
Table of Contents

Practica 2 - Rafael Jose Martin Pelaez	1
Filtrado	3
Distorsion	5
Ecualizador	5
Pulso y fase con distorsion	6
Señal pequeña	7
Señal Amplificada	8
Señal con gran amplitud	9
IP3	11

Practica 2 - Rafael Jose Martin Pelaez

```
[Xp,t,Ts,Fs]=pulso();

d=20;
a=10.^(0.6);
y=atade(Xp,a,d);    %genero el pulso retardado y atenuado

figure(1);
plot(t,Xp);          %el pulso original
hold on;
plot(t,y);           %el pulso retardado y atenuado
legend('Pulso Original','Pulso Ret. y Ate.');
```



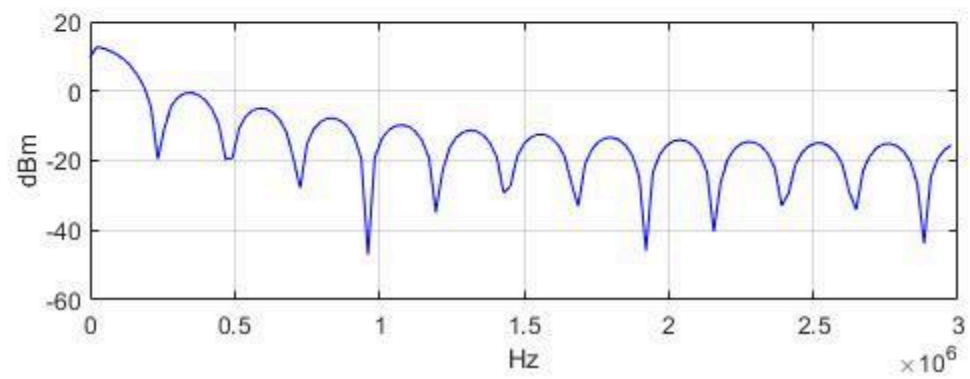
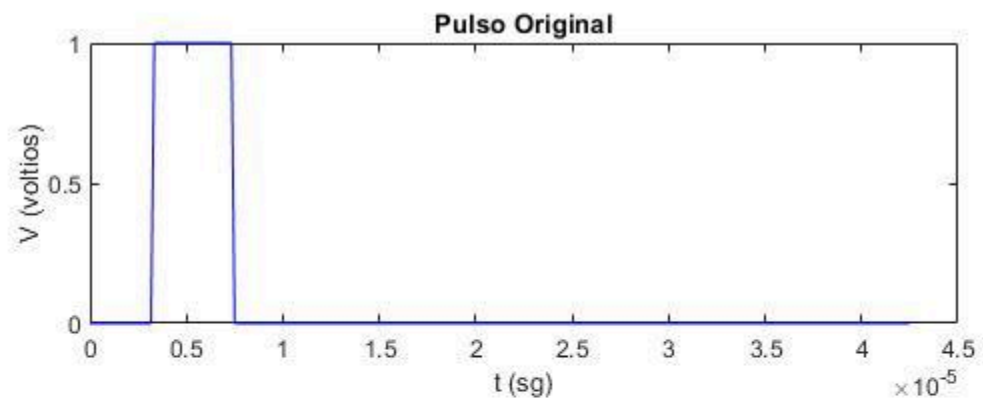
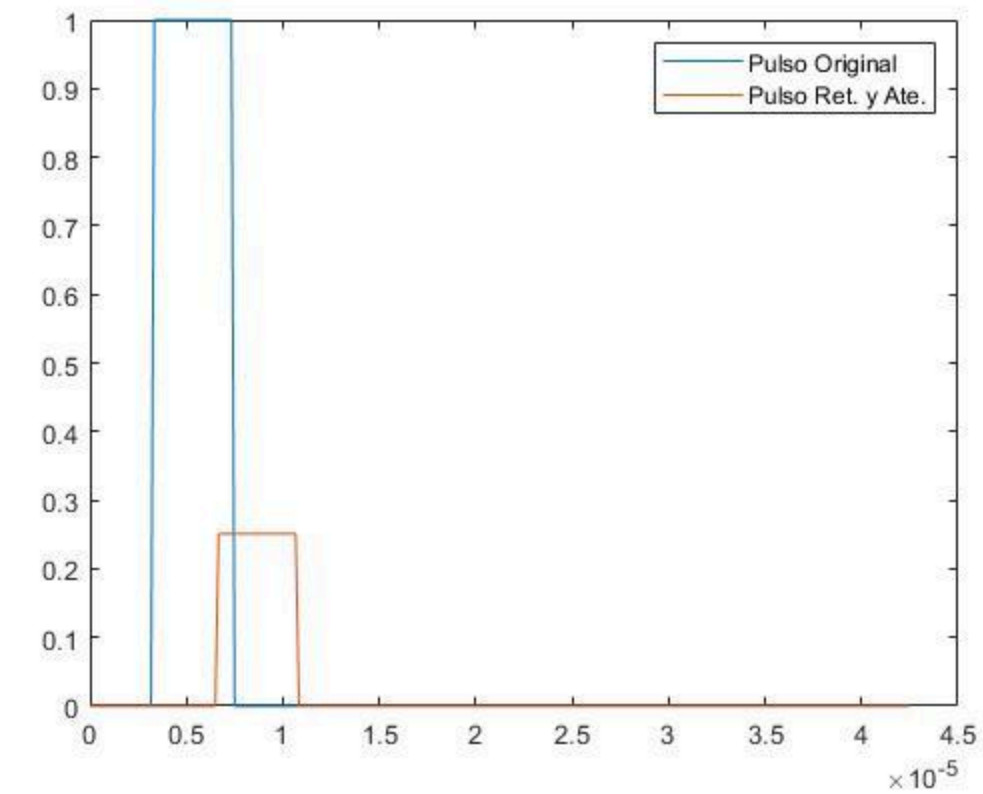
```
lin_log='log';

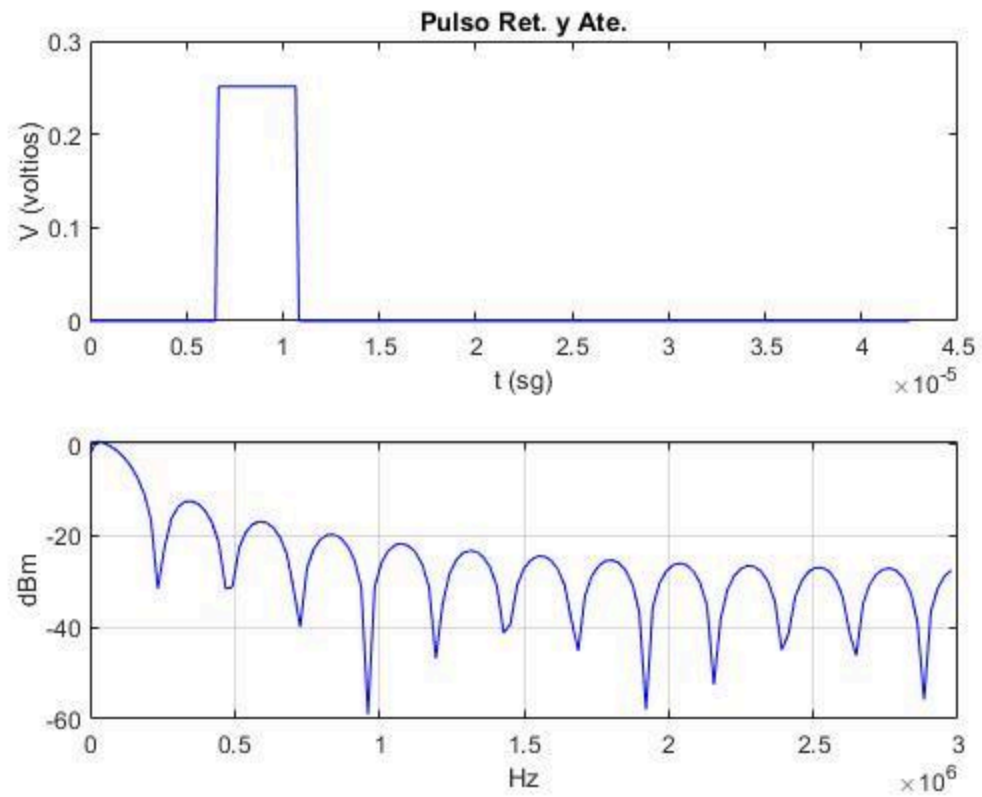
figure(2);
s='Pulso Original';
time_dep(Xp,Ts,1,s,lin_log);

figure(3);
s='Pulso Ret. y Ate.';
time_dep(y,Ts,1,s,lin_log)
```



