



DEPARTAMENTO DE ELECTRÓNICA Y AUTOMÁTICA
FACULTAD DE INGENIERÍA – UNIVERSIDAD NACIONAL DE SAN JUAN

Informe de Practica de DSP N°1
Procesamiento Digital de Señales usando Programación en C

Asignatura: Procesamiento Digital de Señales
Ingeniería Electrónica

Autor:
Avila, Juan Agustin – Registro 26076

1º Semestre
Año 2020

1 Enunciado

1. Realizar un programa en lenguaje C que contenga las funciones necesarias para abrir un archivo de datos, procesarlo y grabar en otro archivo el resultado del procesamiento. Usar cualquier compilador de C.
2. Los archivos de datos para trabajar se encuentran en la siguiente carpeta en el sitio web de la asignatura: Otros Recursos/Señales/Práctico DSP N° 1
3. Los archivos contienen datos de números reales, están en formato de texto, .txt y el primer dato del arreglo es la frecuencia de muestreo de la señal en [Hz]. La unidad de la señal discreta es Volts [V].
4. Para todas las señales guardadas en archivos, programar un algoritmo para calcular el valor medio y restarlo de la señal original, guardando la señal resultante en un archivo diferente.
5. Realizar la suma y la diferencia de dos señales cualesquiera y guardar el resultado en archivo archivos.
6. Para todas las señales, calcular la frecuencia y los valores máximo y mínimo.
7. Realizar la autocorrelación de una señal cualquiera y la correlación entre dos señales.
8. Realizar un informe con los resultados obtenidos, incluyendo gráficas y los programas utilizados. Utilizar el programa Matlab u otro similar para graficar las señales originales y resultados de los procesamientos.

2 Resolución

Se realizó el programa en C (DSP1.c), utilizando como editor “Visual Studio Code” y como compilador “MinGW”. No se reproduce el código del programa en este informe por su extensión, por lo tanto se envía como archivo adjunto.

2.1 Valores medidos en todas las señales.

Los resultados del programa fueron los siguientes:

```
Signal 1
v1medio= 1.998336
MIN: 1.0V      MAX: 3.0V      FREQ ANALOG: 4.0Hz

Signal 2
v2medio= 1.500000
MIN: -0.5V     MAX: 3.5V     FREQ ANALOG: 5.0Hz

Signal 3
v3medio= 0.740741
MIN: 0.0V      MAX: 1.5V      FREQ ANALOG: 20.0Hz

Signal 4
v4medio= -0.987654
MIN: -2.0V     MAX: 0.0V     FREQ ANALOG: 1.0Hz

Signal 5
v5medio= 3.027944
MIN: 1.0V      MAX: 5.0V     FREQ ANALOG: 29.4Hz
```

Una vez ejecutado el programa en C, se comprueban los resultados utilizando MatLab con el siguiente script:

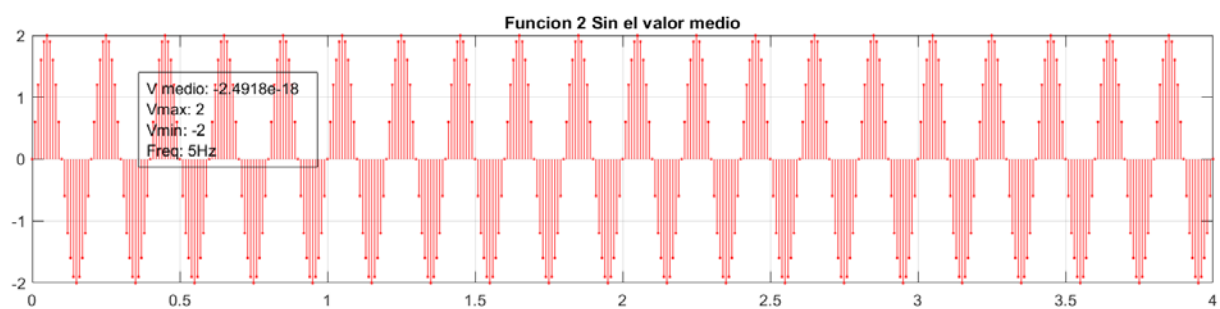
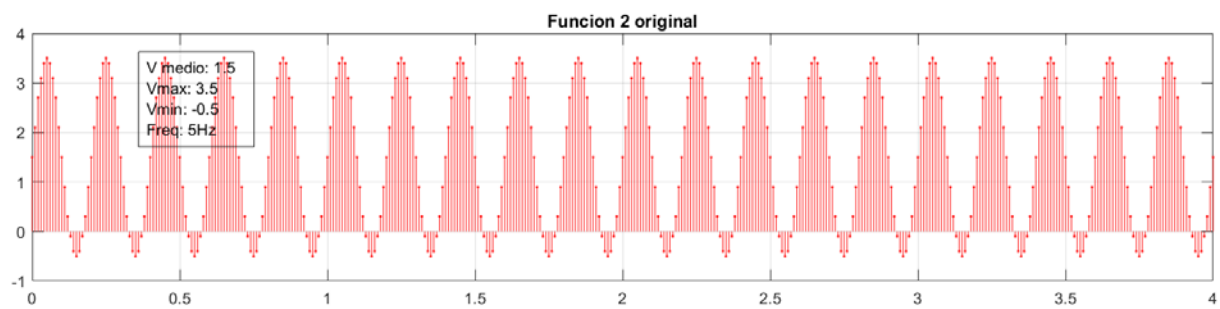
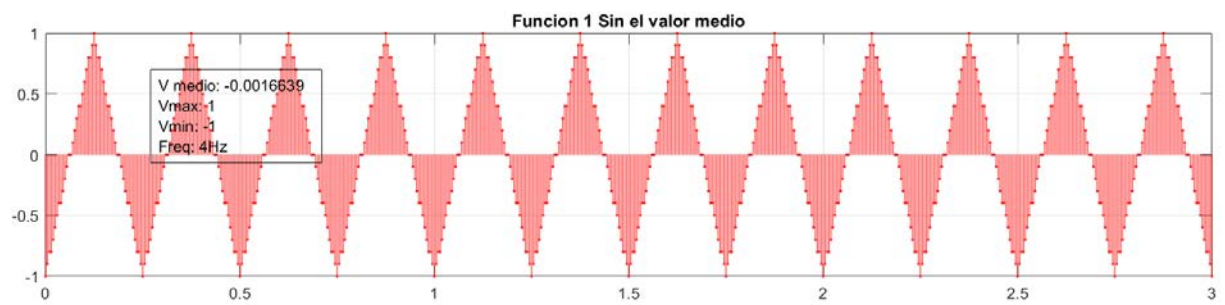
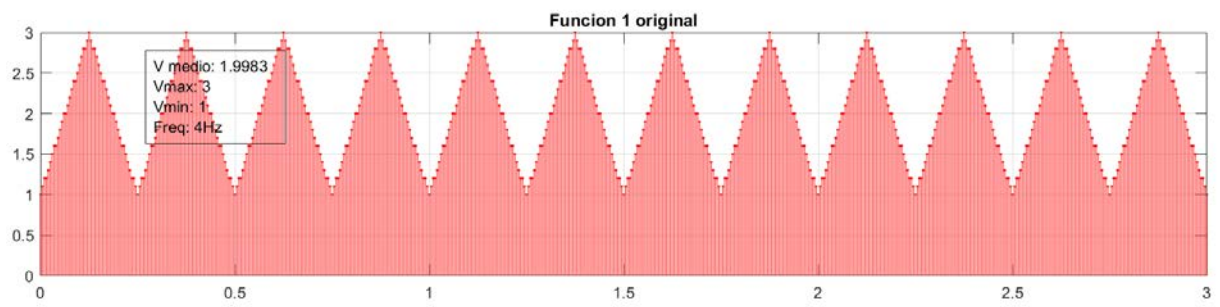
```
%% graficacion de funciones originales y sin valores medios:
for i=1:5 %grafica las 5 funciones
figure("Name","Funcion "+i); %genera la figura
subplot(211); % en la figura superior grafica la
original
graficacionfunciones(i,"","original"); %funcion que grafica
dim=[0.2 0.6 0.3 0.3]; % estas dimensiones son para agregar
un cuadro con datos
valores(i,"",dim); %la funcion valores calcula
max,min,medio y freq
subplot(212); %repite el proceso para la funcion
sin Vmedio
graficacionfunciones(i,"Vmedio","Sin el valor medio");
dim=[0.2 0.1 0.3 0.3];
valores(i,"Vmedio",dim);
end
```

