

ANEXO III: Scripts de Matlab. Análisis ENOB

Archivo: test_bench_Par_ENOB

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Proyecto Fin de Carrera
%
% Autor: Fco Javier Segovia de la Torre
%
% Nombre del proyecto: Convertidor Analógico digital basado en tiempo
%
% Scripts: Análisis paramétrico para analizar la viabilidad de la arquitectura
%
% Descripción: Función para obtener la ENOB en el model Matlab/Simulink del ADC
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Funcion para obtener la ENOB en el modelo simulink del ADC
function [enob1,sndr1]=test_bench_Par_ENOB(Iref,C1,C2,Tdelay,V_offset,Depend_offset,Nitems)

global Iref C1 C2 Tdelay V_offset Depend_offset Iin; %Variable globales: utilizadas por otras
echo on; %funciones. Necesarias en el WorkArea.

Iin=50e-9;
Iref=Iref_;
C1=C1_;
C2=C2_;
Tdelay=Tdelay_;
V_offset=V_offset_;
Depend_offset=Depend_offset_;
echo off;

Nbins=Nitems;
Fs = 1/220e-6; % Frecuencia de muestreo
t = 0:1/Fs:Nbins/Fs-1/Fs; % Vector de Frecuencias
Fin=11/Nbins*Fs; % Frecuencia de entrada
x = cos(2*pi*t*Fin); % Coseno que se aplicara a la entrada

FSper=0.9; % Nivel del FS (fondo de escala ) en la entrada
x2=FSper*x; % Coseno de amplitud 90% del Fondo de escala
Iin_vec=0.5*4*Iref+0.5*4*Iref*x2; % Vector de corriente de entrada: coseno

NIin=length(Iin_vec);
Dout_vec=zeros(1,NIin);
Idealin=Iin_vec/(Iref*4)*1024;

for Nconv=1:NIin % Obtención del dato de salida del modelo simulink cada
% vez que se realiza una conversión

    Iin=Iin_vec(Nconv);
    sim('test_bench_Par'); % Llamada al modelo implementado en Simulink para
% cada valor de corriente de entrada del vector Iin_vec

    data=yout(:,1);
    env_conv=yout(:,2);
    input=yout(:,3);
    timesim=tout;
    N=length(data);
    flag=0;
    Ndata=0;
    for I=1:N

        if(env_conv(I)==1&&flag==0) % Cada vez que env_conv pasa a 1 tenemos
            flag=1; % un dato nuevo
            Ndata=Ndata+1;

        elseif (env_conv(I)==0&&flag==1)
            flag=0;
        else
            Ndata=Ndata;
        end

    end

    % Ndata contiene el número de datos convertidos debe ser uno
    % Puesto que en el modelo simulink solo hacemos una conversión
    dataout=zeros(1,Ndata);
    datain=zeros(1,Ndata);
    tsampling=zeros(1,Ndata);
    J=0;
    for I=1:N

        if(env_conv(I)==1&&flag==0)
            flag=1;
            J=J+1;
            dataout(J)=data(I);
            datain(J)=input(I);
            tsampling(J)=timesim(I);

        elseif (env_conv(I)==0&&flag==1)
            flag=0;
        else
            J=J;
        end

    end

    disp(sprintf('\n ... Percentage of the simulation %f % \n',100.0*Nconv/NIin)); %Información sobre el % de simulación realizado

    Dout_vec(Nconv)=dataout(1); %Aquí tomamos el dato

end

[enob1,sndr1]=enob2(Dout_vec,Fs,11,NIin); %Llamada a enob2 para calcular la ENOB propiamente dicha

