Laboratorio de Tratamiento Digital de Señales

Curso 2020-2021. Sesión 3

Muestreo de señales en tiempo continuo

El muestreo de una señal en tiempo continuo $x_c(t)$ es el proceso mediante el que se obtiene una señal discreta $x_d[n]$ que verifica

$$x_d[n] = x_c(nT_s),\ n \in \mathbb{Z}$$

donde T_s es el periodo de muestreo, medido en segundos. $F_s = \frac{1}{T_s}$ es la frecuencia de muestreo medida en Hz o muestras/segundo. Por tanto, la secuencia de números $x_d[n]$ son los valores de la amplitud de la señal $x_c(t)$ en los instantes temporales nT_s.

Los teoremas de muestreo establecen los valores adecuados de T_s para que la secuencia de medidas represente fielmente a la señal $x_c(t)$. En el caso particular en que la señal $x_c(t)$ sea de banda limitada, con ancho de banda F_0, el periodo de muestreo máximo debe ser $T_s = \frac{1}{2F_0}$.

En el entorno MATLAB el muestreo de una señal $x_c(t)$ definida mediante una función $f(t)$, en el intervalo temporal $[T_1\ T_2]$, y con un período de muestreo T_s, se puede realizar con las siguientes líneas de código

```
t=T1:Ts:T2-Ts;
x=xc(t);
```

El vector `t` contiene los instantes de muestreo (retícula temporal). En la segunda línea se calculan los valores de $x_c(t)$ en los distintos instantes de muestreo.

Ejercicios de la sesión 3 del laboratorio de TDSÑ

1. Muestreo de sinusoides

Considere el siguiente conjunto de sinusoides:

$$x(t) = A\cos(2\pi F_0 t),\ F_0 = 200,\ 850,\ 1850,\ 3800\ \ Hz,\ \ A = 1$$

1.1. Muestree las señales $x(t)$ con una frecuencia de muestreo $F_s = 8000\ Hz$. Dibuje (con `plot(. . .)`, dividiendo la pantalla en cuatro mediante `subplot(2,2,x)`, `x=1, ..., 4`) un segmento de duración $T = 10\ \text{ms}$.

Una solución para este sub-apartado podría ser:

```
% Datos
Frecuencias=[200 850 1850 3800]; A=1;
Tv=10e-3;
Fs=8000; Ts=1/Fs;

% Rejilla temporal para el muestreo
t=0:Ts:Tv-Ts;

% Cálculo y dibujo de las sinusoides
n=1;
for F0=Frecuencias
    subplot(2,2,n)
    plot(t,A*cos(2*pi*F0*t))
    title(['cos(2\pi' num2str(F0,4) 't)']), xlabel('t'), grid
    n=n+1;
end
subplot
```

Observe como con una estructura de repetición `for . . . end` calculamos y pintamos las cuatro señales. También observe que se cierran los cálculos con el comando `subplot` para deshacer la partición de la ventana gráfica.

1.2 Repita el apartado 1.1 pero con una frecuencia de muestreo $F_s = 6000\ Hz$ y el siguiente conjunto de señales:

$$x(t) = A\cos(2\pi F_0 t),\ F_0 = 200,\ 850,\ 1850,\ 5800\ \ Hz,\ \ A = 1$$

1.3 Repita el apartado 1.1 pero con una frecuencia de muestreo $F_s = 6000\ Hz$ y el siguiente conjunto de señales:

$$x(t) = A\sin(2\pi F_0 t),\ F_0 = 200,\ 850,\ 1850,\ 5800\ \ Hz,\ \ A = 1$$

Debe analizar con detenimiento los resultados gráficos, e intentar comprenderlos con ayuda de la teoría de muestreo. Concretamente:

a.- ¿Por qué aparece un efecto parecido a una modulación de amplitud?

b.- ¿Por qué hay un cambio de signo de una de las señales en los apartados 1.2 y 1.3?

2. Interpolación

En el fichero `tds.mat` dispone de la señal correspondiente a la frase "**tratamiento digital de señales**", con una frecuencia de muestreo de 8000 Hz. Sin embargo, suponga que tenemos un reproductor que sólo admite una frecuencia de muestreo de 24000 Hz e intente reproducir la señal con los comandos `p=audioplayer(int16(tds),24000); play(p)`. Al oír la señal se hace evidente la necesidad de interpolar nuestra grabación por un factor L=3 (24000/8000=3). Para realizar esta interpolación se procede de la siguiente manera.

