

Procesamiento Digital de Señales

Práctica 1. Simulación de sistemas con Simulink. Cuantización de señales.

1. Objetivos:

Aprender a construir sistemas basados en diagramas de bloques con Simulink y simular esquemas de muestreo y cuantización de señales.

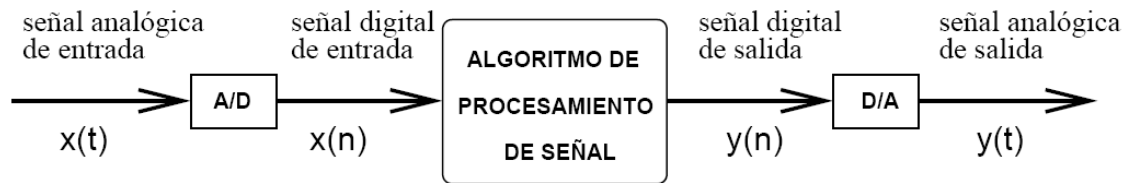
2. Introducción a Simulink

Simulink es una utilidad gráfica para la simulación de sistemas, tanto analógicos como discretos. En realidad, se trata de un recubrimiento de MatLab, un lenguaje interpretado que está especialmente adaptado al cálculo vectorial, lo que hace que sea adecuado para aplicaciones de procesamiento de señales (mono y bidimensionales), comunicaciones, control, etc. Para una primera sesión con Simulink, es conveniente seguir los siguientes puntos:

1. Arrancar MATLAB y fijar el directorio de trabajo.
2. Desde la ventana de comandos de MATLAB ejecutar la orden "simulink".
3. Aparecerá entonces una nueva ventana con la biblioteca de módulos de SIMULINK y otras librerías. Los módulos están agrupados por su función general.
4. Lo primero que debe hacerse es crear un nuevo modelo (File → New, o icono correspondiente) donde construir el esquema deseado.
5. Una vez abiertos los grupos de bloques necesarios, el sistema deseado se va generando mediante selección y desplazamiento de los bloques a la ventana de trabajo.
6. Para unir los distintos módulos del sistema se trazan líneas conductoras pulsando con el ratón en la salida de uno de los bloques y arrastrando hasta la entrada del siguiente (o viceversa).
7. Es posible cortar y copiar bloques como si de texto se tratase.
8. Los bloques y líneas se pueden desplazar seleccionándolos con el botón izquierdo y arrastrando hasta la nueva posición.
9. Cada bloque tiene una serie de parámetros que controla su funcionamiento. Para verlos o modificarlos debe pulsarse dos veces sobre el bloque.
10. El simulador propiamente dicho también tiene una serie de parámetros de control que deben ajustarse adecuadamente, de acuerdo con las necesidades de cada caso. Esto se consigue seleccionando Simulation en el menú de la ventana de trabajo. Dentro de este menú se encuentran también las opciones para arrancar y detener la simulación.

3. Procesamiento digital de señales. Introducción

Un sistema de Procesamiento Digital de Señales (PDS) es aquel que realiza algún tipo de transformación (algoritmo) sobre las muestras de una señal analógica de entrada digitalizada. En muchos casos, el resultado de esa transformación es otra señal digital que puede pasarse nuevamente, si así es requerido, a formato analógico, aunque, en general, la salida del sistema puede ser cualquier tipo de información obtenida de la señal de entrada. Un esquema de un sistema de procesamiento digital de señal sería el siguiente:

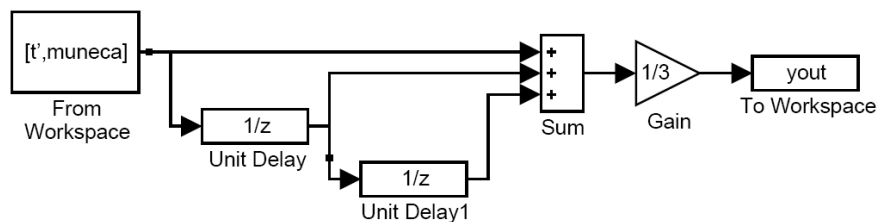


Mientras que una señal analógica se representa como función $x(t)$ de la variable tiempo t , su versión digitalizada se escribe como $x(n)$, donde n es un índice entero que representa el instante temporal nT si la señal se ha digitalizado con una frecuencia de muestreo $F_s = 1/T$. La señal $x(t)$ estará correctamente muestreada si cumple el teorema de muestreo, según el cual la señal no deberá contener componentes en frecuencia más allá de $F_s/2$.