enob1
sndr1

```

Archivo: enob2

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Proyecto Fin de Carrera
%
% Autor: Fco Javier Segovia de la Torre
%
% Nombre del proyecto: Convertidor Analógico digital basado en tiempo
%
% Scripts: Análisis paramétrico para analizar la viabilidad de la arquitectura
%
% Descripción: Función para obtener la ENOB en el model Matlab/Simulink del ADC
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [X,Y]=enob2(vo,fs,M,N)

global Iref C1 C2 Tdelay V_offset Depend_offset
% Esta función calcula la resolución efectiva de bits (ENOB) de la
% señal vo. Fs es la frecuencia de muestro (sampling frequency), N es el
% número de muestras y M es el operador de la frecuencia de
% la señal de entrada, donde fin=M*fs/N.

index=N/M;
fn=M*fs/N;
[px,f]=pwelch(vo, hamming(length(vo)), [], length(vo), fs); % Método Pwelch

bin_l=fs/(2*length(px)); % longitud del bin
bin_sig=floor((fn/bin_l))+1; % posición del bin de señal
bin_stop=floor(((fs/2)/bin_l));
if index>50
    bin_sig1=bin_sig-5; % desde que bin empieza la señal
    bin_sig2=bin_sig+5; % en que bin acaba la señal
    noise=10*log10(sum(px((bin_sig2+1):bin_stop)));
    signal=10*log10(sum(px(bin_sig1:bin_sig2)));
else if index>2.1
    bin_sig1=bin_sig-5; % desde que bin empieza la señal
    bin_sig2=bin_sig+5; % en que bin acaba la señal
    noise=10*log10(sum(px((bin_sig2+1):bin_stop))+sum(px(10:bin_sig1-1)));
    signal=10*log10(sum(px(bin_sig1:bin_sig2)));
else
    bin_sig1=bin_sig-5; % desde que bin empieza la señal
    bin_sig2=bin_sig+5; % en que bin acaba la señal
    noise=10*log10(sum(px(5:bin_sig1-1)));
    signal=10*log10(sum(px(bin_sig1:bin_sig2)));
end
end

snr=signal-noise;
enob=(snr-1.76)/6.02; % ENOB

enob_str=sprintf(' ENOB = %f',enob);
disp(enob_str);
figure();
plot(f,10*log10(px));
text(0,5,sprintf('\nIref %g\nC1 %g\nC2 %g\nTdelay %g\nV_offset %g\nDepend_offset %g',Iref,C1,C2,Tdelay,V_offset,Depend_offset));
title(['Power Spectral Density estimate via Welch method' enob_str]);
xlabel('Frequency (kHz)');
ylabel('Power/frequency (dB/Hz)');

X=enob;
Y=snr;

```

Archivo: test_bench_Par_ENOB_Tdelay

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Proyecto Fin de Carrera
%
% Autor: Fco Javier Segovia de la Torre
%
% Nombre del proyecto: Convertidor Analógico digital basado en tiempo
%
% Scripts: Análisis paramétrico para analizar la viabilidad de la arquitectura
%
% Descripción: Función para obtener la ENOB haciendo un análisis parametrizando
% el retraso del comparador del modelo del ADC de Matlab/Simulink
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Función para obtener la ENOB en función de diferentes parámetros (Retardo) del comparador, Tdelay)

global Iref C1 C2 Tdelay V_offset Depend_offset lin; % Variables globales para obtener los
% valores del WorkArea
Tdelay_vec=[0 2e-9 10e-9 100e-9 1000e-9 10000e-9]; % Vector de retrasos. Parametrización del retraso Tdelay
NTdelay=length(Tdelay_vec);

enob_Tdelay=zeros(1,NTdelay);
snr_Tdelay=zeros(1,NTdelay);

for I=1:NC1
    [enob_Tdelay(I),snr_Tdelay(I)]=test_bench_Par_ENOB(125e-9,5e-12,5e-12,Tdelay_vec(I),0,64); % Llamada a la función test_bench_Par_INL
% para cada valor de retraso
end

```

Archivo: test_bench_Par_ENOB_Cmistmach

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Proyecto Fin de Carrera
%
% Autor: Fco Javier Segovia de la Torre
%
% Nombre del proyecto: Convertidor Analógico digital badado en tiempo
%
% Scripts: Análisis paramétrico para analizar la viabilidad de la arquitectura
%
% Descripción: Función para obtener la ENOB haciendo un análisis parametrizando
% el mismach de capacidades del modelo del ADC de Matlab/Simulink
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Función para obtener la ENOB en función de diferentes de la dispararidad de capacidades

global Iref C1 C2 Tdelay V_offset Depend_offset Iin; % Variables globales para obtener los
% valores del WorkArea

NC1_vec=[4.75e-12 4.85e-12 5.05e-12 5.15e-12 5.25e-12]; % Vector de capacidades
NC1=length(NC1_vec);

enob_Tdelay=zeros(1,NC1);
sندر_Tdelay=zeros(1,NC1);

for I=1:NC1
    [enob_Tdelay(I),sندر_Tdelay(I)]=test_bench_Par_ENOB(125e-9,NC1_vec(I),5e-12,0,0,0,64); %Llamada a la función test_bench_Par_ENOB
    %para cada valor de retraso y de capacidad
end