Nota: en vez de poner cada subapartado en una celda, póngalos todos en la misma celda e inserte el comando `pause` al final del código de cada subapartado; de esta forma, cuando ejecute la celda del apartado 2, podrá avanzar por los subapartados presionando la tecla INTRO.

2.1 Expansión

Expanda la señal `tds` por un factor L=3. Una forma de realizar una expansión en MATLAB es:

```
tdsL = zeros(L*length(tds), 1); %Primero se crea la base de ceros
tdsL(1:L:length(tdsL)) = tds; %Luego se coloca señal entre ellos
```

Dibuje, con `stem(. . .)`, un segmento de 30 puntos de `tdsL` a partir del índice n=3750, y compruebe que hay "L-1 ceros entre cada dos muestras". Escuche la señal `tdsL` y aprecie el ruido "metálico y de alta frecuencia" debido a la expansión.

El comando MATLAB para oír `tdsL` sería:

```
p=audioplayer(int16(tdsL), 24000); play(p);
```

2.2. Filtro interpolador

Para completar la interpolación se filtra la señal expandida mediante un filtro paso bajo ideal con frecuencia de corte $\omega_c = \frac{\pi}{L}$ y amplitud L. La respuesta impulsiva de este filtro es

$$h_i[n] = \begin{cases} \dfrac{sen(\omega_c n)}{\omega_c n}; & n \neq 0 \\ 1; & n = 0 \end{cases}$$

Una buena aproximación se obtiene truncando la respuesta impulsiva entre -100 y 100. Con MATLAB sería:

```
n = -100:100;

wc = pi/L;

h = sin(wc*n)./(wc*n);

h(isnan(h)) = 1;
```

Explique la operación del tercer comando MATLAB.

Dibuje en primer lugar con `plot(. . .)` toda la respuesta impulsiva con el eje horizontal entre -100 y 100, y después sólo 41 puntos en torno al máximo con `stem(. . .)`. Observe como $h[Lk]=0$.

2.3 Filtrado interpolador

Filtre la señal expandida `tdsL` con el comando

```
tdsLi = filter(h, 1, tdsL). % (B = h, A = 1)
```

Escuche la señal `tdsLi` y compruebe que se ha suprimido el ruido "metálico y de alta frecuencia".

"Corte y pegue" las siguientes líneas de código en su *script*:

```
% Visualización de un segmento de 30 puntos de todas las señales
Npuntos=30;
Ncomienzo=1250; % punto de comienzo del tramo de señal original
n=L*Ncomienzo+1:L*Ncomienzo+Nputos;%sumo 1 pues en MATLAB índices comienzan en n=1
nnn=L*Ncomienzo+1:L:L*Ncomienzo+Nputos;
nn=Ncomienzo+1:Ncomienzo+length(nnn);
[hmax,nmax]=max(h);
N=     ;
plot(nnn,tds(nn),'b')
hold
stem(n,tdsLi(n+N),'g')
stem(n,tdsL(n),'r')
hold
grid, xlabel('n')
legend('tds(t)','tdsLi[n]','tdsL[n]','Location','NorthWest')
```

Estas líneas de código representan sendos segmentos de longitud $Npuntos=30$ de `tdsL`, `tdsLi` y de la señal original $tds(t)$. Trate de entender por qué se definen así las diversas rejillas temporales: n, nn y nnn. (Se introduce $+N$ en la octava y décima líneas del código para compensar el retardo de N muestras que introduce el filtro interpolador, ¿cuánto vale ese retardo?).

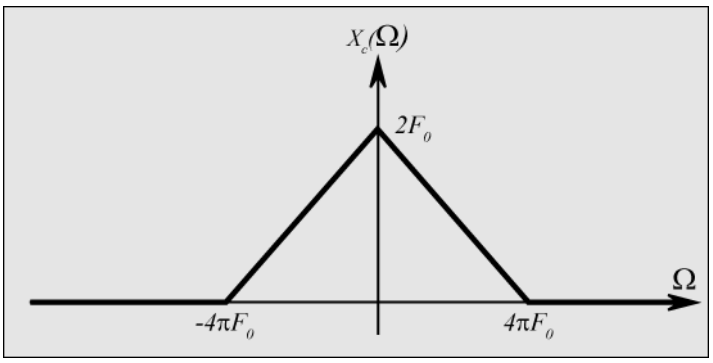
Se aprecia claramente que ambas señales coinciden en los índices kL y que el filtro interpolador "levanta" los ceros que introduce el expansor hasta la posición correcta sobre la gráfica de $tds(t)$.