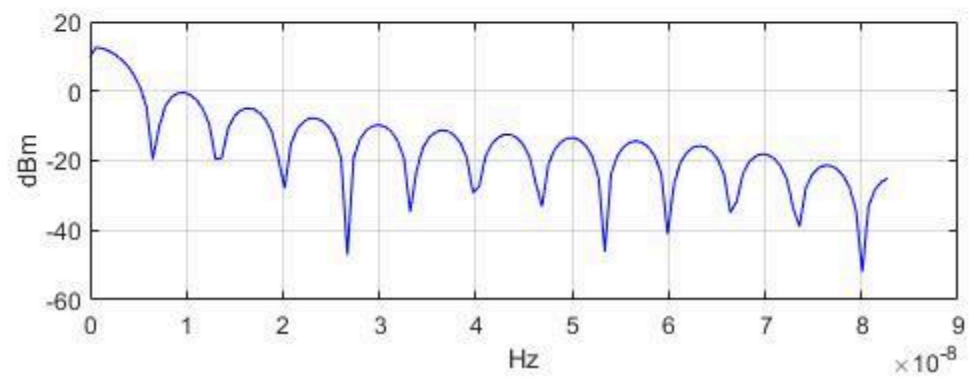
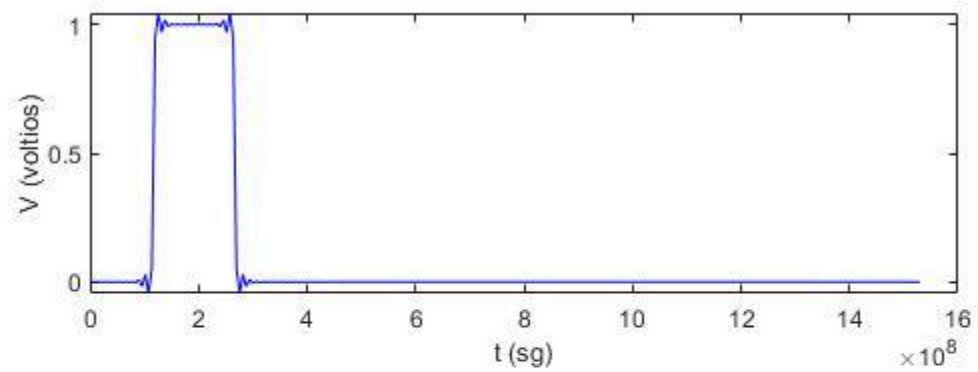
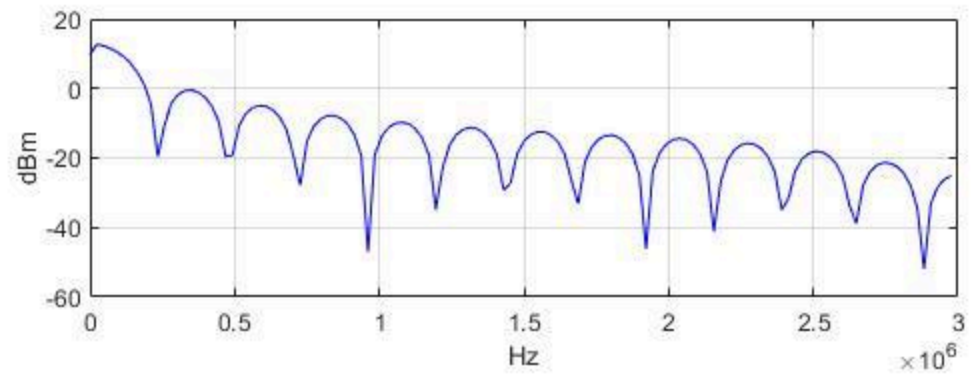
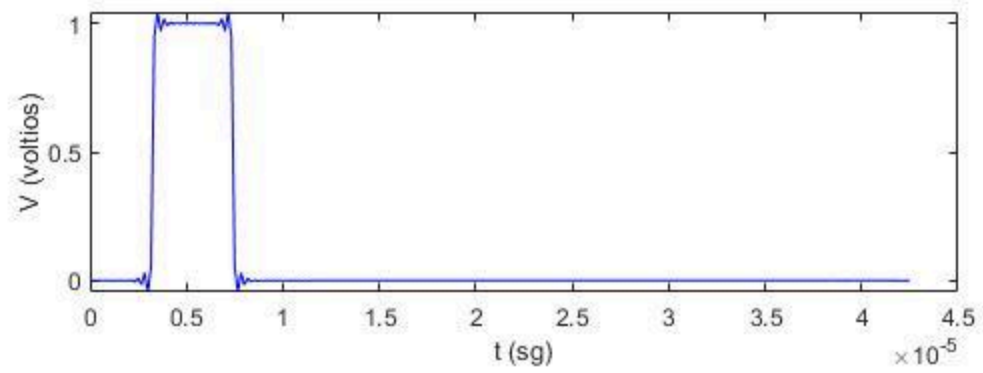
```
%Si se comporaran ambas en frecuencia, y vemos en el eje Y con cuantos
%decibelios empieza cada una, se ve claramente que una empieza con
menos
%potencia que la otra (6dB)
```





