

Examen tercer parcial

Tinoco, Sergio.

Instituto Politécnico Nacional, Escuela Superior de Cómputo.

Procesamiento de Señales

20 de enero de 2023

En el presente examen se va a desarrollar una aplicación de análisis de señales por medio de App Designer en Matlab.

Palabras clave: Fourier, filtro, muestreo.

DESARROLLO

Se genera la interfaz con los componentes necesario para el análisis espectral.

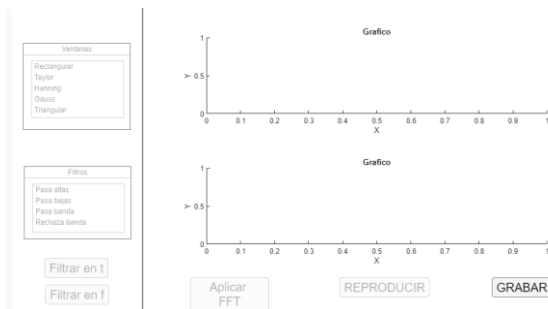


Figura 1. GUI.

Luego se programan cada uno de los componentes a utilizar.

```
function GRABARButtonPushed(app, event)
    senal_salida=audiorecorder(app.fs,16,1); %Creacion del objeto de grabacion
    msgbox('Empezando Grabacion','Grabadora');%Mensaje de informacion
    recordblocking(senal_salida,app.segs); %Grabacion del sonido
    msgbox('Terminando grabación');%Mensaje de informacion
    %Paso los valores del objeto a una señal
    app.x=getaudiodata(senal_salida,'single');
    %Se graba y se guarda la señal
    audiowrite("audio.wav", app.x,app.fs);
```

Figura 2. Se programa el botón de grabar audio.

```
function REPRODUCIRButtonPushed(app, event)
    [y,Fs0] = audioread("audio.wav"); %Abre el archivo de la grabacion
    %Reproducir audio
    sound(app.x,Fs0) %Reproduce el audio
```

Figura 3. Se programa el botón de reproducir audio.

```
switch value
    case 'Rectangular'
        w=rectwin(n);

    case 'Taylor'
        w=taylorwin(n);

    case 'Hanning'
        w=hann(n);

    case 'Gauss'
        w=gausswin(n);

    case 'Triangular'
        w=triang(n);
```

Figura 4. Aplicación de ventanas

```
function AplicarFFTButtonPushed(app, event)
    y=fft(app.x);
    n=(app.fs)*(app.segs);
    p1=2.*(abs(y(1:n/2))/n);
    plot(app.UIAxes_2,p1);
    app.UIAxes_2.Title.String="Señal Original";
    app.UIAxes_2.XLabel.String="Frecuencia";
    app.xf=p1;
```

Figura 5. Aplicar FFT

```
function ListBox_2ValueChanged(app, event)
    value = app.ListBox_2.Value;
    wn=0;
    b=0;
    switch value
        case 'Pasa altas'
            wn=500/(app.fs);
            b=fir1(30,wn,"high");
        case 'Pasa bajas'
            wn=1000/(app.fs);
            b=fir1(30,wn,"low");
        case 'Pasa banda'
            wn=[500/(app.fs) 2000/(app.fs)];
            b=fir1(30,wn,"bandpass");
        case 'Rechaza banda'
            wn=[500/(app.fs) 2000/(app.fs)];
            b=fir1(30,wn,"stop");
    otherwise
        %
```

Figura 6. Generación de filtros.

```
%Filtrar en el dominio del tiempo
app.sft=filter(b,1,app.x);
plot(app.UIAxes, app.sft);

%Filtrar en el dominio de la frecuencia
[h,w]=freqz(b,1); %Aplicar respuesta de freq
app.w1=w;
app.sff=abs(h);
app.FiltrarenfButton.Enable="on";
app.FiltrarenfButton.Enable="on";
```

Figura 7. Aplicación de filtro.

```
function FiltrarenfButtonPushed(app, event)
plot(app.UIAxes,app.xf)
app.UIAxes.Title.String="Señal Original";
app.UIAxes.XLabel.String="Frecuencia";

plot(app.UIAxes_2,(app.w1*app.fs)/pi,app.sff);
app.UIAxes_2.Title.String="Señal Filtrada";
app.UIAxes_2.XLabel.String="Frecuencia";
```

Figura 8. Graficar comparación con filtros.

RESULTADOS

Se genera una señal de audio con una duración de 0.5 segundos.

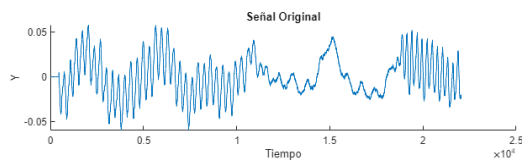


Figura 9. Señal de Audio.

Se aplica una ventana de Taylor.

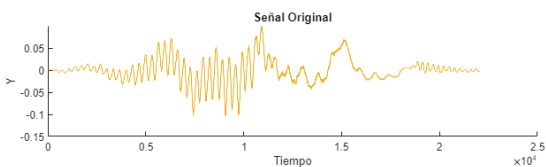


Figura 10. Ventaneo de la señal.

Se aplica la transformada rápida de Fourier.

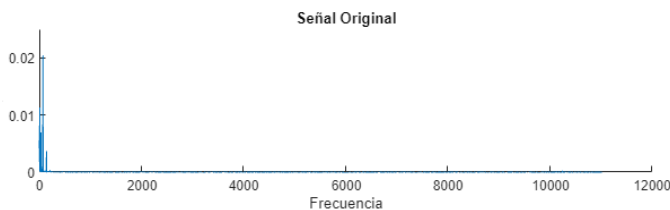


Figura 11. FFT

Se aplica un filtro de pasa baja.

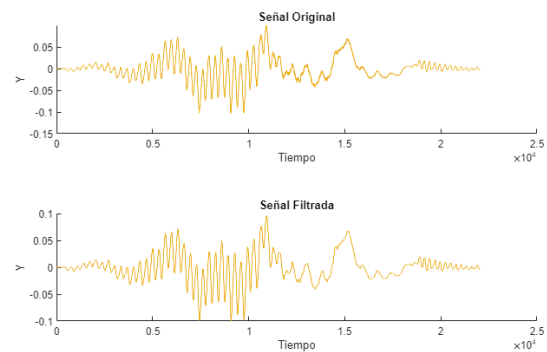


Figura 12. Señal con filtro de pasa baja en el dominio del tiempo.

Se aplica un filtro de pasa baja.

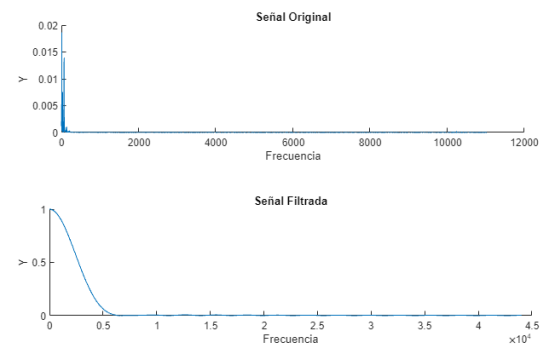


Figura 13. Señal con filtro de pasa baja en el dominio de frecuencia.