3. Diezmado de una señal de banda limitada

La señal:

$$x_c(t) = \begin{cases} \left(\dfrac{sen(2\pi F_0 t)}{\pi t} \right)^2, & t \neq 0 \\ 4F_0^2, & t = 0 \end{cases}$$

es de banda limitada a $2F_0$ Hz, estando su espectro representado en la siguiente figura:



Considere la secuencia de duración finita (2 segundos):

$$x[n] = x_c(nT_s), \ -1 \leq nTs \leq 1$$

donde $T_s = \dfrac{1}{12F_0}$. Esta secuencia se puede considerar de banda limitada (aproximadamente, ya que estamos muestreando en una ventana de 2 segundos solamente), con $\omega_0 = \dfrac{\pi}{M}$, por lo que su "redundancia" es de $M - 1$ muestras de cada M. **¿Cuánto vale M?**

3.1 Cálculo y dibujo de $x(n)$.

Suponiendo $F_0 = 1\ KHz$, calcule la señal $x[n]$. Dibuje con el comando `stem(. . .)`.la señal calculada, pero solo en el rango de índices [-30 30].

Sugerencias: a) calcule una retícula temporal con los instantes de muestreo. b) evite el resultado `NaN` que se produce al dividir por cero mediante el mismo procedimiento que se utilizó en el apartado 2.2.

3.2 Cálculo de $X(\omega)$

Calcule y dibuje, en escala lineal, el módulo de $X(\omega)$ en el intervalo $[0\ 2\pi]$. Para el cálculo de $X(\omega)$ emplee:

```
[X w]=freqz(x,1,'whole'); % Cálculo de la TF mediante muestreo de la TZ
```

Observe el espectro resultante y compruebe que el ancho de banda es $\frac{\pi}{M}$. Mida el máximo de $|X(\omega)|$ y verifique que **su valor es el correcto**.

3.3 Espectro de una señal expandida

Considere la señal

$$x_e(n)=\begin{cases} x\left(\frac{n}{L}\right), & n=\dot{L} \\ 0, & \text{resto} \end{cases}$$

Calcule y dibuje, en escala lineal, el módulo de $X_e(\omega)$ en el intervalo $[0\ 2\pi]$. Para la operación de expansión emplee:

```
L=3;
xe=zeros(L*length(x),1);
xe(1:L:length(xe))=x;
```

Compruebe que el resultado obtenido coincide con el resultado esperado, teniendo en cuenta la operación en frecuencia del expansor discreto.

3.3 Espectro de una señal comprimida

Considere la señal

$$x_d[n]=x[2n]$$

Calcule y dibuje el módulo de $X_d(\omega)$ en el intervalo $[0\ 2\pi]$.

Para la operación de compresión emplee:

```
M=2;
xd=x(1:M:length(x));
```

Compruebe que el resultado obtenido coincide con el resultado esperado, teniendo en cuenta la operación en frecuencia del compresor discreto. ¿Se ha producido "aliasing" al comprimir la señal $x[n]$? ¿Por qué?

4. Cuantificación uniforme

La señal `tds` se obtuvo con una frecuencia de muestreo de $8000\ \text{Hz}$ y con una resolución de 16 bits, por lo que podemos considerar $X_m = 32768$. A continuación procederemos a "recuantificar" la señal `tds` con $N = 8$ bits y a analizar el error de cuantificación. El cuantificador Q[] que aplicaremos a `tds` será del tipo de la figura 4.54 de la referencia principal[1] de la asignatura TDSÑ.

4.1. Cuantificación

Cuantifique las muestras de la señal `tds` con $N = 8$ bits de resolución. Para realizar la cuantificación se sugieren los siguientes comandos:

```
d = 2*Xm/2^N; % tamaño del escalón

tdsQ = d*round(tds/d); % cuantificador por redondeo al nivel más cercano.
```

Analice cómo estos dos comandos pueden obtener la señal cuantificada `tdsQ`. Dibuje en la misma ventana gráfica, utilizando el comando `subplot(. . .)`, la señal sin cuantificar `tds`, la señal cuantificada `tdsQ` y el error de cuantificación `e = tdsQ - tds`.

Oiga las señales con

```
p=audioplayer([int16(tds)' zeros(1,4000) int16(tdsQ)' zeros(1,4000)
int16(e)'],Fs);
play(p)
```

y aprecie las diferencias o similitudes.