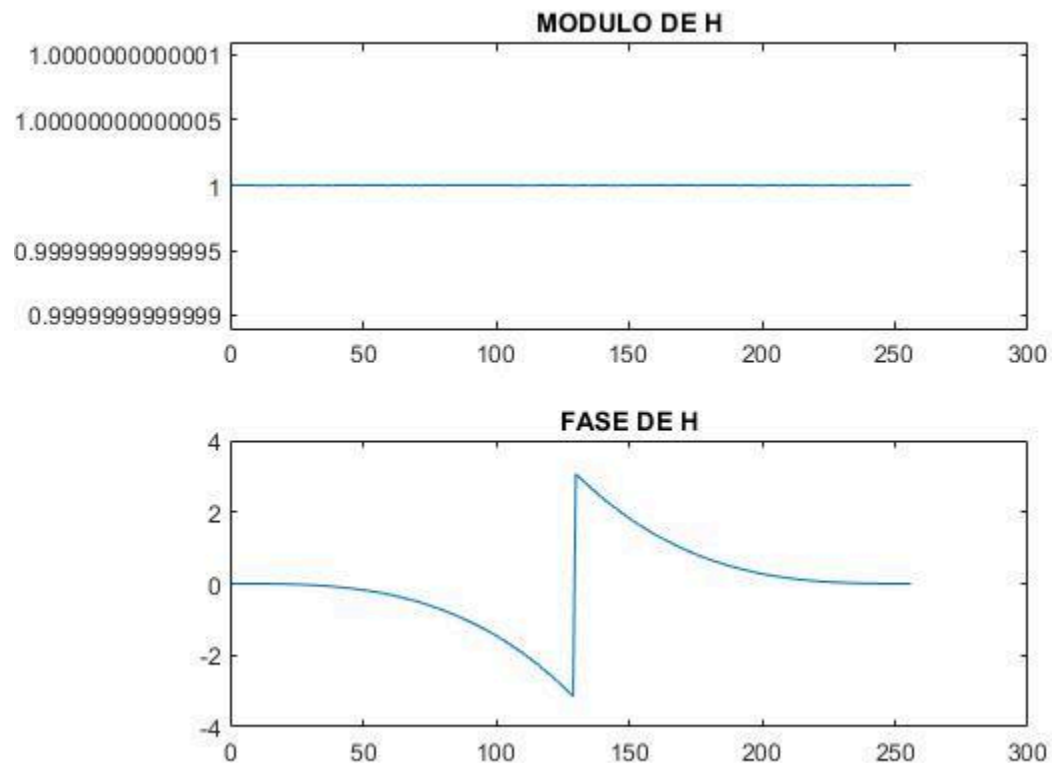
Filtrado

```
fc=2.5*(10.^6);  
n=6;  
fil=lpf(Xp, Fs, fc, n);  
  
s='';  
lin_log='log';  
  
figure(4)  
time_dep(fil,Ts,1,s,lin_log);  
  
figure(5)  
time_dep(fil,Fs,1,s,lin_log);
```



