JIDeTEV
JORNADAS DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO, EXTENSIÓN, VINCULACIÓN Y MUESTRA DE LA PRODUCCIÓN
Universidad Nacional de Misiones
Facultad de Ingeniería

Vol1-Año 2019-**ISSN 2591-4219**

Caracterización de un sensor de temperatura*

Mateo A. Mazur [a] , Jorge A. Olsson [a] , Héctor R. Anocibar [a] ,

[a] *Facultad de Ingeniería, Universidad Nacional de Misiones (UNaM), Oberá, Misiones, Argentina. Departamento de Ingeniería Electrónica*

e-mails: mateo.amz@gmail.com, olsson@fio.unam.edu.ar, anocibar@fiobera.unam.edu.ar

*Relacionado al trabajo de investigación 16/I142 Pequeñas centrales hidroeléctricas - equipos y sistemas

Resumen

Este trabajo fue realizado en la cátedra Mediciones Electrónicas e Instrumentación Industrial. Se propuso el desarrollo de un sistema capaz de caracterizar un sensor de temperatura en cuanto a sus parámetros de no-linealidad, sensibilidad, histéresis, curva característica y modelo matemático. El trabajo se centró en la obtención de estos parámetros de manera semi autónoma, utilizando un sistema de captura de datos con posterior procesamiento mediante computadora, con el objetivo de disminuir las horas hombre requeridas en el proceso de toma de datos y cálculos. Para lograr esto se utilizaron multímetros con interfaz USB en la captura de datos y estos últimos fueron procesados luego utilizando un software basado en Matlab desarrollado para este propósito. Los resultados obtenidos fueron comprobados mediante el cálculo manual de los mismos. Finalmente se concluye que los mismos se encuentran dentro de las tolerancias esperadas, considerados por la cátedra como satisfactorios. Obteniéndose una disminución del 70% del tiempo requerido manualmente.

Palabras Clave *– histéresis, linealidad, ntc, sensibilidad, sensor, temperatura.*

1. Introducción

La caracterización de un sensor de temperatura es un proceso que requiere de su evaluación en distintas curvas de subida y bajada de temperatura, capturando su resistencia en función de la temperatura, como es conocido, este proceso de llevar al sensor por sucesivos procesos de subida y bajada de temperatura debe realizarse a una velocidad relativamente baja para garantizar la fiabilidad de los datos, por lo tanto, requiere que el observador esté en muchos casos horas registrando valores, a este tiempo se le suma el asociado al procesamiento de los datos para el cálculo de los parámetros del sensor, como son: sensibilidad, no-linealidad, histéresis, modelo matemático y gráfico. Este trabajo pretende, a través del uso de multímetros con interfaz USB y del software Matlab, facilitar esta tarea haciéndola más autónoma en cuanto al registro de los valores y obtención de los parámetros del sensor.

2. Desarrollo

Se dividió el desarrollo del proceso de caracterización en dos partes, la primera consistió en la obtención de los datos que conforman una curva de bajada y subida de temperatura del sensor, de esta manera se obtuvieron los datos que corresponden a valores de resistencia en función de la temperatura. Una segunda parte consistió en el procesamiento de estos datos, para obtener a partir de los mismos los parámetros que caracterizan al sensor.

2.1. Obtención de datos

En este ensayo se relevaron los valores de resistencia que posee el sensor para una temperatura de entre 10 °C y -10 °C, para ello se realizó un barrido de bajada y otro de subida de temperatura (en este orden). Ambas variables fueron obtenidas con multímetros UNI-T 61 [1], los cuales tienen la capacidad de medir resistencia [Ohm] y temperatura [°C] a través de su respectiva termocupla.

El primer paso consistió entonces, en lograr que la termocupla del multímetro y el sensor de temperatura bajo ensayo permanezcan cerca uno de otro, esto resultó fácil para el caso estudiado debido a la forma que presenta el sensor bajo estudio, pudiendo adherir ambos sensores utilizando cinta de teflón como se ve en la Fig. 1.