4.2. Relación Señal a Ruido

Calcule la relación SNR total al cuantificar la señal `tds` y compare el resultado con la aproximación teórica $SNR = 6N - 7.2$. Explique el origen de la discrepancia.

4.3. Relación Señal a Ruido Segmental

Calcule y dibuje la relación SNR segmental, que se define como la SNR aplicada a segmentos consecutivos de la señal según se ha visto en prácticas anteriores. Considere segmentos de 20 ms de duración y solapados un 40%.

---

[1] Alan V. Oppenheim, Ronald W. Schafer: "Discrete-Time Signal Processing", (3rd Edition), Ed. Prentice Hall

Dibuje, dividiendo la ventana gráfica con `subplot(2,1,x)`, la señal original `tds` y la SNR segmental. Sobre la gráfica de la SNR segmental trace una línea horizontal de altura igual a la SNR global, con un comando MATLAB del estilo:

```
subplot(2,1,2), plot(tSNRseg,SNRseg,'g',[tSNRseg(1) tSNRseg(end)],[SNRglobal SNRglobal],'r')
```

Comente los resultados. A la vista del resultado anterior. ¿A qué se debe que la SNR segmental alcance valores tan altos?

4.4. Histograma de Amplitudes

Calcule y dibuje el histograma de amplitudes normalizado como estimación de la densidad de probabilidad de distribución de amplitudes del error de cuantificación $e[n]$. Compare la estimación obtenida con el supuesto teórico del cuantificador uniforme. Los comandos MATLAB recomendados para realizar este apartado son:

```
M = 50;
[h, b] = hist(e, M);
bar(b, h/(length(e)*d/M));
hold on
plot(b, (1/d)*ones(size(b)),'g')
title('Diagrama de barras del histograma de amplitudes y densidad de
distribución de amplitudes teórica')
xlabel('e')
hold off
```

siendo M el número de bins. Explique la necesidad del factor de normalización del histograma `1/(length(e)*d/M)`.

Compruebe que los datos de la figura sean correctos.

4.5. Autocorrelación y densidad espectral de potencia de $e[n]$

En este apartado completamos el estudio práctico de la corrección del modelo estocástico aplicado al proceso de cuantificación analizando la señal de autocorrelación y la densidad espectral de potencia del ruido de cuantificación $e[n]$.

El modelo estocástico asume que el error de cuantificación es un ruido blanco con densidad de distribución de amplitudes uniforme. Si Δ es la altura del escalón del cuantificador, $2X_m$ el margen dinámico de la entrada del cuantificador y N el número de bits del cuantificador:

$$\Delta = \frac{2X_m}{2^N}\ ,\ \sigma_e^2 = \frac{\Delta^2}{12}$$

$$p_e(e) = \begin{cases} \frac{1}{\Delta}\ , & -\frac{\Delta}{2} \le e \le \frac{\Delta}{2} \\ 0\ , & resto \end{cases}$$

$$\phi_{ee}[m] = \sigma_e^2 \delta[m]$$

$$P_{ee}(\omega) \equiv \Phi_{ee}(\omega) = \sigma_e^2 \ , -\pi \le \omega \le \pi$$

En donde hemos llamado $P_{ee}(\omega)$ a la densidad espectral de potencia, es decir, lo que en la asignatura teórica solemos llamar $\Phi_{ee}(\omega)$. $\left(\phi_{ee}(m) \xleftrightarrow{F} \Phi_{ee}(\omega) \equiv P_{ee}(\omega)\right)$.

Calcule la autocorrelación de la señal $e[n]$ con el comando MATLAB `xcorr(. . .)`. No olvide normalizar dividiendo el resultado por la longitud del vector e. Calcule una estimación de la densidad espectral de potencia $P_{ee}(\omega)$ con los comandos MATLAB

```
[Pee w]=freqz(ree,1,'whole');%TF de la autocorrelación considerada determinista
Pee=abs(Pee);
```

Dibuje en una misma ventana gráfica la autocorrelación y la densidad espectral de potencia. Para $\phi_{xx}[m]$ emplee el comando `stem(. . .)` y dibuje sólo el intervalo de índices [-30 30]. Para $P_{ee}(\omega)$ emplee el comando `plot(. . .)` y utilice escala logarítmica ($10 * \log 10$ ya que estamos tratando con potencias). Superponga una línea horizontal de altura el valor teórico de $P_{ee}(\omega)$.