Distorsion

```
x_in=lpf(Xp,Fs,fc,n);  
H=disfase(length(x_in),3); % Transferencia del filtro  
H_mod=abs(H); % Módulo del filtro  
H_ang=angle(H); % Fase del filtro  
  
figure(6)  
subplot(2,1,1);  
plot(H_mod),title('MODULO DE H')  
subplot(2,1,2)  
plot(H_ang),title('FASE DE H')  
  
X_in=fft(x_in); % Espectro de la señal a la entrada  
X_out=X_in.*H; % La señal pasa por el filtro  
x_out=real(ifft(X_out,length(X_out))); % Señal temporal
```



Ecualizador

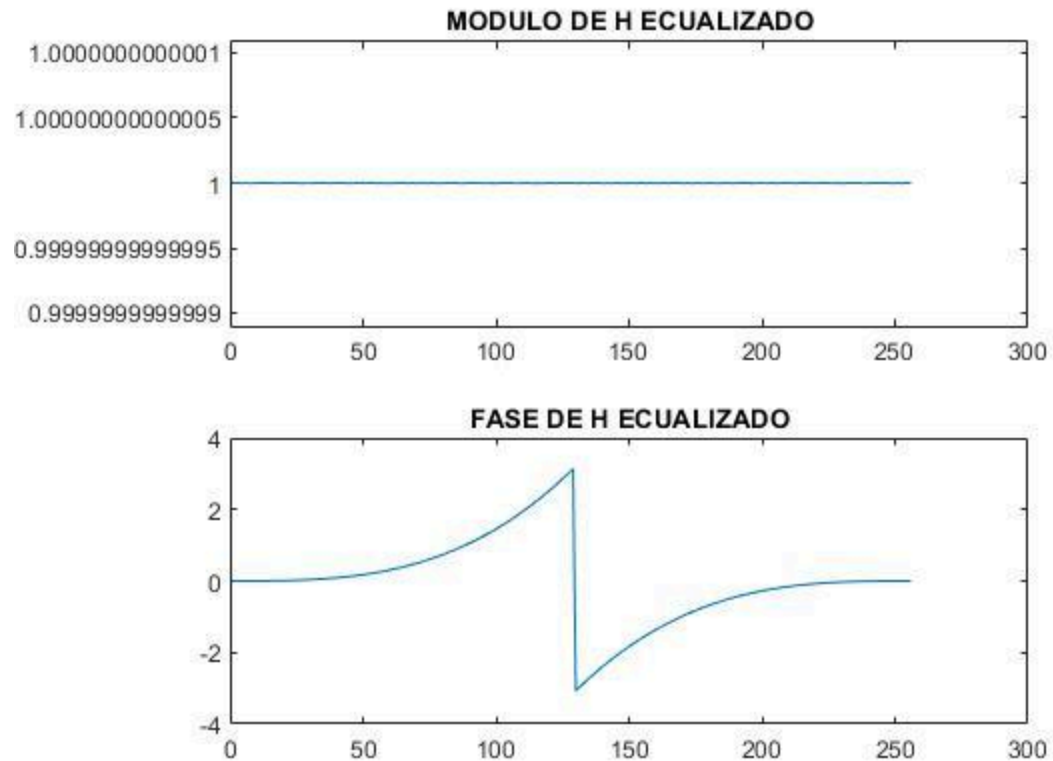
```
x_in = lpf(Xp,Fs,fc,n);  
H = disfase(length(x_in),3); % Transferencia del filtro  
Heq = H.^(-1);  
  
figure(7)  
H_mod = abs(Heq); % Módulo del filtro
```

```

H_ang = angle(Heq); % Fase del filtro
subplot(2,1,1);
plot(H_mod), title('MODULO DE H ECUALIZADO')
subplot(2,1,2)
plot(H_ang), title('FASE DE H ECUALIZADO')

X_in = fft(x_in); % Espectro de la señal a la entrada
X_out = X_in.*H; % La señal pasa por el filtro
x_out = real(ifft(X_out,length(X_out))); % Señal temporal

```

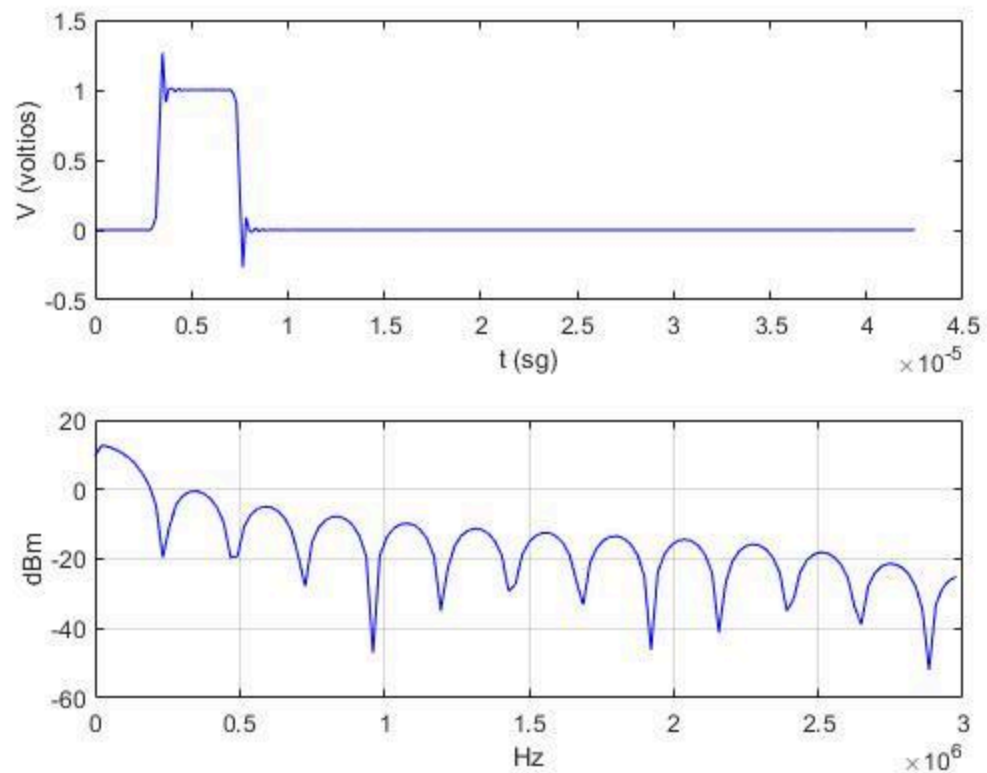


Pulso y fase con distorsion

```

figure(8)
time_dep(x_out,Ts,1,s,lin_log)

```



Señal pequeña

```

Fs=512;           %Frecuencia de muestreo
Ts=(1/Fs);       %Período de muestreo
N = 256;         %Numero de muestras
Freq1 = 30;      %Frecuencia de 1 tono
Freq2 = 34;      % Frecuencia segundo tono
Z = 50;          %Impedancia
t = 0:Ts:0.5;
f = cos((2*pi*Freq1)*t)/100;

```

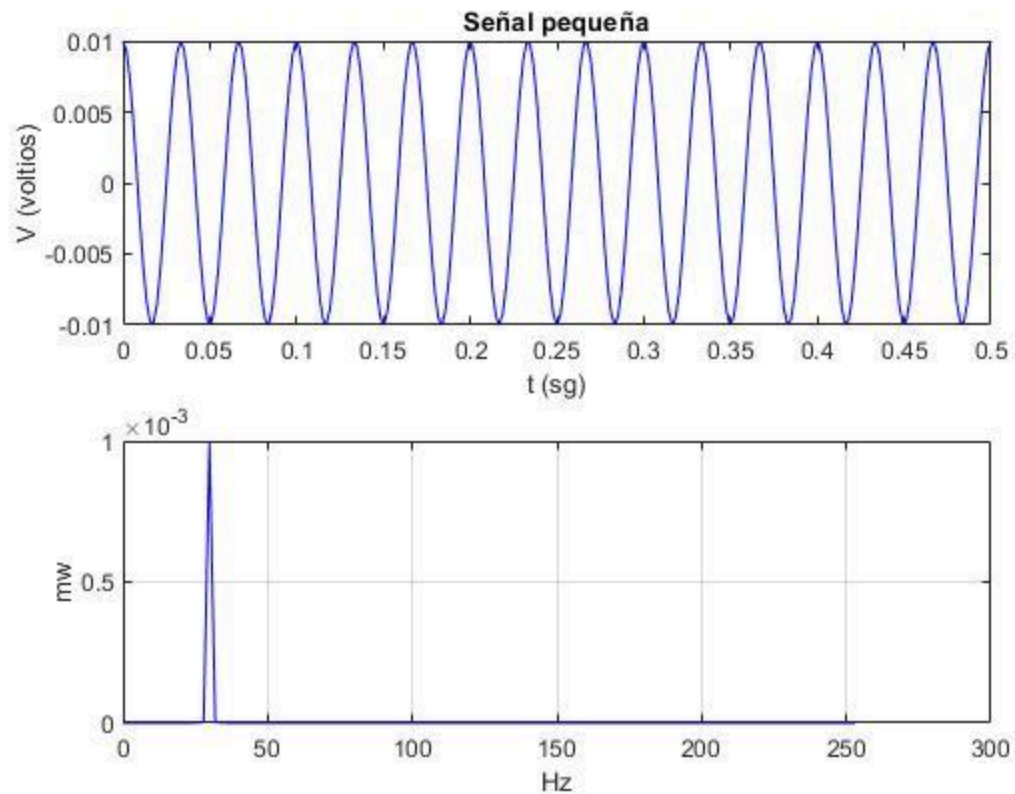
```

s = 'Señal pequeña';
lin_log = 'lin';
figure(9)
time_dep(f,Ts,Z,s,lin_log)

```

%vemos que tiene una amplitud de 0.01 voltios y en frecuencia el pico
%aparece en 30 Hz

*Warning: Integer operands are required for colon operator when used as
index.*

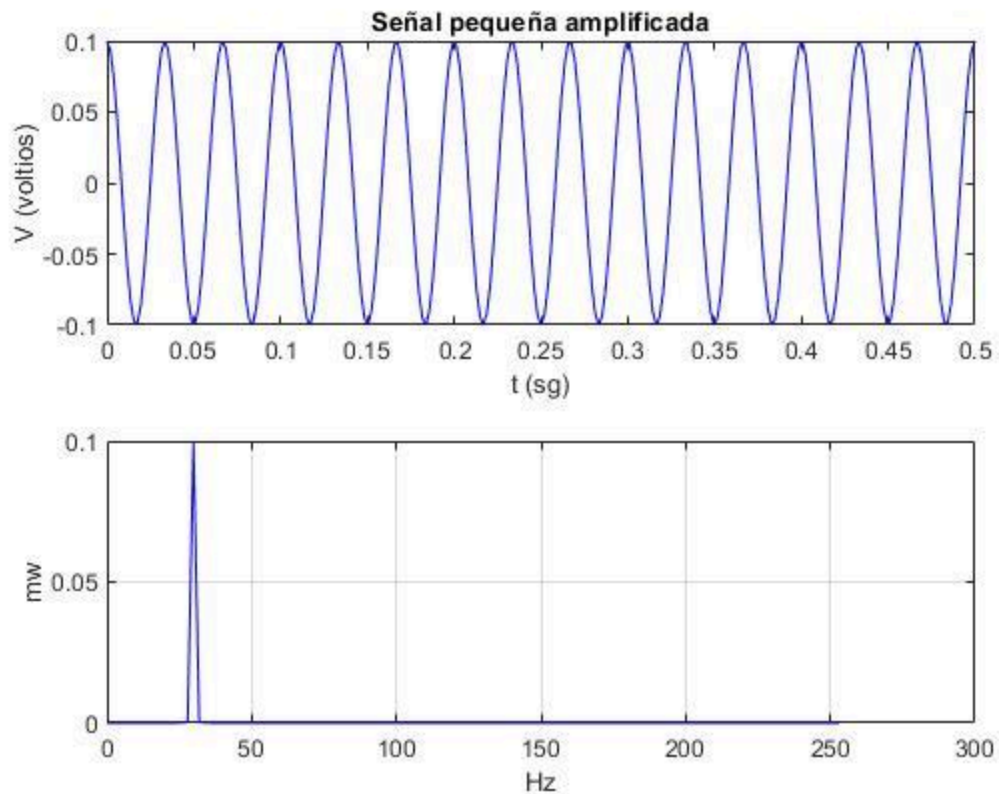


Señal Amplificada

```
a1 = 10;  
a2 = 0;  
a3 = -0.1;  
f = cos((2*pi*Freq1)*t)/100;  
ampli = a1*f + a2*f.^2 + a3*f.^3;  
figure(10)  
s = 'Señal pequeña amplificada';  
lin_log = 'lin';  
time_dep(ampli,Ts,Z,s,lin_log)
```

%La salida sale amplificada el valor de pico ha subido de 0.01 a 0.1,

Warning: Integer operands are required for colon operator when used as index.



Señal con gran amplitud

```

Fs=512;           %Frecuencia de muestreo
Ts=(1/Fs);        %Período de muestreo
N = 256;          %Numero de muestras
Freq1 = 30;       %Frecuencia de 1 tono
Freq2 = 34;       % Frecuencia segundo tono
Z = 50;           %Impedancia
t = 0:Ts:0.5;

```

```

x = 7*cos(2*pi*Freq1*t);
figure(11)
s = 'Señal grande';
lin_log = 'lin';
time_dep(x,Ts,Z,s,lin_log);
a1 = 10;
a2 = 0;
a3 = -0.1;
c = a1*x+ a2*x.^2 + a3*x.^3;
s1 = 'Señal grande amplificada';
figure(12)
time_dep(c,Ts,Z,s1,lin_log)

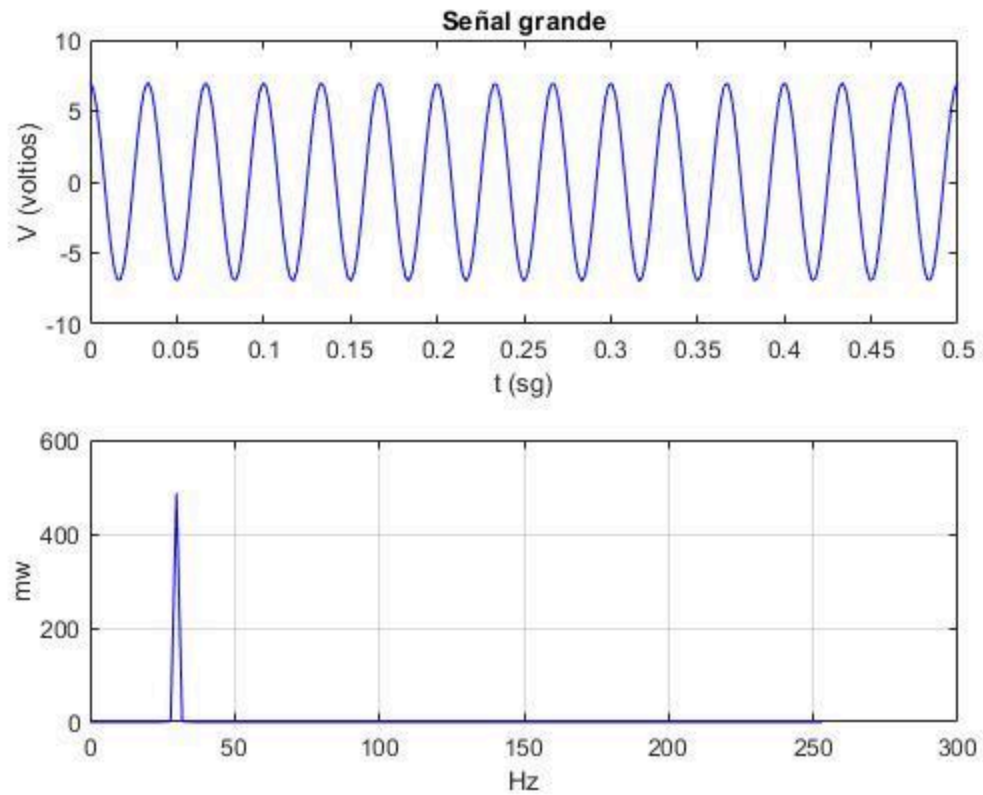
```

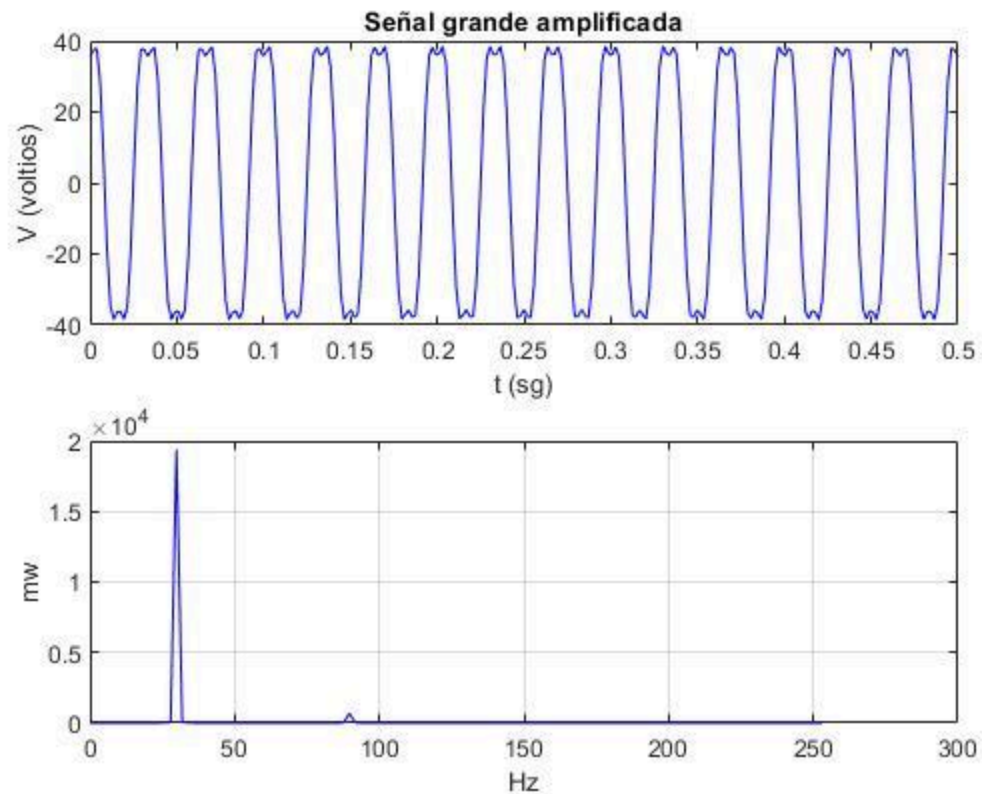
%ahora si que somos capaces de apreciar una fuerte distorsion en los

