

## 1. ANEXO COMPLEMENTARIO.

**1.1 Uso de funciones.**

## 1.1.1 Representar el espectro de una señal.

*pwelch(signal,window,[],points,fs)*

- signal: Nombre de la variable a analizar
- window: Ventana empleada para calcular la fft (hanning,hamming,kaiser,boxcar,...)
- points: Número de puntos de la fft (se aconseja emplear más de 32768 puntos)
- fs: Frecuencia de muestreo

Ejemplo:

Dada la variable “y” que almacena los valores de la señal a analizar, muestreada a una frecuencia de 80MHz, para representar el espectro con 32768 puntos y una ventana kaiser con factor  $\beta=20$ :

*pwelch(y,kaiser(N,20,[],32768,80e6)*

## 1.1.2 Calcular la relación señal-ruido más distorsión (SNDR).

*fsnr(signal,l,points,fs,fi,BW,beta,rip>window,kspec,2,nbin)*

- signal: Nombre de la variable a analizar
- points: Número de puntos de la fft (se aconseja emplear más de 32768 puntos)
- fs: Frecuencia de muestreo
- fi: Frecuencia del tono de señal
- BW: Ancho de banda de la señal
- beta: Factor beta de la ventana de kaiser (si no se usa tal ventana puede tomar cualquier valor).
- rip: Factor de atenuación de la ventana de Chebyshev (si no se usa tal ventana puede tomar cualquier valor).
- window: Ventana empleada para calcular la fft. Posibles valores:
  - 1: Ventana de Kaiser
  - 2: Ventana de Barlett
  - 3: Ventana de Blackman
  - 4: Ventana de Hamming
  - 5: Ventana de Hanning
  - 6: Ventana de Chebyshev
  - 7: Ventana rectangular
  - 8: Ventana triangular
- kspec: Tipo de espectro. Posibles valores:
  - 1: Paso de baja
  - 2: Paso de banda
- nbin: Número de bins en frecuencia considerados para calcular la potencia del tono de entrada. Debe ser un número impar.

Ejemplo:

Dada la variable “y” que almacena los valores de la señal LP a analizar a 1MHz, muestreada a una frecuencia de 80MHz, en un ancho de banda de 2 MHz, para calcular la SNDR con 32768 puntos, una ventana kaiser con factor  $\beta=20$  y considerando 15 bins en frecuencia para calcular la potencia de la señal:

*fsnr(y,1,32768,80e6,1e6,2e6,20,0,1,1,2,15)*

## 1.2 Realizar simulaciones iterativas desde la consola de comandos en Matlab.

Dado un modelo de simulink “model1.mdl” es posible ejecutar la simulación de dicho modelo desde la consola de comandos de Matlab usando el comando “sim”. Este comando puede resultar muy útil para llevar a cabo análisis de Montecarlo o paramétricos, donde se requiere simular en múltiples ocasiones un mismo modelo.

### 1.2.1 Análisis paramétricos.

Supongamos que se desea realizar un análisis paramétrico de un modulador  $\Sigma\Delta$  descrito en Simulink mediante un modelo llamado “c211\_FEIp.mdl”. Se desea evaluar la *SNDR* de la señal de salida del modulador (variable *y*) en función de la amplitud de la señal de entrada (parámetro *A*), que será barrida desde un valor inicial 0.01 hasta un valor final 0.95 en escala logarítmica tomando 40 puntos. Para ello, puede ejecutarse un script similar al siguiente en Matlab:

```

iter=40;
initial_value=0.01;
final_value=0.95;
Avalues=logspace(initial_value,final_value,iter);
for i=1:iter
    A=Avalues(i);
    sim('c211_FEIp.mdl');
    vsndr(i)=fsnr(...);
end
plot(Avalues,vsndr)

```

### 1.2.2 Análisis de Montecarlo.

Supongamos que se desea realizar un análisis de Montecarlo de 100 iteraciones de un modulador  $\Sigma\Delta$  descrito en Simulink mediante un modelo llamado “c211\_FEIp.mdl”. Se desea evaluar la *SNDR* de la señal de salida del modulador (variable *y*) considerando un desajuste entre los condensadores (parámetros *ca*, *cb* y *cc*), que puede modelarse como un error relativo de media cero y desviación estándar,  $\sigma=1\%$ . Para ello, puede ejecutarse un script similar al siguiente en Matlab:

```

iter=100;
ca0=1e-12; %nominal value for capacitor ca
cb0=1e-12; %nominal value for capacitor cb
cc0=1e-12; %nominal value for capacitor cc
sigma=0.01;
vsndr=zeros(iter,1);
for i=1:iter
    ca=ca0*(1+randn*sigma);
    cb=cb0*(1+randn*sigma);
    cc=cc0*(1+randn*sigma);
    sim('c211_FEIp.mdl');
    vsndr(i)=fsnr(...);
end
hist(vsndr) %histogram of the SNDR

```