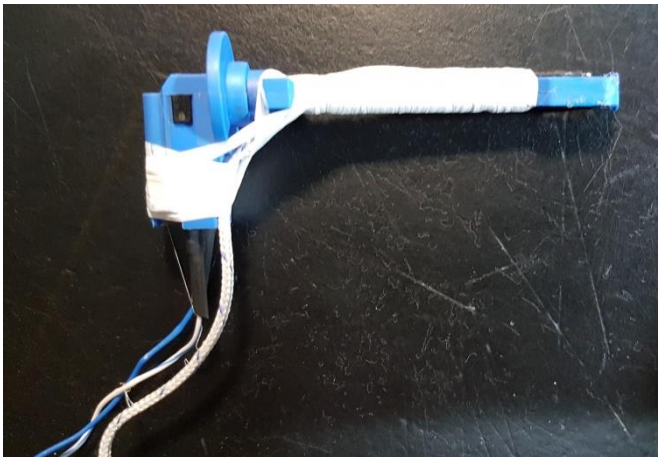

Fig. 1. Sensor bajo ensayo y termocupla.

Debido a las temperaturas bajo cero a las cuales se ensayó el sensor, fue necesario sumergir el mismo en alcohol para evitar el congelamiento. En este punto resulta importante utilizar un recipiente que funcione como aislante térmico, pudiendo ser un recipiente de telgopor o un vaso térmico, ya que de otra manera la temperatura que se registra durante la curva de subida aumenta de manera rápida y puede generar errores en el proceso de obtención de datos.

Una vez listos los sensores y recipientes, se realizó el conexionado correspondiente y la configuración de los multímetros para capturar los datos de manera ininterrumpida. Para esto, los multímetros UNI-T se encendieron manteniendo presionado el botón azul que poseen, esto desactiva el apagado automático de los mismos. Seguidamente ambos multímetros se conectaron mediante su interfaz USB a una computadora y en la misma se abrió el software provisto por el fabricante [2] para la captura y almacenamiento de datos. Cabe mencionar que ambos multímetros funcionan de manera independiente, por lo tanto, se debió abrir una ventana del software para cada multímetro. En Fig. 2 se puede apreciar la ventana principal del software de captura del multímetro.

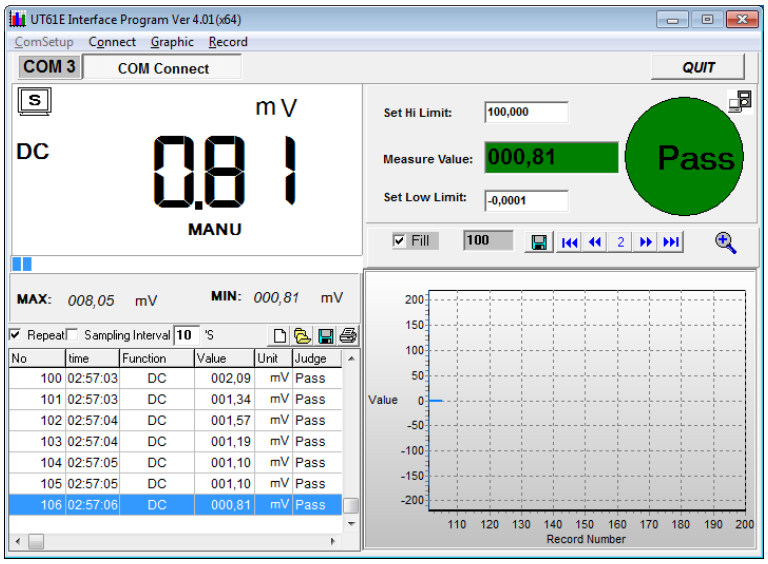

Fig. 2. Software del multímetro.

Desde esta ventana se pudo configurar el periodo de captura de datos, el software se encuentra continuamente leyendo datos del multímetro, pero el almacenamiento de los datos capturados en un formato .xls o .txt se da al presionar el botón de "guardar" en el software.

Teniendo un multímetro configurado para la adquisición de los valores de resistencia del sensor bajo ensayo, y otro asociado a la medición de temperatura con su termocupla, se procedió a iniciar la captura de datos en ambos, presionando el botón "REL" en ambos multímetros a la vez y colocando los sensores en un ambiente frio (para realizar la curva de bajada). A partir de este momento, se dejó el sistema capturando hasta que se alcanzó la temperatura final deseada. Una vez que esto sucedió, se situó el recipiente con los sensores en un ambiente más cálido sin modificar ningún parámetro de los multímetros.

Cuando se alcanzó la temperatura final deseada, se procedió a la captura de los datos obtenidos a partir de ambos multímetros, esto se hizo presionando el botón "guardar" y eligiendo como formato de salida .xls. El archivo generado tiene un formato como el presentado en la Fig. 3.

	A	B	C	D	E	F
1	No	Time	DC/AC	Value	Unit	AUTO
2	1	15:23:04	OHM	3,386	k*	pass
3	2	15:23:09	OHM	3,391	k*	pass
4	3	15:23:14	OHM	3,397	k*	pass
5	4	15:23:19	OHM	3,401	k*	pass
6	5	15:23:24	OHM	3,407	k*	pass
7	6	15:23:29	OHM	3,412	k*	pass
8	7	15:23:34	OHM	3,417	k*	pass
9	8	15:23:39	OHM	3,422	k*	pass
10	9	15:23:44	OHM	3,428	k*	pass

Fig. 3. Tabla de datos a partir de los multímetros.

Donde la cuarta columna presenta la variable que fue capturada por el multímetro. Hecho esto, se terminó con la parte de obtención de datos y se procedió a su procesamiento.

2.2. Procesamiento de datos

Para el procesamiento de los datos y la obtención de parámetros como, sensibilidad, histéresis, función matemática del sensor y gráfico, se diseñó un programa en Matlab.

La primera tarea que realiza el programa consiste en adquirir los datos de los multímetros en formato .xls y organizarlos en una matriz, ya que, dependiendo del periodo de muestreo, existen muchos valores de resistencia asociados a un mismo valor de temperatura, además, los valores de temperatura no se encontraban correctamente ordenados de manera creciente o decreciente, debido a que la temperatura dada por los multímetros oscila cuando esta se encuentra próxima a cambiar un digito. Para generar la matriz con datos útiles, el programa primero debe determinar el punto de inflexión, que separa la curva de bajada de la curva de subida, para esto se diseña el siguiente código.

```
i=2
while res(i-1) <= res(i)          %i determina el punto de inflexión
i=i+1
end
tembaj=tem(1:i)                   % curva de bajada para la temperatura
temsub=tem(i+1 : length(tem))     % curva de subida para la temperatura
resbaj=res(1:i)                   % curva de bajada para la resistencia
ressub=res(i+1 : length(res))     % curva de subida para la resistencia
```

Una vez separadas ambas curvas, el programa determina cuantos valores de temperatura distintos existen, y con esta información, carga una matriz que en su columna 1 posee estos valores. Esta tarea se realiza a través del siguiente código.

```
while(i<length(tem))
      if(tem(i+1)<temp)
      filas=filas+1;
      temp=tem(i+1)
      end
i=i+1;
end
while(i<length(tem))
     if(tem(i+1)<temp)
     matriz(j,1)=tem(i+1)
     temp=tem(i+1)
     j=j+1;
     end
i=i+1;
end
```

Estos valores de temperatura sirven al programa de guía para determinar el valor de resistencia asociado a cada temperatura. Este valor de resistencia se determina a través del promedio de todos los valores capturados que están asociados a un mismo valor de temperatura. Los resultados se cargan en la matriz de temperaturas a través del siguiente código.

```
while(j<=filas)
temp=matriz(j,1);
i=1;
suma=0;
cant=0;
while(i<length(resbaj))

     if(tembaj(i)==temp)
     cant=cant+1;
     suma=suma+resbaj(i)
     end
 i=i+1;
end
r=suma/cant;
matriz(j,2)=r;
j=j+1;
end
```

Un código similar se repitió para procesar los datos de la curva de subida. Generando así una matriz que en su primera columna posee valores de temperatura, en su segunda columna posee los valores de resistencia asociados a la curva de bajada y finalmente en la tercera columna tiene cargados los valores de resistencia de la curva de subida. A partir de esta matriz el programa procede al cálculo de los parámetros que caracterizan al sensor.