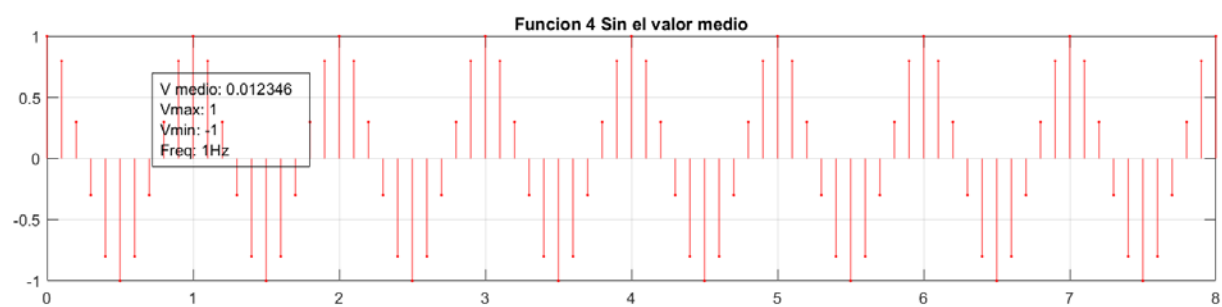
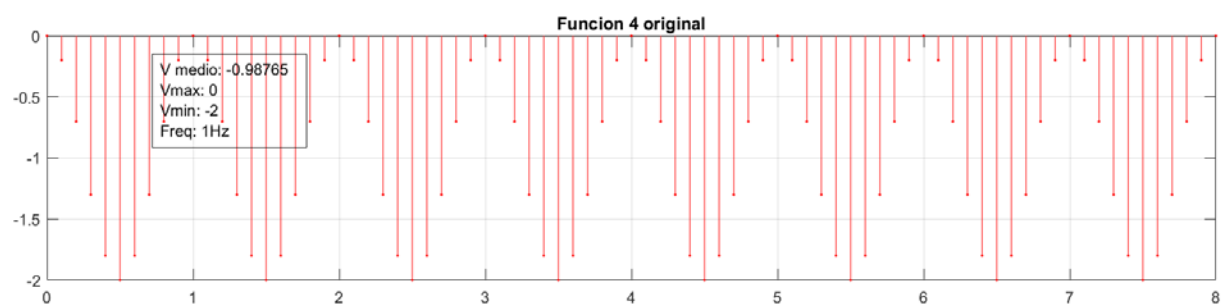
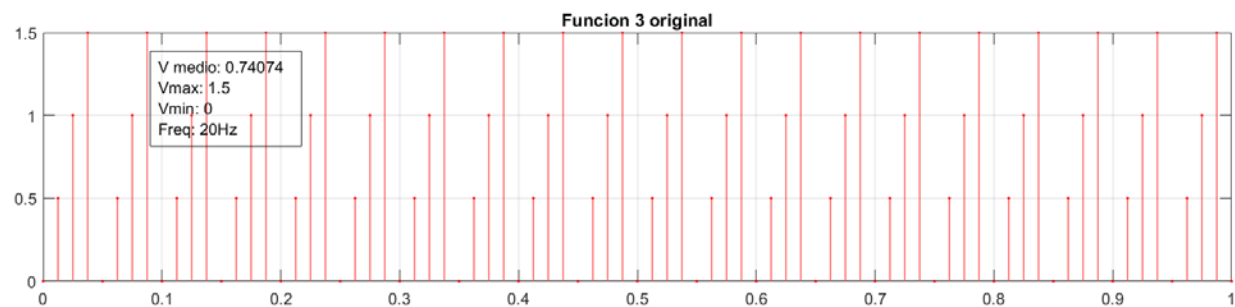
Ademas, se utilizan dos funciones complementarias:

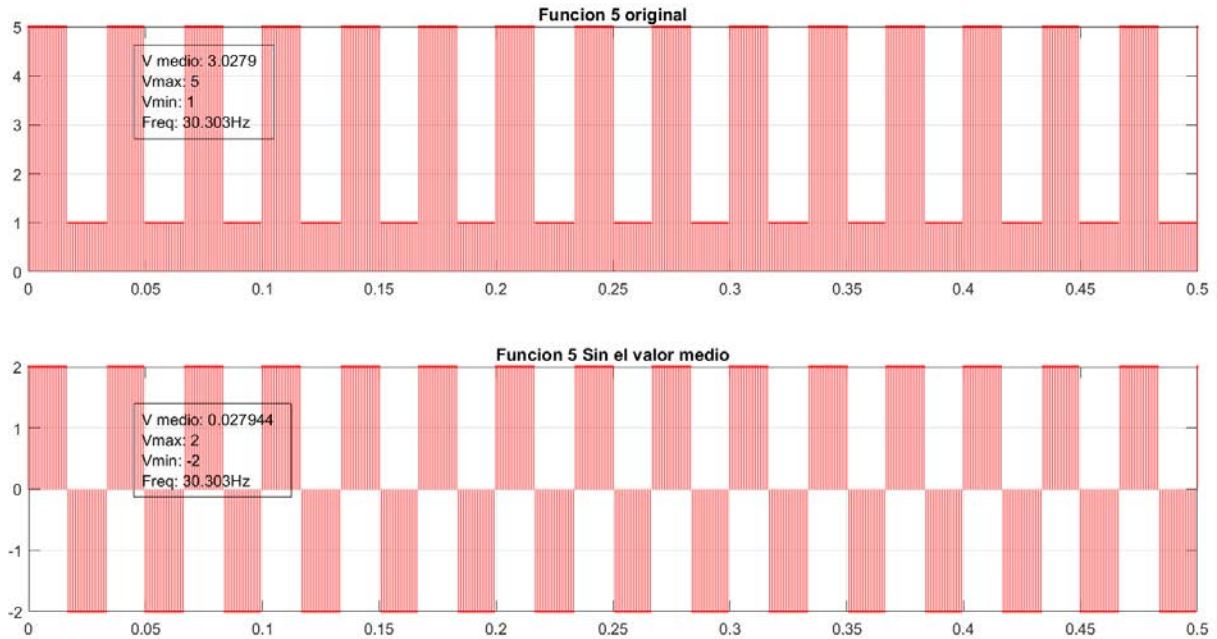
```
%% funciones utilizadas
function tiempo=graficacionfunciones(i,mod,tipo) %%abre el archivo y
lo grafica
arch=load("Signal_0"+i+mod+".txt");
%abre el archivo "i" y mod es para agregar si esta correlacionado,
%es la señal sumada, restada o sin valor medio.
Fs=arch(1); %guarda Fs
n=0:1/Fs:(length(arch)-2)/Fs; %Genera base temporal en base a Fs y
la longitud
dtplot(n,arch(2:length(arch))); %Grafica la funcion
title("Funcion "+i+" "+tipo); %Agrega titulo
grid;
tiempo=(length(arch)-1)/Fs; %Devuelve el tiempo(Util para que
todas tengan la misma longitud)
end

function valores(i,mod,dim) %analiza varios valores
arch=load("Signal_0"+i+mod+".txt"); %abre el archivo, idem anterior
Fs=arch(1) %Idem anterior
Vmedio=sum(arch(2:length(arch)))/(length(arch)-1); %Calcula el
promedio
maximo=max(arch(2:length(arch))); %Obtiene maximo
minimo=min(arch(2:length(arch))); %Obtiene minimo
[m1, n1]=max(arch(2:length(arch-1))); %Se obtienen los indices
de
[m2, n2]=min(arch(n1:length(arch-n1))); %distintos max y min, se
[m3, n3]=max(arch(n1+n2:length(arch-n1-n2))); %evita el primer periodo
ya
[m2, n4]=min(arch(n1+n2+n3:length(arch-n1-n2-n3))); %que en una
funcion
[m3, n5]=max(arch(n1+n2+n3+n4:length(arch-n1-n2-n3-n4))); %no era
igual
freq=Fs/(n4+n5); %Calcula la frecuencia
analogica
%Agrega un textbox al grafico
str = {"V medio: "+Vmedio,"Vmax: "+maximo,"Vmin: "+minimo,"Freq:
"+freq+"Hz"};
annotation('textbox',dim,'String',str,'FitBoxToText','on');
end
```

