

## ANEXO B

### B.1. Código En MATLAB y Arduino para el Tratamiento de las Señales y Obtención de Parámetros

#### B.1.0. Código en el entorno arduino

Se implementa un ciclo *for* con el fin de estimar el tiempo en que se realizan los muestreos por parte de la tarjeta Arduino, calculando la diferencia entre el tiempo en que se abre el puerto  $A_0$  y se cierra, en las variables *time1* y *time*, se tienen en cuenta pequeñas pérdidas de tiempo que se generan al guardar y calcular dichas variables. A continuación se muestra el código que se implementa.

```
void setup(){
Serial.begin(9600);
muestreo();
}
void loop(){ }
void muestreo(){
unsigned long time1=0;
unsigned long time=0;
for(byte i =0; i<5; i++){
time1=micros();
int A=analogRead(A0);
time=micros()-time1;
Serial.print(" Muestra: ");
Serial.print(i+1);
Serial.print(" Tiempo: ");
Serial.println(time); }
}
```

### B.1.1. Funciones de MATLAB Utilizadas en el Código para la Obtención de Parámetros de Identificación

En la tabla 1 se indican las funciones utilizadas en el entorno de MATLAB.

Cuadro 1: **Funciones Matlab. Fuente:[24]**

<b>Función</b>	<b>Descripción</b>	<b>Sintaxis</b>
<b>load</b>	Carga variables de un archivo en el espacio de trabajo.	<b>load('nombre_archivo')</b>
<b>thd</b>	Devuelve la distorsión armónica total, las potencias (en dB) y las frecuencias de los armónicos, incluida la fundamental, donde x indica los valores de la señal sinusoidal, fs es la frecuencia de muestreo y n el número de armónicos.	<b>[r,harpow,harmfreq] = thd(x,fs,n)</b>
<b>db2mag</b>	Devuelve las medidas de magnitud, que corresponden a los valores de decibelios (dB) especificados en ydb.	<b>db2mag(ydb)</b>
<b>rsm</b>	Devuelve la raíz cuadrada media (RMS) como un escalar. Siendo x un vector fila o columna.	<b>rms(x)</b>
<b>findpeaks</b>	Encuentra los picos máximos de una señal, especificado la frecuencia de muestreo deseada Fs, tomando como primera muestra el tiempo con valor cero.	<b>findpeaks(dato,Fs)</b>
<b>xlswrite</b>	Escribe un archivo en la hoja de cálculo y el intervalo especificado en Microsoft Excel, donde filename es el nombre del archivo, A contiene los datos de entrada, sheet número de la hoja en Excel y xlRange es el rango donde se quieren escribir los datos	<b>xlswrite(filename,A,sheet,xlRange)</b>

### CÓDIGO

Para iniciar, se abre el entorno de MATLAB, se cargan los datos del *.txt* dado por el software CASSY lab 2, y se guardan los datos en variables. en donde se desarrollo el siguiente código:

```

% CARGAR DATOS DEL .TXT
load ('TRES_10.txt')
T = TRES_10(:,1)*(10^-3);
Vo = TRES_10(:,3) ;
Cu = TRES_10(:,2);
CosPhi=TRES_10(1,4);

```

Después, se calcula la distorsión armónica total por medio de la función *thd*, dicho valor es dado en *dB* entonces para facilitar los cálculos, con la función *db2mag* obtenemos el valor en magnitud.

```

% DISTORSIÓN ARMONICA TOTAL
[THD_Cu, PotHarm_Cu, FreqHarm_Cu] = thd (Cu,60,15);
[THD_Vo, PotHarm_Vo, FreqHarm_Vo] = thd (Vo,60,15);
MAG_THD_Cu= db2mag(THD_Cu);
MAG_THD_Vo= db2mag(THD_Vo);

```

También con las funciones *thd* y *db2mag* se obtienen los valores de los armónicos impares y la fundamental, esto en magnitud y para finalizar se ordenan los armónicos en un vector fila, de acuerdo a la variable medida.

```

%ARMONICOS
MAG_F_Cu = db2mag(PotHarm_Cu(1,1));
MAG_F_Vo = db2mag(PotHarm_Vo(1,1));
MAG_3_Cu = db2mag(PotHarm_Cu(3,1));
MAG_3_Vo = db2mag(PotHarm_Vo(3,1));
MAG_5_Cu = db2mag(PotHarm_Cu(5,1));
MAG_5_Vo = db2mag(PotHarm_Vo(5,1));
MAG_7_Cu = db2mag(PotHarm_Cu(7,1));
MAG_7_Vo = db2mag(PotHarm_Vo(7,1));
MAG_9_Cu = db2mag(PotHarm_Cu(9,1));

```

```

MAG_9_Vo = db2mag(PotHarm_Vo(9,1));
MAG_11_Cu = db2mag(PotHarm_Cu(11,1));
MAG_11_Vo = db2mag(PotHarm_Vo(11,1));
MAG_13_Cu = db2mag(PotHarm_Cu(13,1));
MAG_13_Vo = db2mag(PotHarm_Vo(13,1));
MAG_15_Cu = db2mag(PotHarm_Cu(15,1));
MAG_15_Vo = db2mag(PotHarm_Vo(15,1));
MAG_ARMONICOS_Cu = [MAG_F_Cu, MAG_3_Cu, MAG_5_Cu, MAG_7_Cu, MAG_9_Cu, MAG_11_Cu,
MAG_13_Cu, MAG_15_Cu]';
MAG_ARMONICOS_Vo = [MAG_F_Vo, MAG_3_Vo, MAG_5_Vo, MAG_7_Vo, MAG_9_Vo, MAG_11_Vo,
MAG_13_Vo, MAG_15_Vo]';

```

Enseguida, Se calcula el valor *rms*, tanto para la curva de voltaje como para la de corriente.

```

% RMS
RMS_Cu= rms(Cu);
RMS_Vo= rms(Vo);

```

Luego, para obtener el factor de distorsión de voltaje y corriente, se utilizó la fórmula:

$$DF = \frac{\text{ValorFundamental}}{RMS} \quad (1)$$

```

%FACTOR DE DISTORSION
DFV = MAG_F_Cu/RMS_Cu;
DFI = MAG_F_Vo/RMS_Vo;

```

Con la función *findpeaks*, se obtiene en valor pico-pico de las señales cada 0.016 s esto equivale a 60 Hz.

```

% Valor Pico
P_Cu = findpeaks (Cu,T,'minpeakdistance',0.016);
Pico_Cu = abs(P_Cu (1));
P_Vo = findpeaks (Vo,T,'minpeakdistance',0.016);
Pico_Vo = abs(P_Vo (1));

```

Conociendo el valor pico de las señales, se procede a calcular el factor de cresta cuya fórmula es,

$$FC = \frac{ValorPico}{RMS} \quad (2)$$

```
% Factor de Cresta
Factor_Cresta_Cu = Pico_Cu/RMS_Cu;
Factor_Cresta_Vo = Pico_Vo/RMS_Vo;
```

Para calcular las potencias, es necesario conocer el valor del factor de potencia, dicho valor es dado por el software CASSY lab 2, ahora bien, para calcular las potencias se aplican las siguientes fórmulas:

$$PotenciaActiva = \sqrt{3} * V * I * \cos(\varphi) \quad (3)$$

$$PotenciaReactiva = \sqrt{3} * V * I * \sin(\varphi) \quad (4)$$

$$PotenciaAparente = \sqrt{PotenciaActiva^2 + PotenciaReactiva^2} \quad (5)$$

Y se guardan los valores obtenidos en variables, para llevar un orden en el código.

```
% Potencias
Phi = acos(CosPhi);
Activa = sqrt(3)*RMS_Cu*RMS_Vo*cos(Phi);
Reactiva = sqrt(3)*RMS_Cu*RMS_Vo*sin(Phi);
Aparente = sqrt((Activa^2)+(Reactiva^2));
P = Activa (end);
Q = Reactiva (end);
S = Aparente (end);
```

Se guardan los datos en vectores fila, llevando un orden para no tener confusiones.

```
% Guardamos Datos
Datos_Cu= [MAG_THD_Cu,RMS_Cu,Pico_Cu,Factor_Cresta_Cu,DFI]';
Datos_Vo= [MAG_THD_Vo,RMS_Vo,Pico_Vo,Factor_Cresta_Vo,DFV]';
Armonicos_Cu =[MAG_ARMONICOS_Cu];
Armonicos_Vo =[MAG_ARMONICOS_Vo];
POTENCIAS = [P,Q,S,CosPhi]';
CORRIENTE = [Datos_Cu ; Armonicos_Cu];
VOLTAJE = [Datos_Vo ; Armonicos_Vo];
```

Por último se guardan los todos los datos en un vector fila, que gracias a la función *xlswrite*, escribe los datos en un documento de Excel.

```
TOTAL = [POTENCIAS;CORRIENTE;VOLTAJE];
xlswrite('MOTOR_VARIANZAS_T_Y.xlsx',TOTAL,3,'K2')
```