```

Archivo: test-bench_Par_ENOB_T_delay_V_offset_Depend_offset

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Proyecto Fin de Carrera
%
% Autor: Fco Javier Segovia de la Torre
%
% Nombre del proyecto: Convertidor Analógico digital badado en tiempo
%
% Scripts: Análisis paramétrico para analizar la viabilidad de la arquitectura
%
% Descripción: Función para obtener la ENOB haciendo un análisis parametrizando
% el retraso del comparador , el offset del comparador y del offset dependiente
% de la señal en el comparador del modelo del ADC de Matlab/Simulink
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Función para obtener la INL en función de diferentes parámetros (Retardo
% del comparador (Tdelay), offset del comparador (V_offset) y offset
% dependiente de señal en el comparador (Depend_offset))

global Iref C1 C2 Tdelay V_offset Depend_offset Iin; % Variables globales para obtener los
% valores el WorkArea

Tdelay_vec=[0 2e-9 10e-9 100e-9 1000e-9 10000e-9 ]; % Vector de retrasos
NTdelay=length(Tdelay_vec);

V_offset_vec=[-0.0005 -0.0025 -0.0050 -0.0075 -0.0100 ]; % Vector de offset independiente de señal
NV_offset=length(V_offset_vec);

Depend_offset_vec=[0.002 0.004 0.006]; % Vector de offset dependiente de señal
NDepend_offset=length(Depend_offset_vec);

enob_Tdelay=zeros(1,NTdelay);
sندر_Tdelay=zeros(1,NTdelay);

for I=1:NTdelay
    for J=1:NV_offset % Llamada a la función test_bench_Par_INL para cada valor de retraso
        for K=1:NDepend_offset % para cada valor de offset, y para cada valor de offset dependiente
            [enob_Tdelay(I),sندر_Tdelay(I)]=test_bench_Par_ENOB(125e-9,5e-12,5e-12,Tdelay_vec(I),V_offset_vec(J),Depend_offset_vec(K),30);
        end
    end
end

```

Archivo: test_bench_Par_ENOB_T_Vo_Do_mstch

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Proyecto Fin de Carrera                                     %
%                                                            %
% Autor: Fco Javier Segovia de la Torre                     %
%                                                            %
% Nombre del proyecto: Convertidor Analógico digital basado en tiempo %
%                                                            %
% Scripts: Análisis paramétrico para analizar la viabilidad de la arquitectura %
%                                                            %
% Descripción: Función para obtener la ENOB haciendo un análisis parametrizando %
% el retraso del comparador , el offset del comparador, del offset dependiente %
% y del mismatch capacidades de la señal en el comparador del modelo del ADC %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Función para obtener la ENOB en función de diferentes parámetros (Retardo
% del comparador (Tdelay), offset del comparador (V_offset) y offset
% dependiente de señal en el comparador (Depend_offset))

global Iref C1 C2 Tdelay V_offset Depend_offset Iin;          % Variables globales para obtener los
                                                                % valores del WorkArea

Tdelay_vec=[0 2e-9 10e-9 100e-9 1000e-9];                    % Vector de retrasos
NTdelay=length(Tdelay_vec);

C1_vec=[4.75e-12 4.85e-12 4.95e-12 5.05e-12 5.15e-12 5.25e-12]; % Capacidades. Parametrización del mismatch
NC1=length(C1_vec);

V_offset_vec=[-0.0005 -0.0025];                               % Vector de offset del comparador
NV_offset=length(V_offset_vec);
Depend_offset_vec=[0.002 0.004 0.006];                       % Vector de offset dependiente en el comparador
NDepend_offset=length(Depend_offset_vec);

INL_Tdelay=zeros(1,NTdelay);

for I=1:NTdelay
    for L=1:NC1
        for J=1:NV_offset
            for K=1:NDepend_offset
                [enob,Tdelay(I),sdr,Tdelay(I)]=test_bench_Par_INL(125e-9,C1_vec(L),5e-12,Tdelay_vec(I),V_offset_vec(J),Depend_offset_vec(K),30);
            end
        end
    end
end

```