}^{D-1} V(e^{j(w-\frac{2\pi k}{D})}) \quad (12)$$

Por otro lado, sabemos que:

$$Y(z) = \sum_{n=-\infty}^{\infty} y(n)z^{-n} = \sum_{n=-\infty}^{\infty} v(mD)z^{-mD} \quad (13)$$

Y que:

$$y(n) = v(mD) = v'(mD) \quad (14)$$

Haciendo uso de la definición de Transformada de Fourier para  $v'(z)$  y usando la ecuación (14) podemos reescribirla como [3]:

$$V'(z) = \sum_{m=-\infty}^{\infty} v(mD)z^{-mD} \quad (15)$$

Aplicando la propiedad de cambio de frecuencia de la Transformada de Fourier, tenemos que:

$$V'(e^{j\frac{w}{D}}) = \sum_{m=-\infty}^{\infty} v(mD)e^{-jwm} \quad (16)$$

Al sustituir  $y(m) = v(mD)$  en (16):

$$V'(e^{j\frac{w}{D}}) = \sum_{m=-\infty}^{\infty} y(m)e^{-jmw} = Y(e^{jw}) \quad (17)$$

Finalmente, usando a (12) podemos reescribir a (17) como:

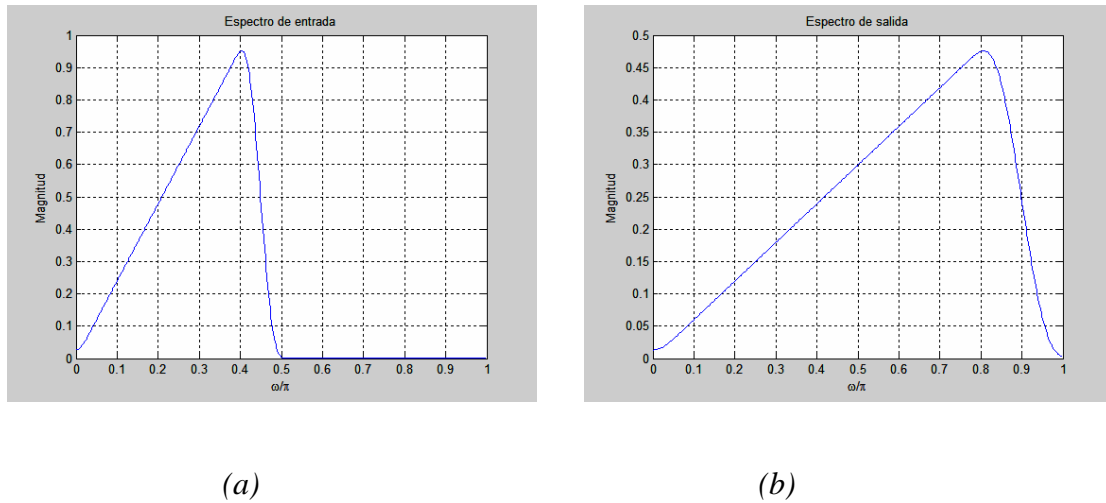
$$Y(e^{jMw}) = V'(e^{jw}) = \frac{1}{D} \sum_{k=0}^{D-1} V(e^{j(w - \frac{2\pi k}{D})}) = \frac{1}{D} \sum_{k=0}^{D-1} V(zW_D^k) \quad (18)$$

donde:

$$W_D^k = e^{-j\frac{2\pi k}{D}}, \quad k = 0, \dots, D-1 \quad (19)$$

Este resultado puede ser interpretado gráficamente como sigue: (a) alarga el espectro  $V(e^{jw})$  por un factor  $D$  para obtener  $V(e^{jw/D})$ , (b) crea  $D-1$  copias de esta versión alargada en cantidades sucesivas de  $2\pi$  y (c) agrega todas estas versiones alargadas al espectro sin alargar de  $V(e^{jw/D})$  y la amplitud se modifica en un factor de  $1/D$ . Los resultados anteriores se pueden entender mejor mediante la Figura 2.5 [3]:





*Figura 2.5: (a) Espectro de entrada antes del proceso de downsampling, (b) Espectro de salida después del proceso de downsampling con un factor  $D=2$*

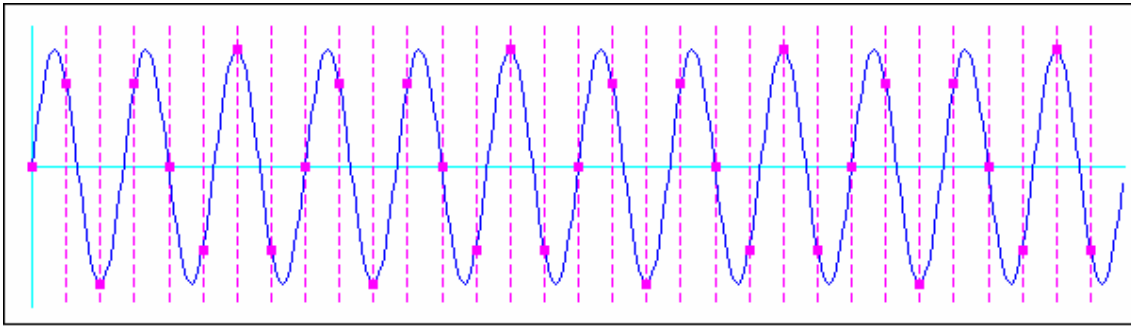
Como se puede observar en la Figura 2.5, el espectro de la señal después del proceso de downsampling es el espectro de entrada alargado por un factor de 2. La frecuencia está escalada por  $D=2$ , por ejemplo, los puntos 0.4 y 0.5 en la escala de frecuencia, se convierten en los puntos 0.8 y 1 de la señal después del downsampling. La magnitud de la señal después del proceso de Downsampling también se ve afectada, en este caso, por un factor de  $1/D = 1/2$  tal como se predijo anteriormente. Recordemos que en el dominio del tiempo sucede lo opuesto (la señal resultante fue comprimida por un factor  $1/D$ , ver Figura 2.2), ya que las representaciones en tiempo y frecuencia son mutuamente inversas.

## 2.4 ALIASING

De acuerdo al teorema del muestreo una forma de onda periódica puede ser correctamente reconstruida de los valores muestreados, en discretos e intervalos uniformes de tiempo, usando una frecuencia de muestreo  $F_N$  (o frecuencia de Nyquist) de al menos dos veces la frecuencia más alta de la forma de onda  $F_{MAX}$ , el enunciado anterior puede ser expresado de la siguiente manera:

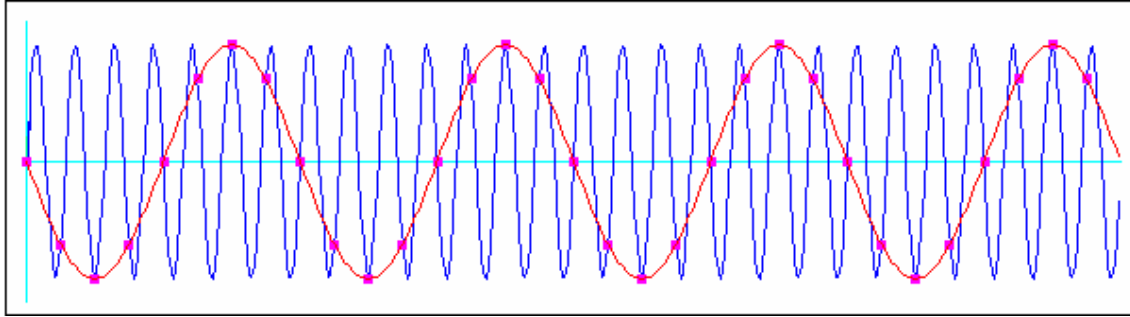
$$F_N = 2F_{MAX} \quad (20)$$

Mientras se siga la regla de muestrear con este límite, no habrá problema alguna en reconstruir la señal [4], en el dominio del tiempo podemos observar el siguiente ejemplo que se muestra en la Figura 2.6, en donde se muestra una señal sinusoidal con una frecuencia de 3Khz, una frecuencia de muestreo de 8000 muestras por segundo (una muestra cada 0.125 milisegundos) durante un tiempo de 0.004 segundos:



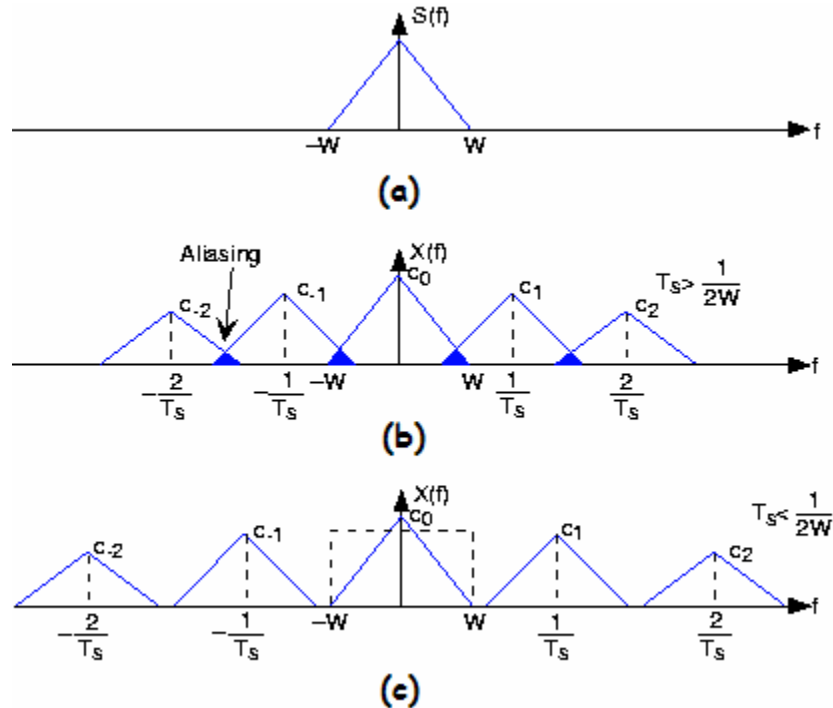
*Figura 2.6: Señal senoidal muestreada por arriba de la frecuencia de Nyquist*

Sin embargo cuando se muestra a una frecuencia por debajo de la frecuencia de Nyquist, más de una señal sinusoidal continua en el tiempo puede ser representada mediante los puntos muestreados, y existe una ambigüedad en cuanto a la reconstrucción de la señal que los valores muestreados representan, en la Figura 2.7 se muestra en azul la señal original con una frecuencia de 7Khz, y una frecuencia de muestreo de 8000 muestras por segundo (una muestra cada 0.125 milisegundos) y durante un tiempo de 0.004 segundos, claramente se puede observar que la frecuencia de muestreo es mucho menor que la frecuencia de la señal original y que mediante los puntos muestreados, se puede representar más de una señal sinusoidal.



*Figura 2.7: Señal sinusoidal muestreada por debajo de la frecuencia de Nyquist*

El fenómeno de *aliasing* también se presenta en el dominio de la frecuencia, y en esta caso, los espectros repetidos de la señal original se traslapan, por lo que la forma del espectro de las señales será indeseablemente alterado, en la Figura 2.8(a) se muestra el espectro de una señal limitada en Banda, la Figura 2.8(b) muestra el fenómeno de aliasing al escoger un intervalo de muestreo  $T_s$  relativamente grande con respecto al ancho de Banda, la Figura 2.8(c) muestra la evasión de aliasing al elegir el intervalo de muestro suficientemente pequeño (cumpliendo el criterio de Nyquist).



*Figura 2.8: (a) Espectro de una señal limitada a  $W$  Hz.. (b) Fenómeno de aliasing, (c) Evasión de aliasing.*

Como se puede observar en la ecuación (18) el espectro que se obtiene después del proceso de downsampling, son réplicas repetidas del espectro original. Sin embargo, debido al factor de downsampling  $D$ , la señal original debe ser limitada en una banda de  $\pi/D$ , ya que si la señal original no se limita dentro de esta banda, las réplicas se solaparán y se presentará el problema de *aliasing* en frecuencia. Para evitar el *aliasing*, es necesario limitar el espectro de la señal antes del proceso de downsampling por debajo de  $\pi/D$ . Es por este motivo que es necesario que un filtro digital pasabajas preceda al downsampler como lo muestra la Figura 2.1.

A este filtro se le llama *filtro antialiasing*. Las especificaciones exactas del filtro dependen de cuánto *aliasing*, si es que se puede, permitir. Por ejemplo en el banco QMF (Capítulo 5), un cierto grado de *aliasing* es usualmente permitido ya que posteriormente se puede cancelar. Sin embargo, algunas aplicaciones permiten un cierto grado de *aliasing* durante el proceso de downsampling. Las especificaciones para el filtro *antialiasing* pasabajas están dadas por [1]:

$$|H(e^{jw})| = \begin{cases} 1, & |w| \leq w_c / D, \\ 0, & \pi / D \leq |w| \leq \pi, \end{cases} \quad (21)$$

donde  $w_c$  es la frecuencia máxima que necesita ser conservada en la señal decimada.

Para poder entender mejor la implicación de la ecuación (18), consideremos un factor  $M=2$  con una entrada  $x(n)$  cuyo espectro se muestra en la Figura 2.9(a). Por conveniencia asumimos que  $X(e^{jw})$  es una función real con una respuesta en frecuencia asimétrica, de la ecuación (18) tenemos que:

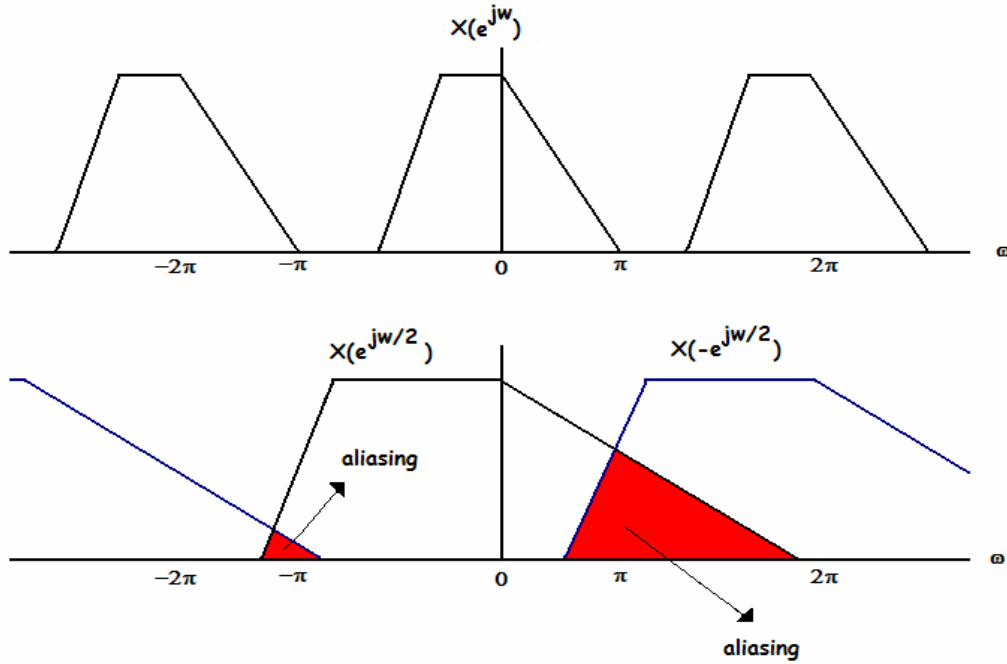
$$Y(e^{jw}) = \frac{1}{2} [X(e^{jw/2}) + X(-e^{jw/2})] \quad (22)$$

La gráfica de  $\frac{1}{2}X(e^{j\omega/2})$  se muestra en la línea continua en la Figura 2.9(b).

Para determinar la relación del segundo término en la ecuación (22) con respecto al primero, observamos que:

$$X(-e^{j\omega/2}) = X(e^{j(\omega-2\pi)/2}) \quad (23)$$

Como se puede observar, el segundo término de la ecuación (22) se obtiene al cambiar el primer término  $X(e^{j\omega/2})$  a la derecha por una cantidad de  $2\pi$ , como se muestra en las líneas en azul de la Figura 2.9(b). La gráfica de los dos términos de la ecuación (22) tienen traslape, y por lo tanto, la forma original de  $X(e^{j\omega})$  se pierde cuando  $x(n)$  pasa por el proceso de downsampling. Este traslape causa el *aliasing*. Por lo que no hay traslape sólo si  $X(e^{j\omega})$  es cero para  $|\omega| \geq \pi/2$ .

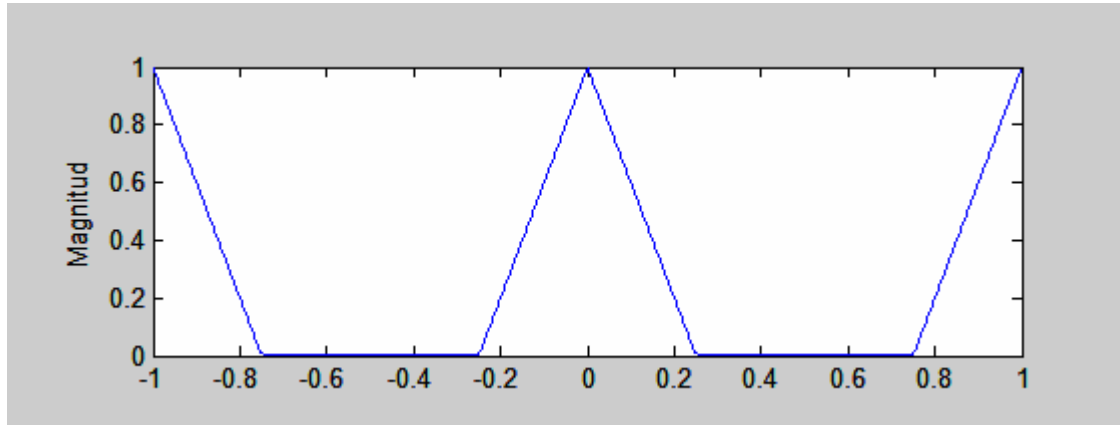


*Figura 2.9: Ilustración de Aliasing en el dominio de la frecuencia*

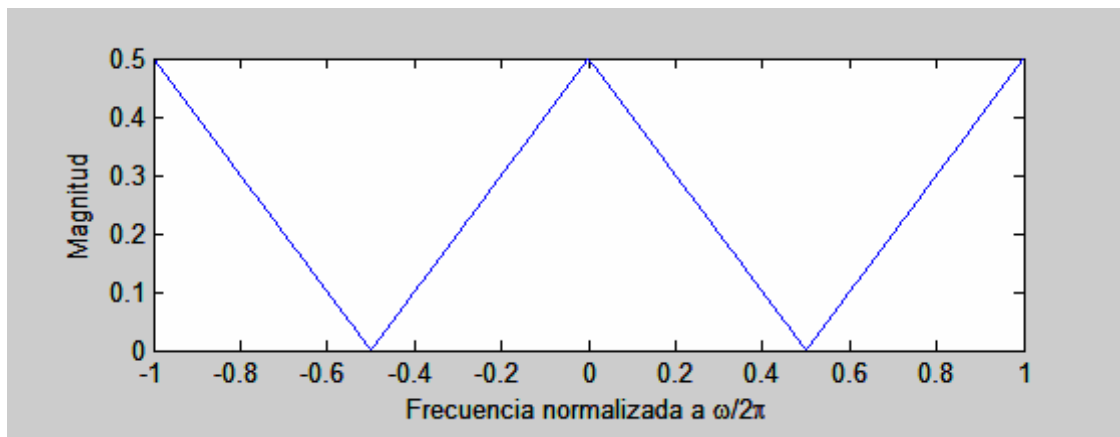
Los efectos de la decimación en el dominio de la frecuencia se muestran en la siguiente figura, en donde se utiliza la función  $\text{sinc}(x)$  la cual está definida como:

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x} \quad (24)$$

y su espectro es de forma triangular, en la Figura 2.10(a) se muestra el proceso de decimación sin un filtro *antialiasing*, y en la Figura 2.10(b) se puede observar que para un factor de decimación  $D=2$  no hay ningún problema y que en este caso no se presenta aliasing:



(a)



(b)

*Figura 2.10: Proceso de decimación con un factor  $D=2$  sin filtro antialiasing*

Como se pudo observar en la figura anterior, en este caso no fue necesario aplicar un filtro *antialiasing* con un factor  $D=2$ , pero si el factor  $D$  fuera mayor, los problemas de aliasing se presentarían.

En la Figura 2.11 (a) se muestra la misma entrada sin aplicar un filtro pasabajas, por lo que el espectro no está limitado, en la Figura 2.11 (b) se muestra la salida después del proceso de decimación con un factor  $D=3$ , y el espectro se traslapa, por lo que si se quisiera reconstruir, es imposible obtener la señal original, en la Figura 2.11 (c) se aplica un filtro a la señal de entrada y se obtiene su espectro, en la Figura 2.11 (d) se muestra la salida de la señal, la salida que se obtiene no es perfecta debido a que se usó un filtro no ideal, ya que en la práctica no se pueden implementar filtros ideales por ser sistemas no causales, y por lo tanto físicamente no realizables para aplicaciones de señales en tiempo real, recordemos que un sistema no causal es aquel que su salida depende no sólo de entradas de valor presente y valores anteriores; sino también de valores futuros.