Ejercicios adicionales de la tercera sesión de laboratorio

2.1 La salida de un sistema, medida con un sensor de voltaje, cuando a la entrada se le aplica un escalón de tensión $v_i(t) = V\,u(t)$ viene dada por la fórmula

$$v_o(t) = V\,t^3\,e^{-t^2}u(t)$$

Se quiere medir el valor máximo que alcanza $v_0(t)$ y el instante t_0 en que se produce este máximo. Para ello:

- Muestree la señal $v_0(t)$ con un periodo de muestreo tal que el error que se cometa al medir t_0 sea inferior o igual a $1\ ms$. Considere un intervalo de muestreo [0 5]s y $V = 10\ vol$. Dibuje con el comando `plot(. . .)` la señal muestreada.
- Mida $v_0(t_0)$ y t_0 con `ginput` y preséntelos por pantalla.
- Calcule y presente por pantalla los valores exactos que se obtienen calculando los nulos de la derivada primera de $v_0(t)$.
- La precisión de las medidas con `ginput` está limitada por la resolución gráfica del terminal. Calcule $v_0(t_0)$ y t_0 numéricamente con ayuda de la función `max(. . .)`. Presente por pantalla el error cometido en el cálculo de t_0 y compárelo con el objetivo de diseño ($\le 1\ ms$).

2.2 Señales cardiográficas

Abra con el editor de MATLAB el fichero `cardio100.m`. Como podrá observar leyendo los comentarios iniciales, en este fichero se almacenan dos señales cardiográficas, en concreto las derivaciones MLII y V5 y los instantes de muestreo de estas señales. Ver el URL http://es.wikipedia.org/wiki/Electrocardiograma.

Visualización. Ejecute el comando cardio100. Obtendrá una matriz de dimensiones 3600 x 3 llamada `cardio`. La primera columna (en MATLAB `t=cardio(:,1))` es la base de tiempos, la segunda columna `cardio(:,2)` es la derivación MLII y la tercera columna `cardio(:,3)` es la derivación V5. Dibuje con `plot(. . .)` los primeros 10 segundos de la señal $V_5(t)$.

Cálculo del ritmo cardiaco (en latidos por minuto, lpm). Mida con `[tMx, Mx]=ginput` los máximos de los complejos QRS. Habrá obtenido un vector `Mx` con los valores de los máximos y un vector de tiempos `tMx` correspondientes con los máximos de los complejos QRS.

Dibuje con `stem(. . .)` el ritmo cardiaco pulso-a-pulso medido en lpm. Esta señal se obtiene con el comando MATLAB `60./diff(tMx)`. Presente por pantalla el ritmo cardiaco medio, calculado mediante `60/mean(diff(tMx))`, durante los 10 primeros segundos de la grabación del ECG.

¿Detecta alguna anomalía en la señal $V_5(t)$?

2.3 Señal de audio musical.

En el fichero `Invention8.wav` se encuentran las muestras de un segmento de los dos primeros compases de la **Invention 8** en Fa mayor de J. S. Bach.

2.3.1 Lea la señal con el comando MATLAB `audioread(. . .)`. No olvide leer la frecuencia de muestreo, la necesitará posteriormente. Escuche la señal con el comando `sound(. . .)` y represéntela con el comando `plot( . . )`.

2.3.2 Calcule la potencia localizada de la señal con una ventana de duración $10\ ms$ y solapamiento 25%. Dibuje la señal y la potencia localizada en una misma ventana gráfica.

2.3.3 Las primeras seis notas musicales son

$$Fa_4,\ La_4,\ Fa_4,\ Do_4,\ Fa_4 \text{ y } Fa_5.$$

A partir de la gráfica de la potencia localizada segmente las notas y calcule el espectro de cada una de ellas mediante el siguiente comando MATLAB:

```
[Nota F]=freqz(nota,1,1e5,Fs),
```

donde `nota(.)` es la señal de una nota musical. Dibuje en escala logarítmica el módulo del espectro. Calcule la frecuencia correspondiente al máximo absoluto de $\left|Nota(F)\right|$ (lo puede hacer

gráficamente con `ginput` –no se olvide de amplificar la zona del espectro en torno al máximo antes de medir- o con la ayuda del comando `max(. . .)`:

```
[Mx Ind]=max(abs(Nota));
Fnota=F(Ind);
```

Compruebe los resultados obtenidos con la ayuda de una tabla de frecuencias de las notas musicales (que se puede obtener fácilmente en Internet, consulte por ejemplo http://www.lpi.tel.uva.es/~nacho/docencia/ing_ond_1/trabajos_05_06/io2/public_html/escalas.html).