2.3. Calculo de sensibilidad:

La sensibilidad es para este caso el cociente entre la variación de la resistencia y la variación correspondiente de temperatura. A partir de la matriz antes determinada por el programa, la misma se determina con el siguiente código.

```
deltaX= max(matriz(:,1)) - min(matriz(:,1));
deltaY1= max(matriz(:,2)) - min(matriz(:,2));
deltaY2= max(matriz(:,3)) - min(matriz(:,3));
s=zeros(1,2);
s(1,1)= deltaY1/deltaX;
s(1,2)= deltaY2/deltaX;
sensibilidad=max(s(1,:))
```

El programa calcula la sensibilidad de la curva de subida y de la curva de bajada, pero conserva el valor máximo hallado.

2.4. Calculo de no-linealidad:

La no-linealidad se define como la desviación máxima de la curva (de subida o bajada) característica, respecto a una línea recta ideal. Debido a esto, el programa debe en principio calcular esta recta ideal y luego comparar la misma con la curva que posee en su matriz de datos. Esta tarea se realiza con el siguiente código.

```
syms x
y=((((matriz(filas,2)) - (matriz(1,2)))/((matriz(filas,1))
- (matriz(1,1)))) * (x - (matriz(1,1)))) + (matriz(1,2))
i=1; nl1=0;
while(i<filas)
    temp=subs(y,matriz(i,1))          %evalúa la recta ideal
    temp2=abs(temp-matriz(i,2))       %calcula la diferencia
    if(temp2>nl1)
      nl1=temp2
    end
    i=i+1;
end
```

En este caso la no linealidad calculada se asocia a la curva de bajada, se repite un código similar para calcular la no-linealidad de la curva de subida. El programa evalúa ambos valores y conserva el mayor.

2.5. Calculo de histéresis:

La histéresis se calcula a partir de la máxima diferencia que se observa entre la curva de subida y de bajada. De esta manera el código utiliza la columna 2 y 3 de su matriz (asociadas a las curvas de bajada y subida respectivamente). El código que calcula la histéresis es el siguiente.

```
while(i<filas)
    temp=abs(matriz(i,2) - matriz(i,3))
    if(temp>his)
        his=temp
    end
    i=i+1
end
fe=max(fondoescala(1,:))
histeresis=(his/fe)*100;
```

2.6. Obtención del modelo matemático del sensor:

Matlab posee una función denominada “fit” [3], la cual, a partir de un modelo matemático definido, obtiene los coeficientes que se corresponden a ese modelo referido a un conjunto de valores. Esta función se utilizó para hallar el modelo matemático del sensor, partiendo del conocimiento de que el mismo responde a una exponencial decreciente, comportamiento característico de un NTC. El código que se ejecuta es el siguiente.

```
f1=fit(matriz(:,1),matriz(:,2),'exp1')
coef=zeros(1,2)
coef=coeffvalues(f1)
a=coef(1,1)
b=coef(1,2)
```

2.7. Gráfico de los datos:

Para graficar los datos obtenidos se utilizan la función “plot” [4], el código realiza un gráfico para la curva de bajada y otro para la curva de subida a través del siguiente código.

```
figure
plot(matriz(:,1),matriz(:,2));
hold on;
plot(matriz(:,1),matriz(:,3));
```

Un objetivo planteado para este proyecto fue el de hacerlo sencillo de utilizar, por lo tanto se pretende que el programa pueda ser ejecutado en una computadora que no necesariamente deba tener instalado el software Matlab, este objetivo se logró cumplir utilizando la App de Matlab denominada “Application Compiler” [5], la cual toma un código y lo procesa para formar un programa que pueda ser instalado y ejecutado de manera independiente, es decir, sin contar con una previa instalación de Matlab. Se obtuvo por lo tanto un archivo de extensión .exe que puede ser fácilmente instalado en una computadora y que al ejecutarlo procesa los datos como se ha mencionado anteriormente. Este programa arroja los resultados como se muestra en la Fig. 4.

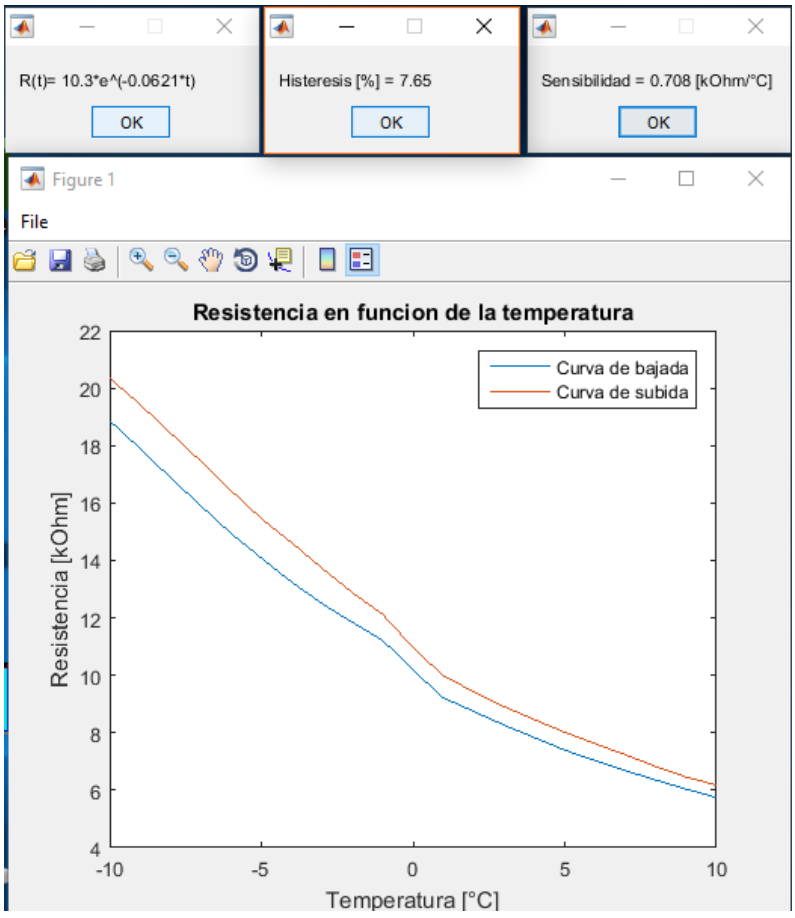

Fig. 4. Resultados.

Se puede observar que los cuadros de dialogo arrojados por el programa indican el modelo matemático del sensor, el cálculo de la histéresis, la sensibilidad y el grafico a partir de los datos obtenidos en el archivo .xls ubicado en la misma carpeta desde donde se ejecuta el programa.

3. Conclusiones

Se concluye que la implementación de un sistema semi automático para la caracterización de sensores libera al usuario luego de preparar el ensayo con la captura de datos en forma automática. Se obtiene el beneficio adicional de que estos pueden ser procesados con software específicos. Además de lograrse los objetivos planteados, se concluye finalmente que el sistema puede ser mejorado en nuevas etapas incorporando nuevas funciones o adaptándolo a otro tipo de ensayos.

Referencias

[1] Uni-Trend Technology. (s.f.). UT61C Modern Digital Multimeter. Recuperado de http://www.uni-trend.com/html/product/General_Meters/Digital_Multimeters/UT61_Series/UT61C.html

[2] Uni-Trend Technology. (s.f.). Interface Software. Recuperado de http://www.uni-trend.com/html/servicesupport/DriverFirmwareSoftware/Interface1Software/5.html

[3] The MathWorks, Inc. (s.f.). fit. Recuperado de https://la.mathworks.com/help/curvefit/fit.html

[4] The MathWorks, Inc. (s.f.). plot. Recuperado de https://la.mathworks.com/help/Matlab®/ref/plot.html

[5] The MathWorks, Inc. (s.f.). Getting Started with MATLAB® Compiler. Recuperado de https://la.mathworks.com/help/compiler/getting-started-with-Matlab®-compiler.html