%valores de la amplitud andemas de que en frecuencia se hace
apreciable un
%pico de 90 Hz

Warning: Integer operands are required for colon operator when used as index.

Warning: Integer operands are required for colon operator when used as index.



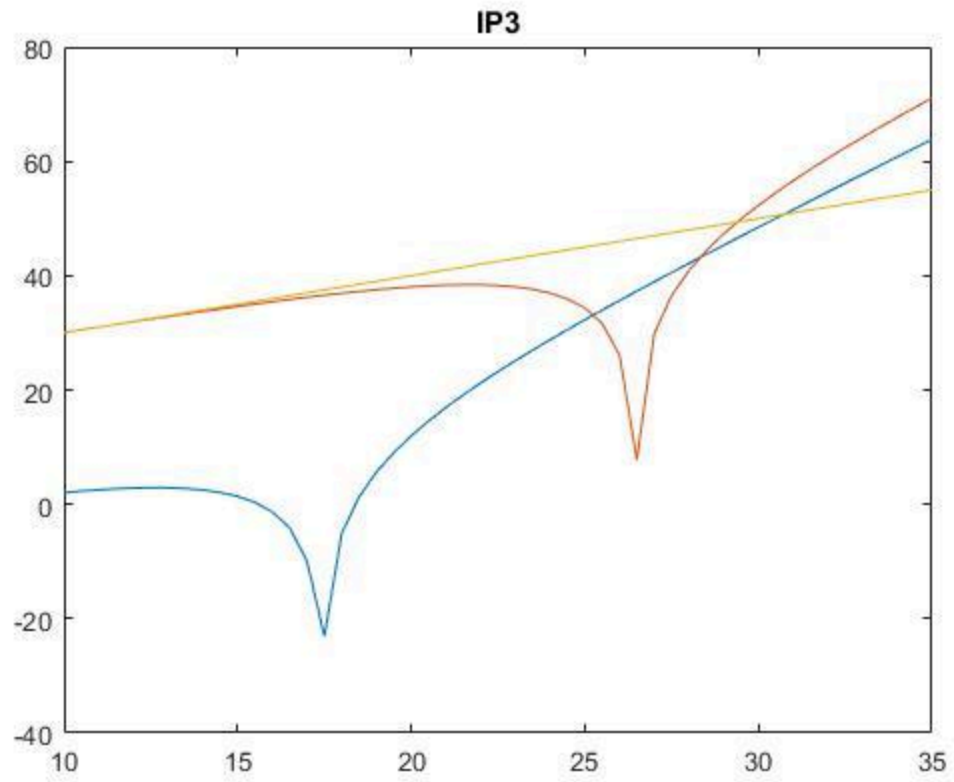


IP3

```

a1 = 10;
a2 = 0;
a3 = -0.1;
Gi = 20*log10(a1); % ganancia ideal, de pequeña señal, sin saturación
Pin = 10:0.5:35; % pot. de entrada (dBm) de 1 tono; 51 elementos
Pout_i = Pin + Gi; % pot. de salida sin saturación (la calculo aquí)
for i = 1:length(Pin) % para recorrer los 51 valores
    A = sqrt(10.^(Pin(i)/10)*0.002*50); % calculo A para cada pot.
    x1 = A*cos(Freq1*2*pi*t); % frecuencia f1, amplitud A
    x2 = A*cos(Freq2*2*pi*t); % frecuencia f2, amplitud A
    x = x1 + x2; % suma de los 2 tonos
    y = a1*x + a2*x.^2 + a3*x.^3; % pasa por el polinomio
    Pout_3(i) = 10*log10(powmeter(y,Fs,2*Freq1-Freq2,1,Z))+30; % batido
    Pout_f(i) = 10*log10(powmeter(y,Fs,Freq1,1,Z))+30; % fundamental con
    sat.
end
figure(14)
plot(Pin,Pout_3,Pin,Pout_f,Pin,Pout_i)
title('IP3');

```



Published with MATLAB® R2018b