Wikipédia

Relation de Steinhart-Hart

La **relation de Steinhart–Hart** modélise l'évolution de la résistance électrique d'un semi-conducteur selon sa température. Les composants exploitant cette propriété s'appellent des thermistances. Cette loi peut s'écrire :

$$\frac{1}{T} = A + B\ln(R) + C(\ln(R))^3$$

Avec :

- T sa température (en kelvins);
- R sa résistance électrique (en ohms);
- A, B et C les coefficients de **Steinhart-Hart** qui caractérisent chaque thermistance.

Cette relation est valide sur toute la plage de fonctionnement du composant. Il existe en revanche des formules plus faciles à manipuler mais limitées à une gamme restreinte de températures (voir l'article thermistance).

L'équation contient aussi, en théorie, un terme en $(\ln R)^2$ généralement négligeable devant les autres coefficients. C'est pourquoi il n'est pas considéré ici. (En sa présence il y aurait alors 4 coefficients.)

Sommaire

Utilisation

Cette relation permet souvent d'estimer avec précision la résistance d'une thermistance selon la température sur toute sa plage de fonctionnement. Tandis que les équations, certes plus simples, données par les fabricants ne sont souvent précises que sur certains intervalles de température. Il peut donc parfois être utile de disposer de cette loi plus délicate mais toujours précise.

Les coefficients de Steinhart–Hart sont parfois publiés par les fabricants. Si ce n'est pas le cas, il faut résoudre un système à 3 équations et 3 inconnues pour trouver ces constantes A, B et C.

Inversion

On peut chercher la relation réciproque (obtenir R en sachant T), avec T en kelvins et R en ohms.

$$R = \exp\left(\sqrt[3]{y - \frac{x}{2}} - \sqrt[3]{y + \frac{x}{2}}\right)$$

Avec

$$x = \frac{1}{C}\left(A - \frac{1}{T}\right)$$
$$y = \sqrt{\left(\frac{B}{3C}\right)^3 + \left(\frac{x}{2}\right)^2}$$

Programme en Python 2.7 pour calculer la relation ci dessus:

```
#!/usr/bin/python2.7
# Cyrus.rocher@femto-st.fr/ (Femto-st Institut)Last modification; 02/Oct/2018
import math
from math import log as ln# on francise le log en log neperien ln
from math import exp
#---coefficients de Steinhart-Hart pour l'exemple---#
A= 0.0011268740732306604
B= 0.00023452183442732656
C= 8.590172470421073e-08
#---------------------------------------------------#
x=(1/C)*(A-1/T)
y=((B/3/C)**3+(x/2)**2)**0.5
Tc= T-273.15 # temperature en degres Celsius
R= exp(pow((y-(x/2)),1/3.0)-(pow((y+(x/2)),1/3.0)))
#print(x,y)
#print(A,B,C)
print(R,Tc)
```

Coefficients de Steinhart-Hart

Pour trouver les coefficients de Steinhart-Hart il suffit de connaître trois points de fonctionnement et de poser un système. Pour cela, on utilise trois valeurs de résistance données pour trois températures connues.

$$\begin{cases} A + (\ln R_1).B + (\ln R_1)^3.C = \dfrac{1}{T_1} \\ A + (\ln R_2).B + (\ln R_2)^3.C = \dfrac{1}{T_2} \\ A + (\ln R_3).B + (\ln R_3)^3.C = \dfrac{1}{T_3} \end{cases}$$

Avec R_1, R_2 et R_3 les valeurs de la résistance aux températures respectives T_1, T_2 et T_3, on peut alors exprimer A, B et C après quelques substitutions :

On pose $Y_1 = \dfrac{1}{T_1}$ $Y_2 = \dfrac{1}{T_2}$ et $Y_3 = \dfrac{1}{T_3}$ ainsi que $L_1 = \ln R_1$ $L_2 = \ln R_2$ et $L_3 = \ln R_3$

On obtient alors :

$a = \left(\dfrac{L_2 - L_3}{L_1 - L_2}\right) \times (L_2^3 - L_1^3) + (L_2^3 - L_3^3)$ et $b = Y_2 - Y_3 - \left(\dfrac{L_2 - L_3}{L_1 - L_2}\right) \times (Y_1 - Y_2)$

Les coefficients s'obtiennent comme ceci :

$$C = \frac{b}{a}$$

$$B = \left(\frac{1}{L_1 - L_2}\right) \times \left[Y_1 - Y_2 - (L_1^3 - L_2^3) \times C\right]$$

$$A = Y_1 - L_1 \cdot B - L_1^3 \cdot C$$

D'où le petit programme (largement améliorable) ci-dessous écrit en Python2.7, permettant de déterminer ces coefficients.

```
#!/usr/bin/python2.7

# on francise le log en log neperien ln !
from math import *
from math import log as ln

Tun = eval(input('Entrez la temperature T1 du premier point (en degres Celsius) : '))
Run = eval(input('et la resistance R1 du premier point (en ohms) : '))
print('\t---------------------------------')
Tdeux = eval(input('Entrez la temperature T2 du deuxieme point (en degres Celsius) : '))
Rdeux = eval(input('et la resistance R2 du deuxieme point(en ohms) : '))
print('\t---------------------------------')
Ttrois = eval(input('Entrez la temperature T3 du troisieme point (en degres Celsius) : '))
Rtrois = eval(input('et la resistance R3 du troisieme point (en ohms) : '))

# calculs en kelvins
Tun = Tun + 273.15
Tdeux = Tdeux + 273.15
Ttrois = Ttrois + 273.15

# changement de variables
Yun = 1/Tun
Ydeux = 1/Tdeux
Ytrois = 1/Ttrois

Lun = ln (Run)
Ldeux = ln (Rdeux)
Ltrois = ln (Rtrois)

# calculs intermediaires
a = (Ldeux-Ltrois)/(Lun-Ldeux)*(pow (Ldeux,3) - pow (Lun,3)) + (pow (Ldeux,3) - pow (Ltrois,3))
b = Ydeux - Ytrois - ((Ldeux-Ltrois)/(Lun-Ldeux))*(Yun-Ydeux)

# calculs de A, B et C
C = b / a
B = (1/(Lun-Ldeux))*(Yun-Ydeux-C*(pow(Lun,3) - pow(Ldeux,3)))
A = Yun - B*Lun - C*pow (Lun,3)

#Affichages de A, B et C
print('\t ###################################################################')
print('Dans l\'equation 1/T = A + B*ln R + C*(ln R)^3 on sait désormais que :')
print('\t ###################################################################')
print('Le coefficient A vaut ', A)
print('Le coefficient B vaut ', B)
print('Le coefficient C vaut ', C)
```

Origines de la relation

Le nom de cette équation vient de John S. Steinhart et Stanley R. Hart qui ont été les premiers à la publier[1]. C'est à la Carnegie Institution of Washington que la formule a été trouvée.

Le professeur Steinhart (1929-2003), était membre de l'université de Madison au Wisconsin de 1969 à 1991 [1] (http://www.secfac.wisc.edu/senate/2004/0405/1775(mem_res).pdf).

Le docteur Hart, scientifique éminent de l'Institut océanographique de Woods Hole, était aussi membre entre autres de la Geological Society of America et de la **(en)**European Association of Geochemistry.

Références

1. "Calibration curves for thermistors", Deep Sea Res., **15**, 497-503 (1968).