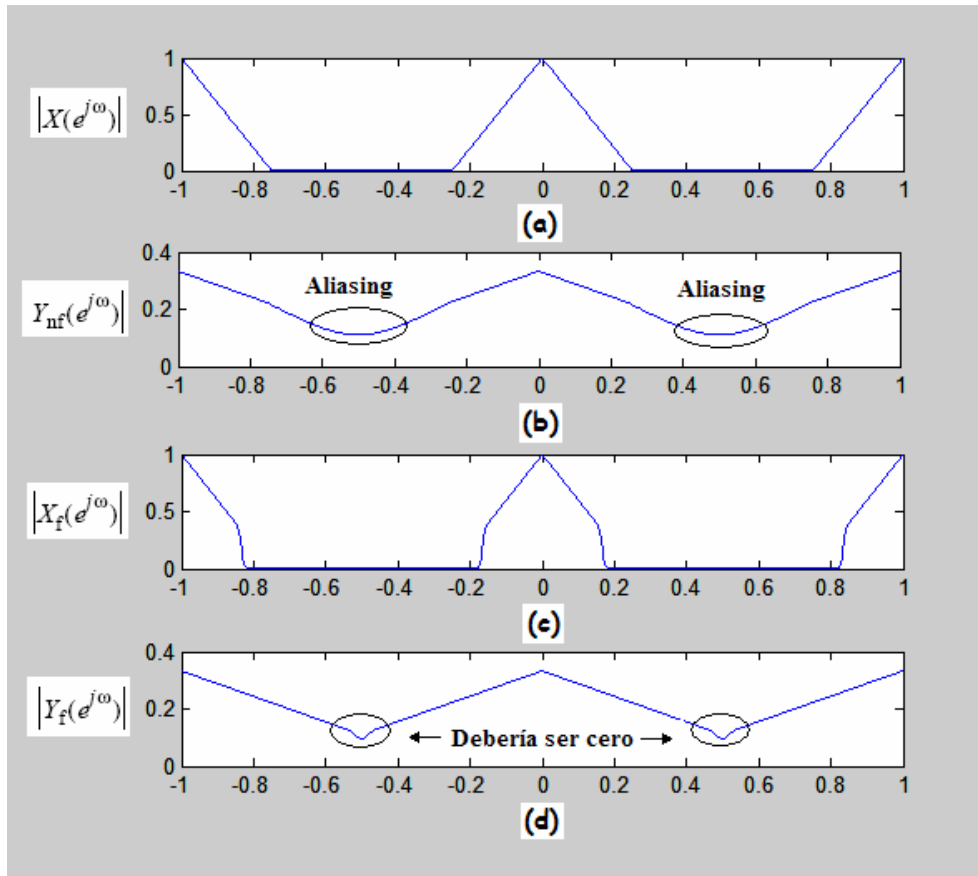


Figura 2.11: (a)  $E$ spectro de la señal de entrada sin etapa de filtrado, (b)  $E$ spectro de la señal de salida después de decimar con factor  $D=3$ , (c)  $E$ spectro de la señal de entrada con etapa de filtrado, (d)  $E$ spectro de la señal de salida con etapa de filtrado no ideal.

Finalizamos este capítulo señalando que el downsampler es un sistema lineal, a continuación examinamos si es variante en el tiempo. En general, si un retraso de la señal de entrada ocasiona el mismo retraso en la señal de salida, el sistema es invariante en el tiempo, tomando en cuenta la ecuación (2), su salida  $y_1(m)$  para una entrada  $v_1(m) = v(m-m_0)$  es entonces dada por [4]:

$$y_1(m) = v_1(mD) = v(mD - m_0) \quad (25)$$

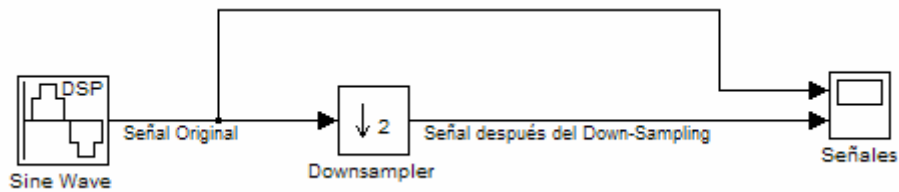
Recordando a la ecuación (2):

$$y(m - m_0) = v[(m - m_0)D] = v[mD - m_0D] \neq y_1(m) \quad (26)$$

El downsampler es un sistema lineal y variante en el tiempo, consecuentemente, un retraso en la señal de entrada no da como resultado el mismo retraso de la señal después del proceso de downsampling.

## 2.5 SIMULACIÓN DEL DOWNSAMPLER

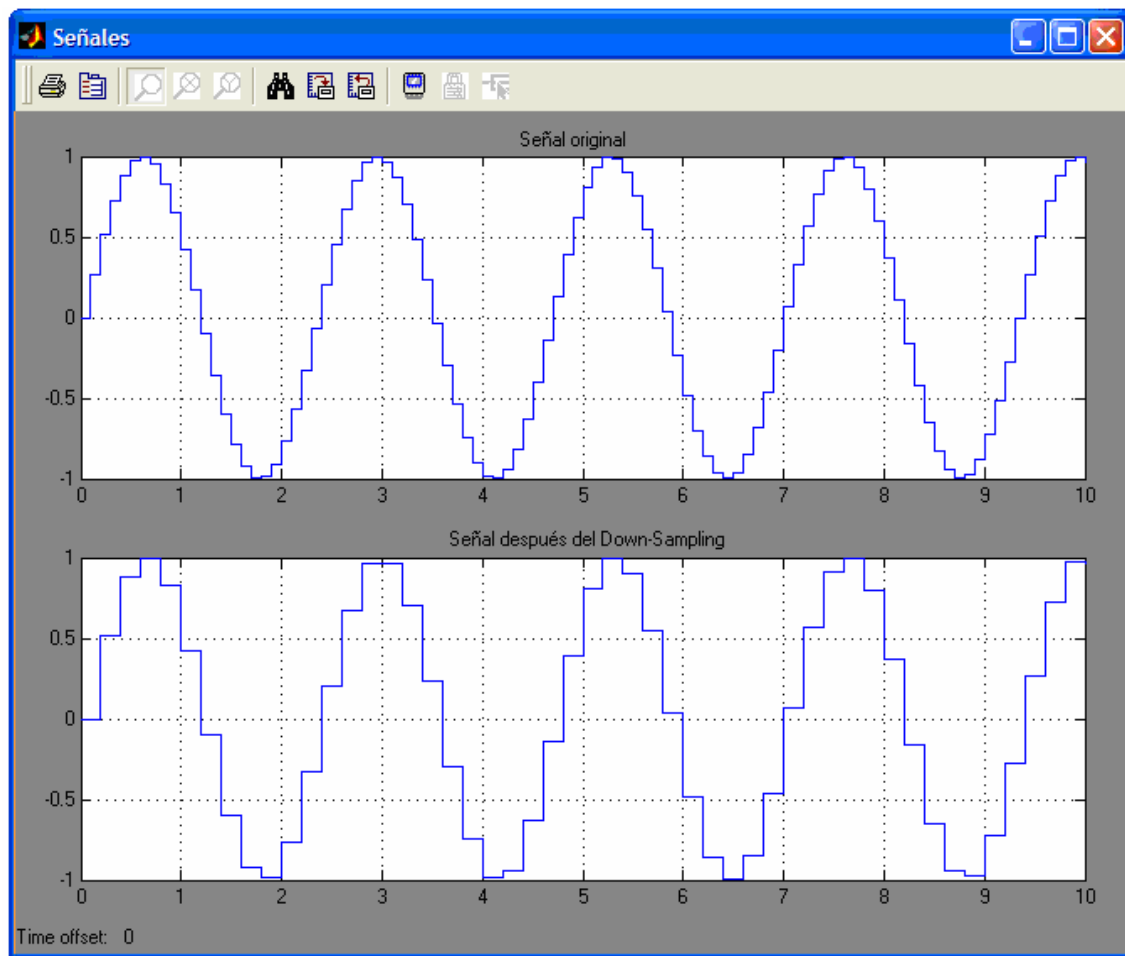
A continuación se muestra una simulación del *downsampler* en Simulink



*Figura 2.12: Simulación del downsampler*

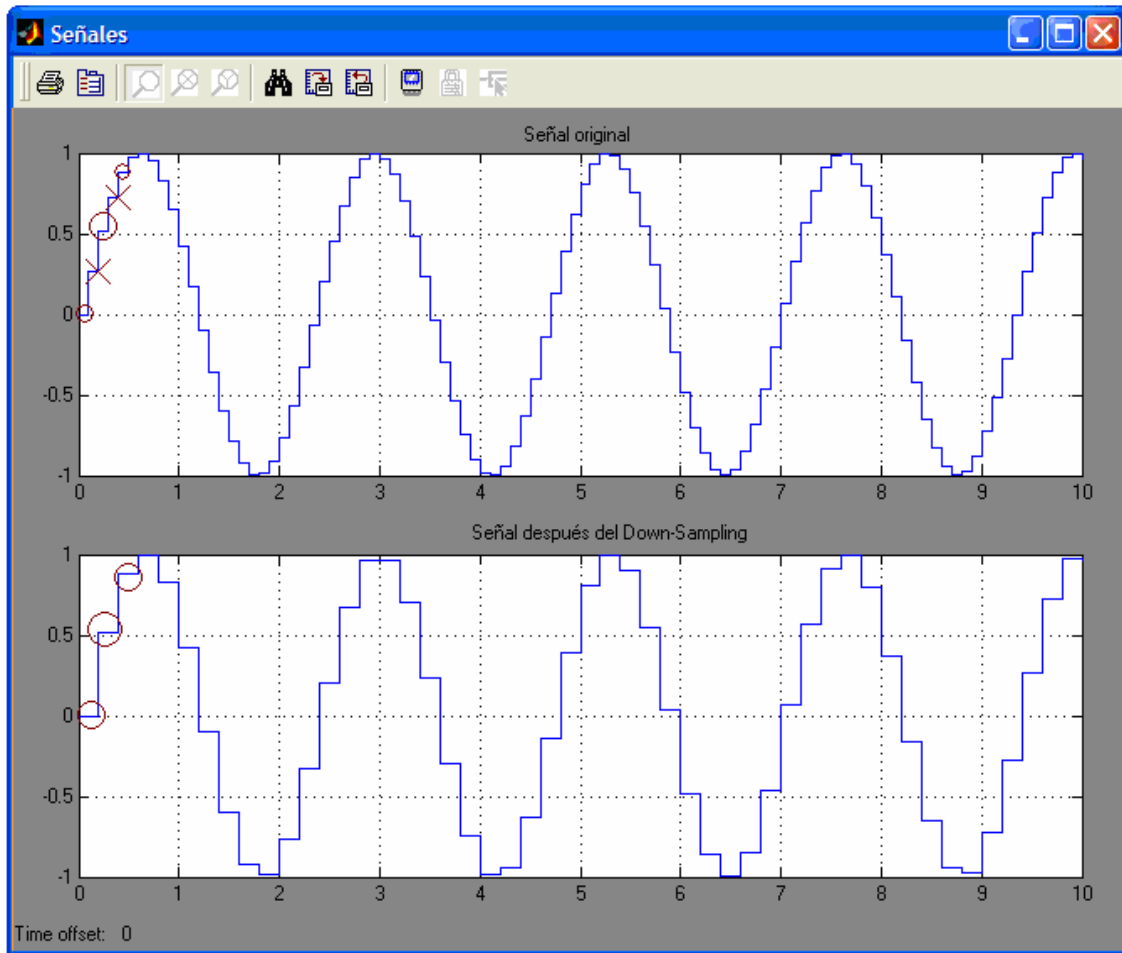
La señal sinusoidal tiene una frecuencia de 0.43Hz y un tiempo de muestreo de 0.1 segundos. La salida de las dos señales se muestran en la Figura 2.13 se muestra las salidas de este sistema:





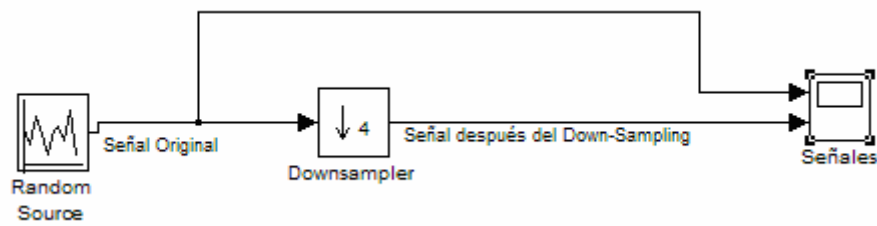
*Figura 2.13: Salidas del sistema simulado*

Se puede observar en la Figura 2.13 que la señal después del downsampling tiene la forma de la señal original pero con una frecuencia menor, ya que tiene  $M-1$  muestras de la señal original, como se muestra en la Figura 2.14, en donde se encierran en círculo rojo las muestras que han sido tomadas de la señal original para construir la señal después del downsampling, y se marca con una equis las muestras que han sido descartadas.

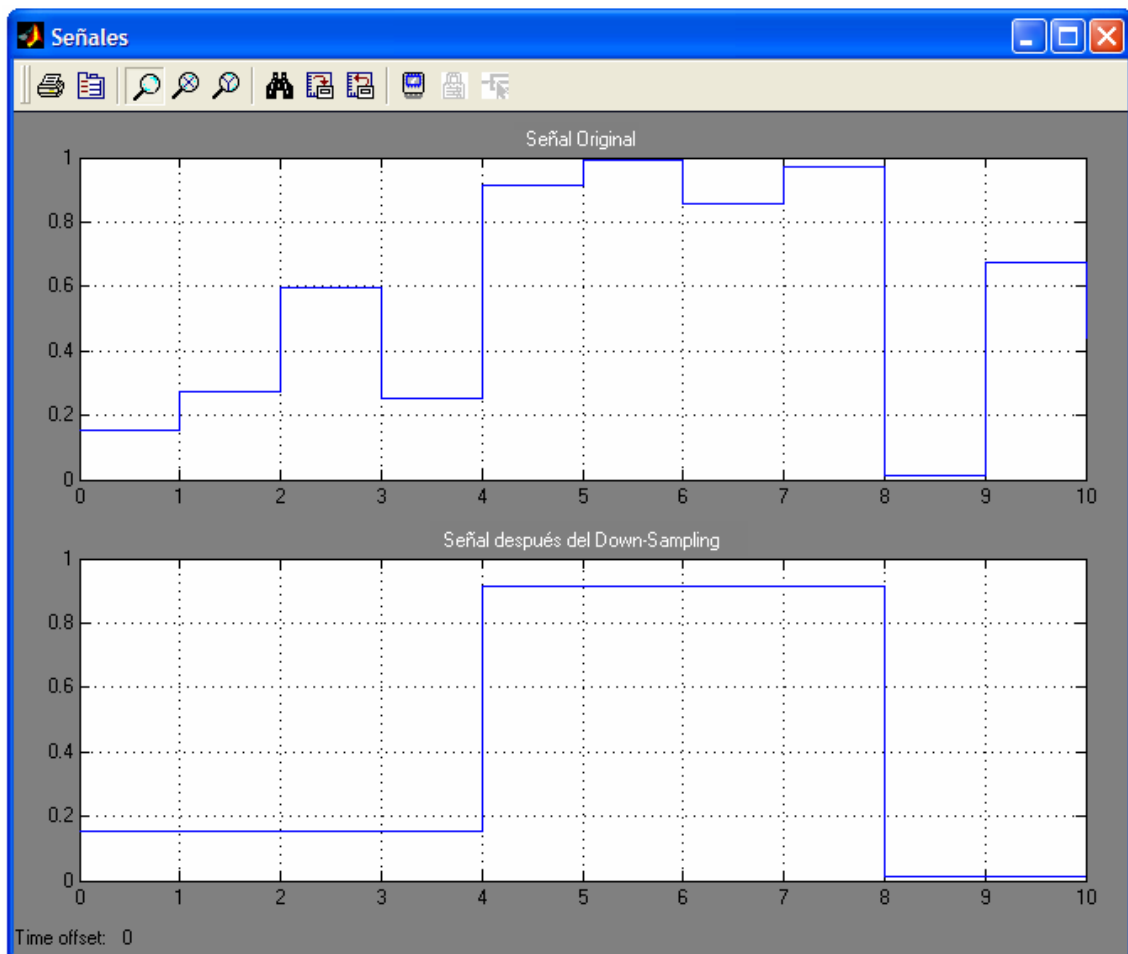


*Figura 2.14: Señalización de las muestras descartadas mediante el proceso de downsampling*

Dado que como señal de entrada no siempre se tiene una señal senoidal, en la Figura 2.15 se muestra la simulación del downsampler teniendo como entrada una señal aleatoria, la cual, fue generada mediante el bloque de “Random Source” de la librería *DSP Blockset*. La Figura 2.16 muestra la salida correspondiente a la señal decimada



*Figura 2.15: Simulación del downsampler con entrada aleatoria*



*Figura 2.16: Salida del sistema para una señal aleatoria decimada por un factor  $D=4$*

Claramente se puede observar que el sistema toma una de cada cuatro muestras, produciendo así la señal de salida correspondiente.

## 2.6 SIMULACIÓN DEL CONVERTIDOR A/D SOBREMUESTREADO

### SIGMA-DELTA

La idea básica en un convertidor sobremuestreado A/D es la de incrementar la frecuencia de muestreo de la señal a tal grado que con un cuantizador de baja resolución sea suficiente. Con el sobremuestreo podemos reducir el rango dinámico de los valores de la señal entre muestras sucesivas y por lo tanto reducir los requerimientos de resolución del cuantizador [5].

El convertidor A/D sigma-delta, realiza una cuantización de baja resolución, pero opera a una frecuencia de muestreo del orden de 10 veces o más la frecuencia de Nyquist. Una de las ventajas en este convertidor es que, el ruido de cuantificación se puede mejorar al aumentar la frecuencia de muestreo, el ruido de cuantificación se distribuye uniformemente desde la frecuencia 0 hasta la frecuencia de muestreo. Si se aumenta la frecuencia de muestreo, la potencia del ruido disminuye proporcionalmente.

Dos técnicas importantes, llamadas *sobremuestreo* y *quantization noise shaping*, son usadas en el convertidor sigma-delta para compensar la resolución de cuantización con frecuencia de muestreo [5].

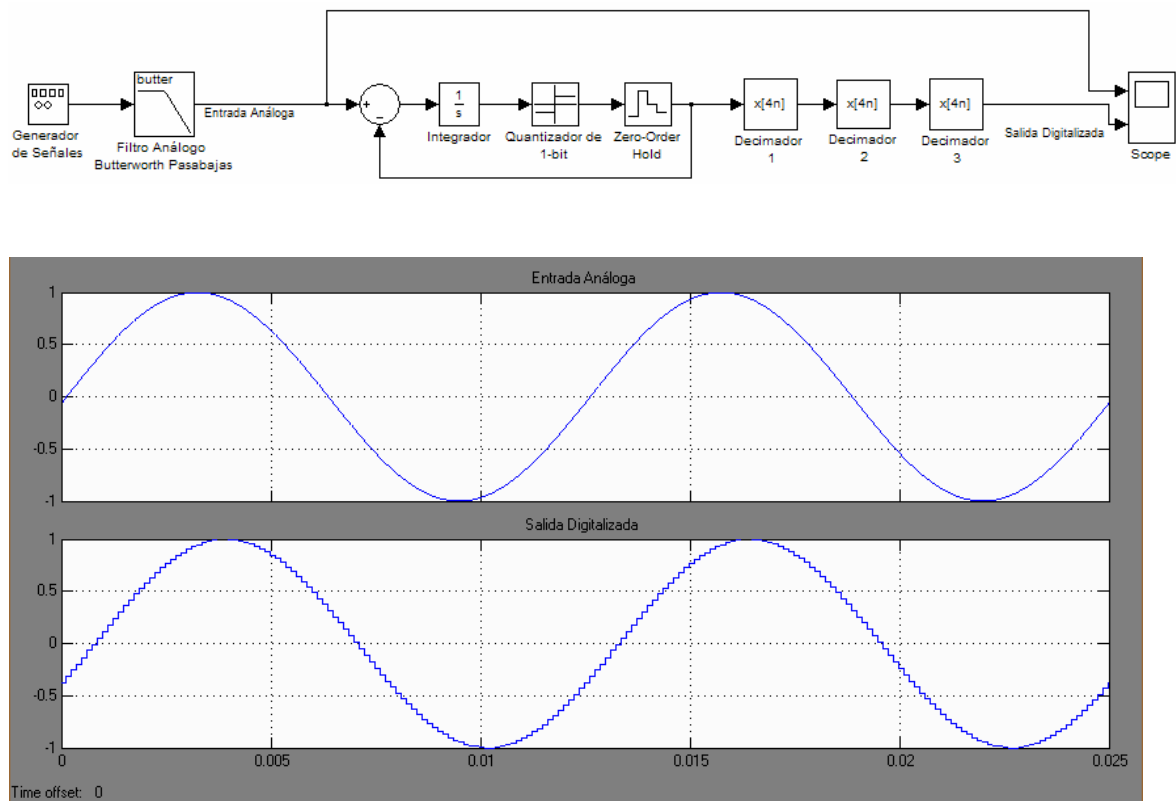
Como su nombre lo dice, en este tipo de convertidor la señal análoga es muestreada a una frecuencia mucho mayor que la frecuencia de Nyquist, dando como resultado que las amplitudes de las dos muestras consecutivas sean muy pequeñas, permitiéndoles ser representadas en forma digital, usando muy pocos bits, usualmente uno. La relación que existe entre la frecuencia de sobremuestreo y la frecuencia de Nyquist es [1]:

$$M = \frac{f_B}{F_{Nyquist}} \quad (27)$$

Donde  $M$  es el factor de sobremuestreo y  $f_B$  es el ancho de banda de la señal análoga.