Un ejemplo típico de sistema de procesamiento de señal es el filtro FIR. Un filtro FIR es un sistema discreto que proporciona una salida $y(n)$ que es combinación lineal de muestras ya recibidas de la señal entrada $x(n)$. Un ejemplo muy sencillo de filtro FIR es un filtro de media que calcule el promedio de las 3 últimas muestras recibidas

$$y(n) = \frac{1}{3} \sum_{k=0}^2 x(n-k)$$

Este es un filtro pasabaja que puede implementarse en Simulink de la siguiente manera:



Este sistema se construye de la siguiente forma:

- El bloque From Workspace se toma de la librería de Sources de Simulink. Su función es llevar una señal desde el espacio de trabajo de MatLab a Simulink. Previamente se debe haber leído una señal. Por ejemplo, situándose en el directorio de trabajo ejecutar:

```
load muneca.asc
plot(muneca)
```

La señal quedará almacenada en la variable muneca (muestreada con una frecuencia de 8 KHz). El bloque tiene como argumentos un vector t con los instantes de tiempo correspondientes a cada muestra, y la propia señal muneca. Ambos argumentos deben pasarse como vectores columna (la trasposición en MatLab se obtiene añadiendo una prima; p.ej. t'). Para realizar todo esto más la audición de la señal deben ejecutarse los siguientes comandos:

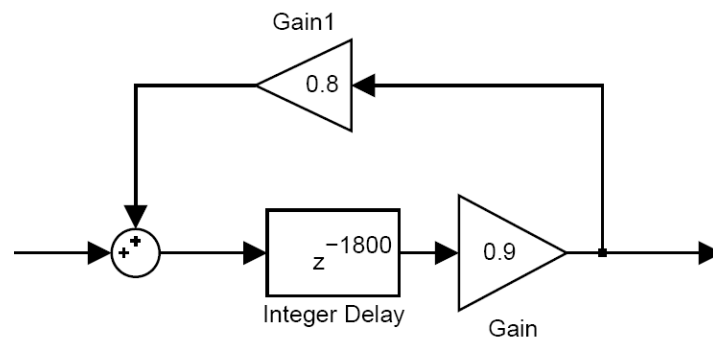
```
Fs=8000
N=length(muneca)
t=0:1/Fs:(N-1)/Fs;
plot(t,muneca)
```

```
sound(muneca/max(muneca))
```

- Los bloques de Unit Delay y suma se obtienen de las librerías de Discrete y Linear, respectivamente.
- El bloque To Workspace se obtiene de la librería Sinks de Simulink, y se encarga de devolver la señal procesada al espacio de trabajo de MatLab como un vector denominado `yout` por defecto (puede modificarse). Hay que especificar también el número de puntos de salida, que este caso será N .
- Antes de ejecutar la simulación deben especificarse los parámetros de la misma. Así, la simulación comenzará en 0 y terminará en $(N-1)/F_s$ y los parámetros de Step y Tolerancia se fijarán a $1/F_s$.

Adicionalmente, se comprobará el efecto auditivo que el aumento progresivo de la longitud de la media sobre la señal de salida. Ésto puede realizarse introduciendo más elementos de retardo o haciendo uso del bloque Discrete \rightarrow Discrete Filter.

Finalmente, se desarrollará un nuevo sistema simulink que realice algún efecto de libre elección sobre una señal de voz. Entre estos efectos podría considerarse un sistema de reverberación de voz como el de la figura.

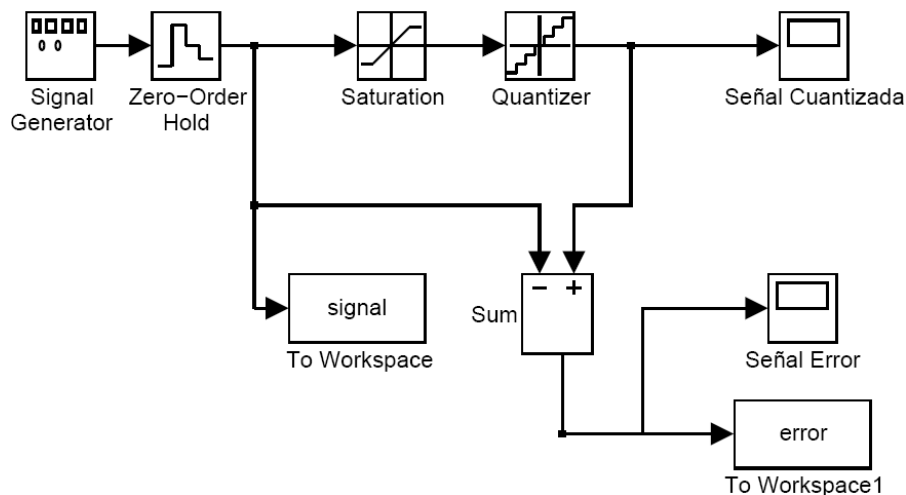


4. Representación digital de señales

La transmisión (o almacenamiento) digital de señales analógicas implica un cambio en la representación de las mismas a un formato digital. El primer paso para representar digitalmente una señal analógica consiste en discretizar el eje temporal, es decir, obtener muestras de la señal en una serie de instantes de muestreo. Adicionalmente, el valor de cada muestra debe representarse en formato binario. En esta parte de la práctica se utilizarán dos métodos para ello: cuantización (uniforme y ley μ), que es la forma más usual de representación, y modulación delta.

4.1. Cuantización uniforme

En primer lugar se realizará un cuantizador uniforme de 4 bits para una señal sinusoidal de amplitud 1 y frecuencia 1 Hz. El esquema es el siguiente:



Descripción:

- En *Signal Generator* seleccionar una señal seno $x(n)$ con la amplitud y frecuencia requeridas.
- El bloque *Zero-Order Hold* juega el papel de muestreador temporal. El periodo de muestreo se fijará a 0.01 ($F_s = 100$ Hz).
- El bloque de saturación fija el rango en el que la señal puede tomar valores. En este caso el rango es de ± 1 . Dado que la señal no superan el rango permitido, la aparición de este bloque es puramente formal.
- En el bloque cuantizador se especificará un cuanto de $\Delta = 2/16$, que indica que se trata de un cuantizador de 4 bits en el rango ± 1 .

Se estudiará la forma de onda de la señal error y su relación con el número de bits de cuantizador para 4bits, 3 bits ($\Delta = 2/8$) y 2 bits ($\Delta = 2/4$).

La señal de error $e(n)$ corresponde a un ruido de cuantización ya que la señal cuantizada $y(n)$ se obtiene como $y(n) = x(n) + e(n)$. Una medida de calidad de una señal degradada con ruido es la Relación Señal-Ruido (SNR), cuya definición general y específica para un cuantizador uniforme son las siguientes,

$$SNR = 10 \log_{10} \frac{\text{Energía de la señal}}{\text{Energía del ruido}} = 10 \log_{10} \left(\frac{E_x}{\Delta^2 / 12} \right)$$

En el caso de una señal sinusoidal la potencia E_x es $A^2/2$ (A es la amplitud del seno). La unidad de SNR es el decibelio (dB).

Se comprobará la coincidencia entre la SNR calculada mediante la fórmula anterior y la SNR medida a partir de la simulación de Simulink para cuantizadores de 4, 3 y 2 bits. Para obtener la SNR experimental se aplicará en la ventana de comandos de MatLab la expresión

```
10*log10(sum(signal.*signal)/sum(error.*error))
```

A continuación se observará el efecto de la cuantización uniforme sobre una señal de voz (tomada del mismo fichero muneca.asc de la práctica anterior). Para ello se construirá el esquema simulink de la siguiente figura 1.

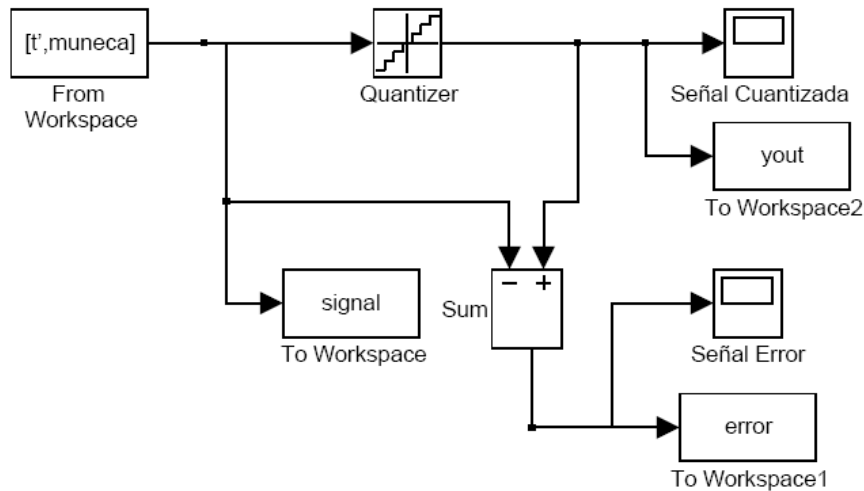


Figura 1. Cuantización uniforme de una señal de voz.