Se generaron las siguientes graficas:







Se observa que, al cargar los datos de las funciones a las que se les restó el valor medio, su valor medio no es exactamente cero. Esto se debe al redondeo de los valores flotantes cuando se guardó el nuevo archivo (un solo valor decimal). Esto se hace evidente en la graficación de las señales sin valor medio, ya que en algunas el valor absoluto del máximo y mínimo no coinciden. También hay una pequeña diferencia entre la frecuencia analógica calculada para la función 5 en matlab y en c. (29.4Hz vs 30.3Hz).

2.2 Suma y resta de dos funciones:

Se sumaron y restaron las señales 1 y 2. Para esto, fue necesario lograr que ambas señales sean similares (mismo tiempo y frecuencia de muestreo). Para ello, se tomó el menor de los tiempos (calculándolo con la cantidad de datos y la frecuencia de muestreo), y se eligió como frecuencia de muestreo la mayor de las dos. Al ser las frecuencias distintas, fue necesario realizar una interpolación lineal a la señal de menor frecuencia para los instantes en los que mostraba la señal de mayor frecuencia. Para comprobar los resultados obtenidos, se graficaron ambas operaciones en matlab utilizando los siguientes comandos:

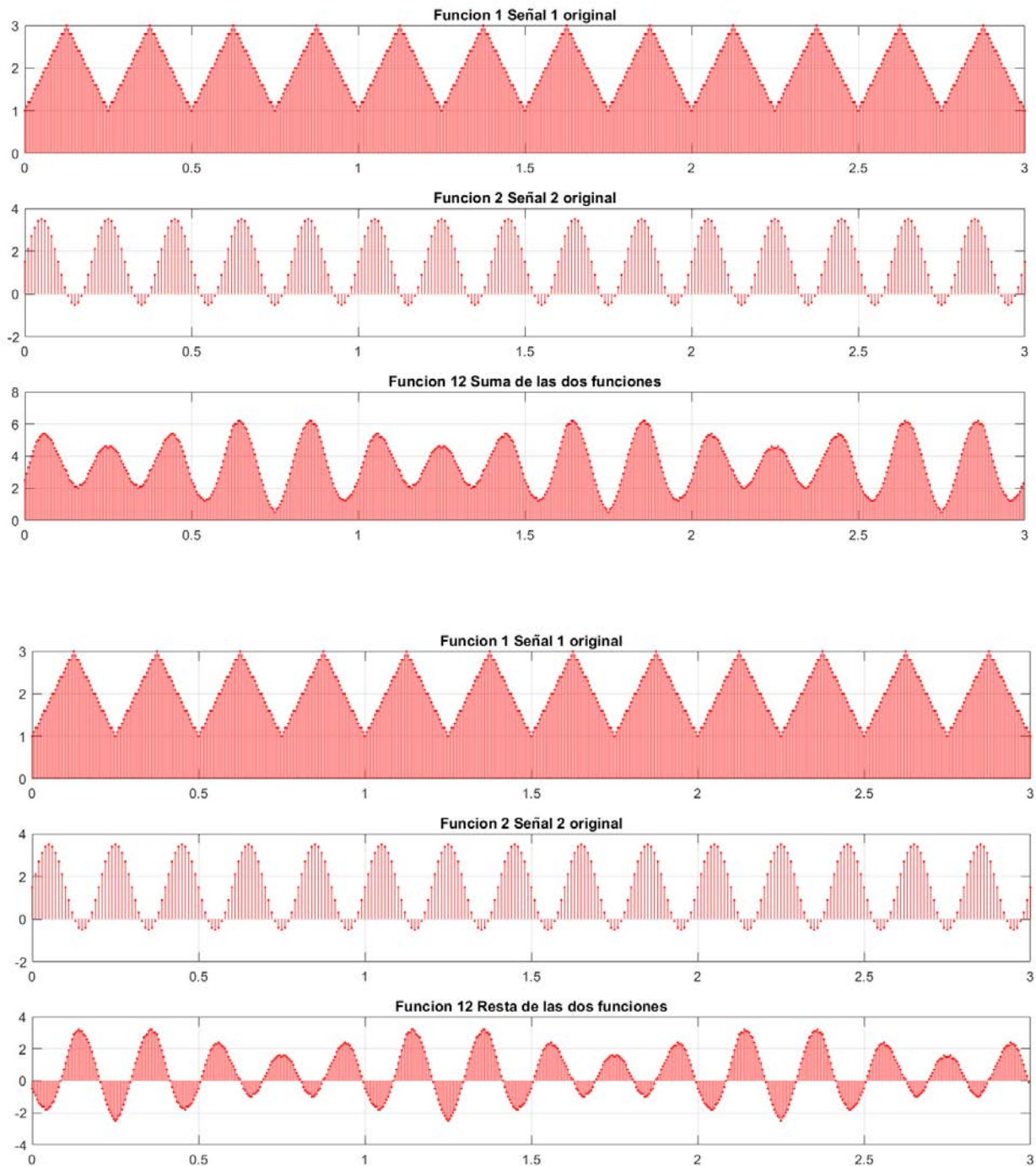
```
%% grafica de la suma y resta
s1=1;s2=2; %%funciones que se sumaron o restaron
sumayresta(s1,s2,"Suma");
sumayresta(s1,s2,"Resta");
```

Y la siguiente función auxiliar:

```