La segunda técnica que utiliza el convertidor sigma-delta es *quantization noise shaping*, la idea es la de reformar el ruido mandándolo a frecuencias más altas, o sea, mandándolo a una banda de frecuencia que no sea relevante, esto se lleva a cabo mediante el lazo de retroalimentación, que actúa como un filtro pasa-bajas para la señal de entrada y como un filtro pasa-altas para la cuantización del ruido [9].

El diagrama del convertidor sigma-delta usado en la simulación se muestra a continuación. El término “sigma- delta” viene del hecho de que hay un punto de suma (sigma) y un modulador delta (integrador y cuantizador de 1 bit) [1].



*Figura 2.17: Diagrama del convertidos Sigma/Delta*

PARÁMETROS DE SIMULACIÓN

	FORMA DE ONDA	AMPLITUD	FRECUENCIA	UNIDADES
Generador de Señales	Seno	1	80	Hertz

	Diseño	Tipo	Orden	Frecuencia Paso en rads/seg.
Filtro Analógico	Butterworth	Pasabajas	5	$2\pi \cdot 4000$

	Tiempo de Muestreo
Zero Order Hold	$1/(64 \cdot 2 \cdot 4000)$

	Coeficientes Filtro FIR	Factor de Decimación	Framing
Decimadores	<code>fir1(31,0.15)</code>	4	Maintain input frame size

	Parámetros de Simulación
Tipo	Variable-step Ode45

*Tabla 1: Parámetros de Simulación del convertidor Sigma/Delta*

Como se puede observar en la simulación, se tiene una señal análoga, en este caso una señal sinusoidal, la cual se hace pasar por un filtro pasabajas, esto con el fin de establecer un ancho de banda, forzar el criterio de Nyquist y evitar *aliasing*, a la salida del filtro se tiene una aproximación de la señal de entrada, esta señal entra al modulador delta, la cual después de pasar por el *Zero-Order Hold* se resta con la señal original.

La señal diferenciada, la cual es la parte “delta”, es acumulada en el integrador, el cual es la parte “sigma” y provee un promedio local de la entrada [9].

En muchas aplicaciones como ésta, es necesario hacer una decimación por un factor entero grande. Aunque esto se puede lograr al diseñar un filtro  $H(z)$ , es más eficiente (en términos de número de multiplicaciones por unidad de tiempo) diseñar un filtro de decimación en múltiples estados. Para este ejemplo, un factor de decimación de 64, se puede lograr mediante 3 etapas de decimación con un factor de 4 cada una

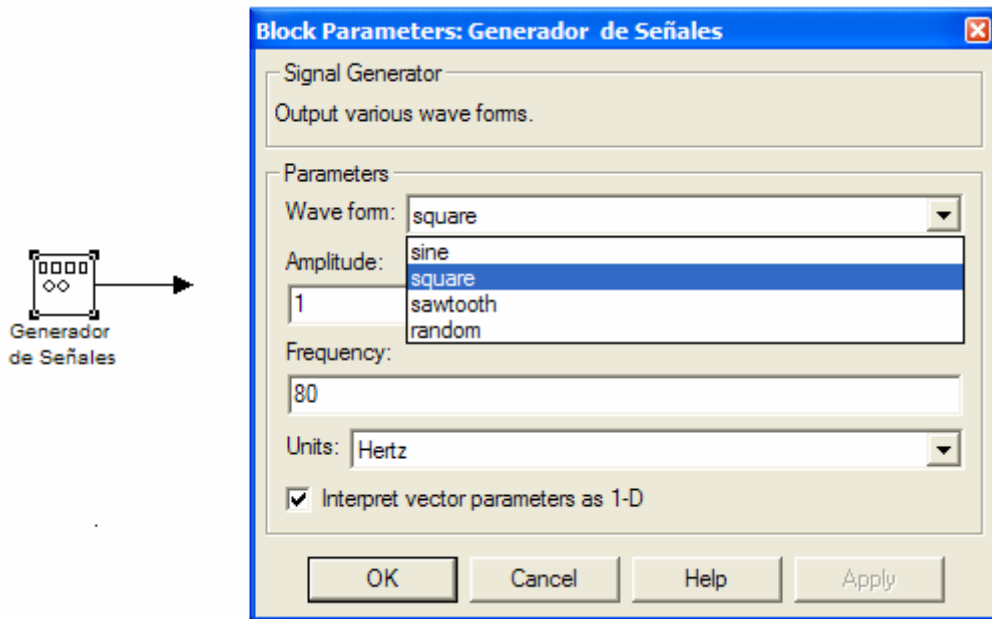
La decimación multi-estado ( $4*4*4$ ) reduce la frecuencia de muestreo a la original. Durante este proceso, remueve la cuantización del ruido que fue introducido por el lazo de retroalimentación y remueve las componentes no deseadas que no fueron removidas por el simple prefiltro análogo.

El bloque de decimación se puede encontrar en Simulink en la librería de *Multirate Filters* dentro de *DSP Blockset*.

El bloque de decimación tiene integrado el filtro anti-aliasing y el down-sampler, matlab cuenta con la función *fir1* la cual diseña un filtro pasabajas, los parámetros que se le tiene que especificar son el orden representado con la letra  $n$  y la frecuencia de corte representada por la letra  $Wn$ , ambos parámetros deben ser especificados dentro del bloque de decimación.

Una vez especificado las características del filtro, se debe indicar el factor de *downsampling* y el bloque de decimación habrá quedado caracterizado como fue el caso en el ejemplo anterior.

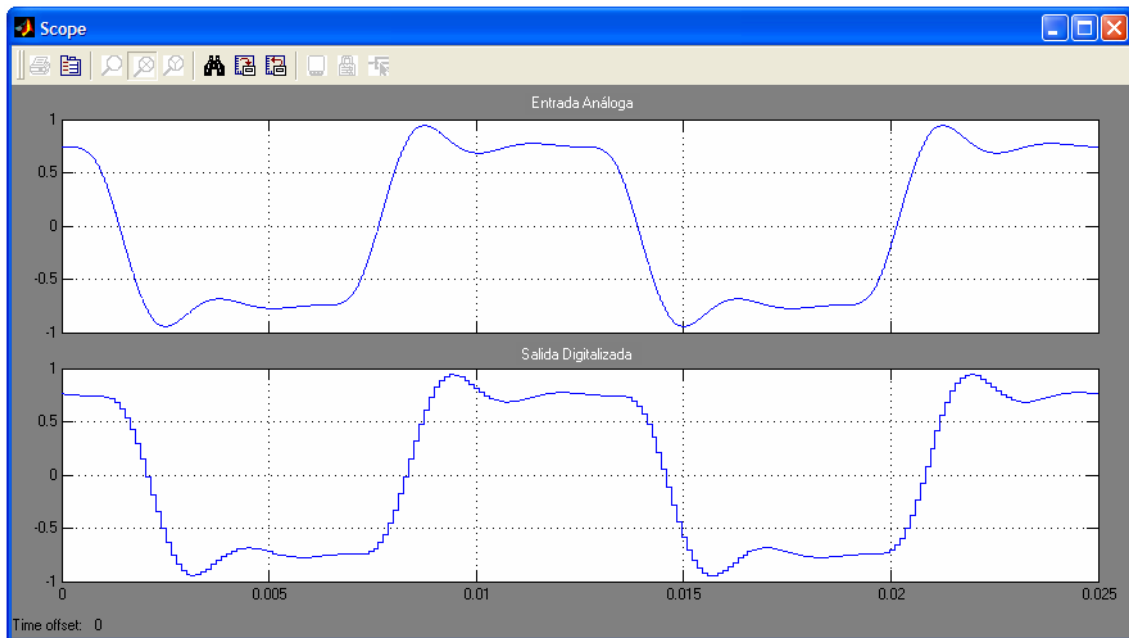
El Generador de Señales puede ser reconfigurado para obtener diversas entradas al convertidor, esto mediante la simple elección de la señal que se desee como lo muestra la Figura 2.18



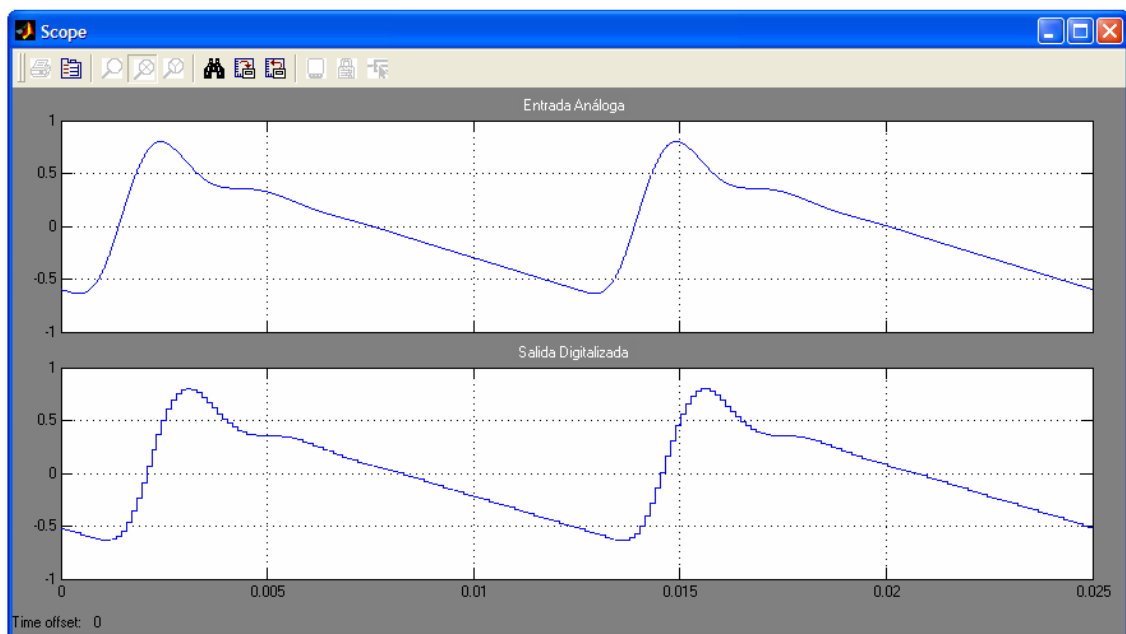
*Figura 2.18: Configuración del Generador de Señales*

Dentro del bloque del Generador de Señales, se caracteriza por completo la señal de entrada, al indicar su frecuencia y amplitud para las diversas señales disponibles. A continuación, en la Figura 2.19 se muestran diversas señales de entrada aplicadas al convertidor sigma-delta así como sus respectivas salidas. Todas las señales de entrada tienen una frecuencia de 80 Hertz y una amplitud unitaria.

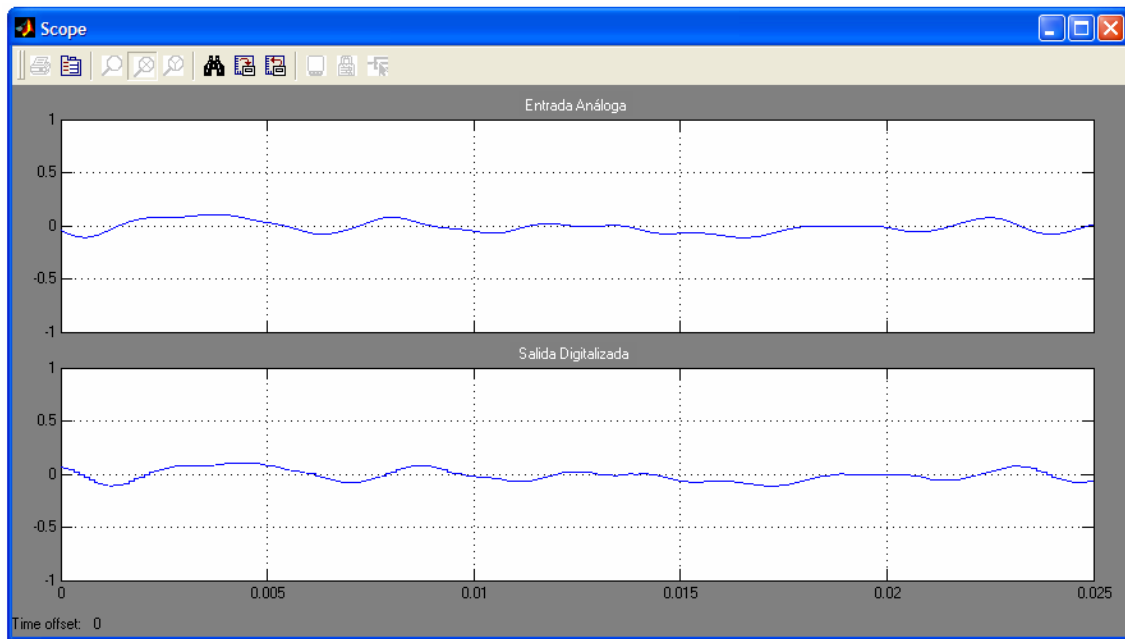




*(a) Entrada y Salida para una señal cuadrada*



*(b) Entrada y Salida para una señal diente de sierra*



*(c) Entrada y Salida para una señal aleatoria*

*Figura 2.19: Conversión de diferentes señales de entrada al convertidor  
sigma-delta*