La señal original está muestreada con 12 bits (rango ± 2048). La señal será recuantizada a 8, 6 y 4 bits de la siguiente manera:

- 8 bits: el cuanto se fijará a $2^{12-8} = 16$.
- 6 bits: el cuanto se fijará a $2^{12-6} = 64$.
- 4 bits: el cuanto se fijará a $2^{12-4} = 256$.

Medir la SNR para cada uno de los casos anteriores y reproducir la señal cuantizada en la tarjeta de audio. Comentar los resultados.

4.2. Cuantización mediante ley μ

El cuantizador basado en la ley μ se basa en la realización de una operación no lineal sobre las muestras de entrada tal que el valor del cuanto será más pequeño para valores de muestra próximos a cero y mayor para amplitudes grandes. Para una señal como la de voz, para la que la mayoría de las muestras toman valores pequeños, la ventaja es evidente: la mayor parte de las muestras (las de amplitudes bajas) están cuantizadas con un cuanto más pequeño. Es decir, se realiza un mayor esfuerzo de cuantización allá donde es más necesario. La operación no lineal es la siguiente:

$$y(n) = F[x(n)] = X_{\max} \frac{\ln \left[1 + \mu \frac{|x(n)|}{X_{\max}} \right]}{\ln[1 + \mu]} \text{sign}[x(n)]$$

Un valor típico de μ es 255 y $X_{\max} = 2048$ en este caso. A continuación se modificará el cuantizador uniforme para señales de voz desarrollado anteriormente para implementar el cuantizador ley μ de la figura 2. Comparar su comportamiento con el del cuantizador uniforme estudiado anteriormente para 8, 6 y 4 bits, mediante medida de la SNR y audición.

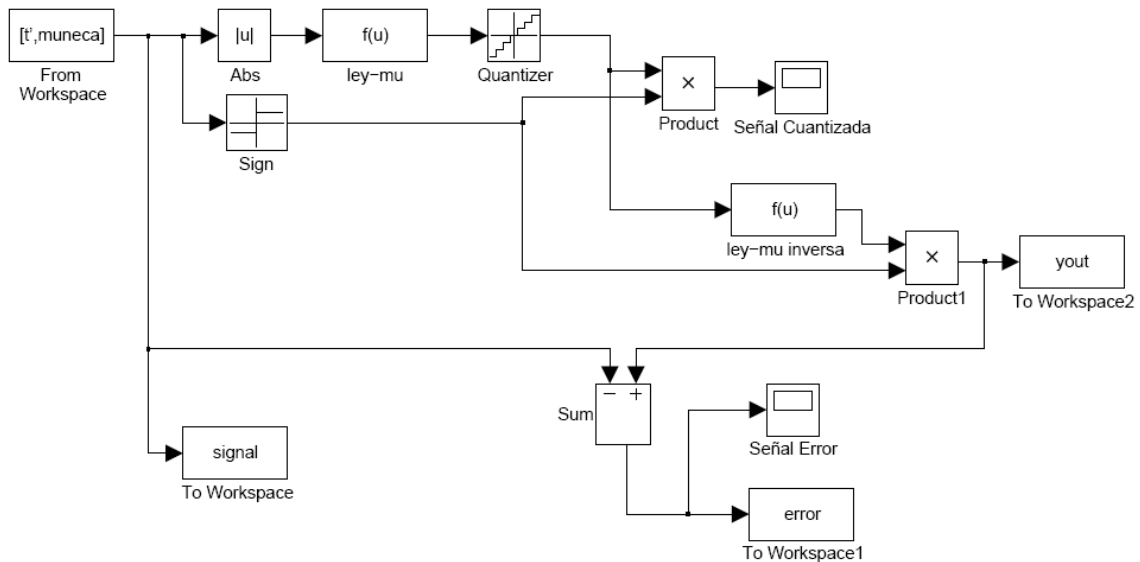


Figura 2. Cuantización de una señal de voz mediante ley μ .

4.3. Modulación delta

Mediante modulación delta, cada muestra de señal queda representada por un único bit que indica si la señal se ha incrementado o decrementado respecto a la muestra anterior. A continuación, se construirá el modulador delta de la figura 3. En el generador de señal, se fijará una onda senoidal de frecuencia 1 Hz y amplitud 1. El módulo “Pulse Generator” genera un tren de impulsos de amplitud unitaria a la frecuencia de muestreo F_s requerida (en este caso será de 50 Hz, es decir, un periodo de $T_s = 0.02$ seg) que se encarga de realizar el muestreo de la señal propiamente dicho. La anchura τ de los pulsos de este tren puede ser cualquiera entre 0 y T_s (fijar, por ejemplo, a $\tau = T_s/2$). A partir de este esquema, deben estudiarse los siguientes puntos:

- Demostrar que el valor del cuanto es $q = G\tau$, donde G es el factor de ganancia introducido por el módulo correspondiente.
- Sabiendo que la amplitud máxima X_{\max} que puede tener la señal de entrada para que el modulador funcione correctamente es,

$$X_{\max} = \frac{qF_s}{2\pi f}$$

donde f es la frecuencia de la señal sinusoidal de entrada, deducir el valor más pequeño permitido para el valor de la ganancia G . Describir el efecto que se produce cuando la ganancia se fija por debajo de ese valor mínimo.

- Fijando el valor de ganancia como $G = 20$, compruébese el efecto de variación de la frecuencia de muestreo. Determinar el valor mínimo de la misma para que no haya sobrecarga de pendiente.
- Describir qué sucede cuando se usan otras formas de onda, como un diente de sierra. Un demodulador delta consta de un módulo de ganancia y un integrador (lo mismo que en la rama de realimentación del modulador) más un filtro pasa-baja Butterworth. Añadir este demodulador al sistema desarrollado y describir su funcionamiento.

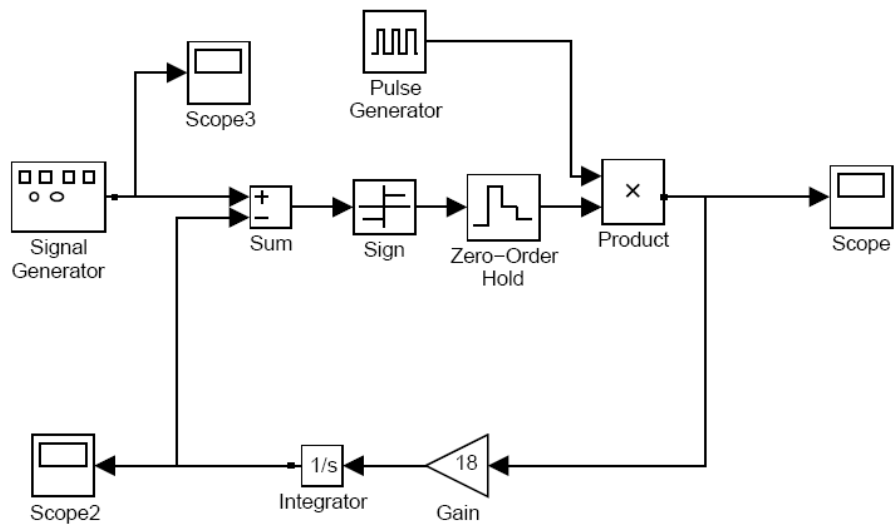


Figura 3. Modulador delta.

4.4. Modulación delta de voz (opcional)

Se construirá un modulador delta para la señal de voz de los apartados anteriores para una frecuencia de muestreo F_s de 8 KHz y se medirá la SNR entre la señal original y la demodulada. Ha de tenerse en cuenta que la máxima componente en frecuencia de la señal de entrada es $f = 4000$ Hz y que $X_{\max} = 2048$ a la hora de seleccionar el factor de ganancia del modulador y demodulador. Para poder comparar con las representaciones de las secciones anteriores ha de tenerse en cuenta el bit-rate de la representación, es decir, el número de bits requeridos por segundo. Así, por ejemplo, la representación con cuantización uniforme o ley μ con 4 bits a 8 KHz tiene un bit-rate de $4 \times 8000 = 32000$ bits por segundo, mientras que con modulación delta a la misma frecuencia de muestreo es de $1 \times 8000 = 8000$ bps.