function sumayresta(s1,s2,tipo)    %grafica ambas señales y suma o
resta
figure();
subplot(311);
graficacionfunciones(s1,"","Señal "+s1+" original");
subplot(312);
graficacionfunciones(s2,"","Señal "+s2+" original");
subplot(313);
t=graficacionfunciones(s1*10+s2,tipo,tipo+" de las dos funciones");
%devuelve el tiempo de la señal sumada o restada para ajustar los
```

```
%ejes de las graficas originales y asegurarse que haya
%correspondencia en las graficas
subplot(311);xlim([0 t]);
subplot(312);xlim([0 t]);
end
```

Los resultados obtenidos son los siguientes:



En las gráficas se observa que la señal resultante está muestreada a la misma frecuencia que la señal de mayor frecuencia. A su vez, se ven pequeñas irregularidades en los valores de salida dadas por la interpolación lineal.

2.3 Correlación.

La correlación se realizó desde $k=0$ en vez desde $-\infty$. Se realizó la correlación entre la señal 1 y 2, y la autocorrelación de la señal 2. Para la resolución de este problema, se volvieron a generar señales similares, con la misma duración y frecuencia de muestreo. Se graficó la salida en matlab utilizando los siguientes comandos:

```
%% correlacion:
figure("Name","Correlacion de señales 1 y 2");
graficacionfunciones(12,"correlacion","correlacionada");
title("Correlacion de señales 1 y 2");
figure("Name","Autocorrelacion de señal 2");
graficacionfunciones(2,"correlacion","autocorrelacionada");
```

Y las salidas resultantes son las siguientes:

