

# A Neural Network Approach for Creating a NTC Thermistor Model Library for PSPICE

Lian Ming Wang  
Institute of Applied Electronic Technology,  
Northeast Normal University  
ChangChun, China  
Wang\_lianming@yahoo.com.cn

Yu Fen Deng , Xian Long Zhao, Bao Liang Liu  
Institute of Hydrographic Surveying and Charting,  
Tianjin,China  
dengyufen@yahoo.com.cn , zhaoxianlong@126.com  
baoliang\_liu@126.com

**Abstract**—most sensors can not be modeled easily, which leads to the problem that a circuit with sensors can not be simulated in PSPICE. A method based on the neural network for modeling NTC thermistors and creating a NTC thermistor model library for PSPICE is presented to solve the problem. Firstly, a multi-layer feedforward neural network is used to approximate the characteristics of a NTC thermistor. Secondly, the achieved structure of the neural network is described in the PSPICE language to form a subcircuit. Thirdly, the structure is used to model the same series of NTC thermistors by changing weights and biases of the neural network. Finally, the subcircuits for the series of NTC thermistors can be packed into a file to create a model library. During PSPICE simulation, the variations of a non-electric quantity imposed on a sensor are replaced with those of an electric quantity. The availability of this method is verified in circuit simulation. This method can be extended to model other sensors.

**Keywords**—Model library, Neural network, PSPICE, NTC thermistor

## I. INTRODUCTION

The success of modern electronics is built on the possibility to accurately predict system behavior by the use of simulation tools. PSPICE is a powerful general purpose analog circuit simulator that is used to verify circuit designs and to predict the circuit behavior. Since there are abundant model libraries of electronic devices in PSPICE, it is convenient to simulate all kinds of electronic circuits. Unfortunately, it is hard to simulate circuits with sensors; the reason is that there are few sensor model libraries in PSPICE. The ability to simulate both sensors (non-electronic components) and electronic devices together in PSPICE will render possible effective optimizations at system design; thus modeling sensors in PSPICE is imperative and significant.

The biggest difference between modeling a sensor and modeling an electron device is that the input of a sensor is always a non-electric quantity. Since it is impossible to generate a non-electric signal in PSPICE, circuits with sensors are hard to be simulated.

Some companies [3] introduced model libraries of NTC thermistors, enabling PSPICE users to simulate temperature measurement and compensation. Each NTC model in the libraries is composed of three Voltage-Controlled Current Sources, several resistors and capacitors. The controlling relationship of the three Voltage-Controlled Current Sources is

described by three different function expressions in PSPICE. It is unfortunate that the circuits using these NTC thermistor models can only be simulated when the Temperature Analysis is performed in PSPICE, since the temperature is expressed by using the global variable TEMP, in the model. Thus, this method can not be generalized to other sensors, such as pressure, acceleration, shock and vibration sensors.

Generally, there are two approaches to model devices: physical modeling and table modeling [6]. The physical approach for obtaining a suitable compact model for circuit simulation has to make use of available physical knowledge, and to forge that knowledge into a numerically well-behaved model. A major disadvantage of physical modeling is that it usually takes years to develop a good model for a new device.

Table modeling is a modeling technique that can in principle obtain a static model of any required accuracy by providing a sufficient amount of discrete data. It can not be a general approach, since a general approach for developing a model should not only closely match the behavior as specified in the data set, but also yield reasonable outcomes for situations not specified in the data set. However, high accuracy in table models can be obtained only by using enormous table points, since the expression between two points is implemented by piecewise linear interpolation[6].

Neural network applications in device and subcircuit modeling for circuit simulation are discussed by Meijer [7]. But emphasis is merely put on modeling electronic devices and systems. Some special sensors are modeled with different methods in different application areas [8] [9] [10].

To find a general, simple and effective method to model different kinds of sensor, a multi-layer feedforward neural network is considered, which can approximate any given continuous function on any compact subset to any degree of accuracy, providing that a sufficient number of hidden layer neurons are used. Since characteristics of the sensors within the same series are always similar, they can be approximated by neural networks with the same structure and different weights and biases. This idea lays foundation for our approach to creating a sensor model library, which will be exemplified by the process of creating a NTC thermistor model library below.

It should be noticed that the thermistor can in many cases be treated as a constant, static resistor, whose resistance only varies with temperature. However, when the power dissipation

in the thermistor raises the device temperature considerably, a more complicated nonlinear, dynamic model must be applied to yield accurate simulation results [11] [12] [13]. We just focus on static modeling approach for NTC thermistor using neural network in this paper.

## II. NEURAL NETWORK TRAINING

We chose EPCOS B57164 series NTC thermistor as the sensors to be modeled.

A thermistor is a kind of electronic components that exhibits a large change in resistance with a change in its body temperature. The thermistors have either large positive temperature coefficient of resistance (PTC devices) or large negative temperature coefficient of resistance (NTC devices).

The Resistance-Temperature characteristic of a NTC thermistor can be theoretically expressed as:

$$R_T = R_{T_0} \exp\left[\frac{\beta(T_0 - T)}{TT_0}\right] \quad (1)$$

Where:  $R_T$  is the resistance at an absolute temperature  $T$  expressed in Kelvin ( $^{\circ}\text{C}+273.15$ );  $\beta$  is the material constant;  $R_{T_0}$  is the resistance at a specified reference temperature,  $T_0$ , which is also expressed in Kelvin. However, for a real NTC thermistor, the Resistance-Temperature characteristic can not be depicted as an explicit expression, since  $\beta$  is not a constant, which varies with temperature.

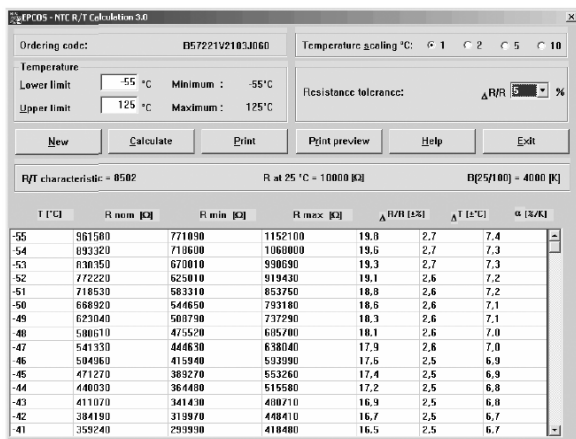


Figure 1. Interface of EPCOS NTC R/T Calculation

A NTC resistance - temperature relationship calculation program, EPCOS NTC R/T Calculation, is provided by the EPCOS Company, in which a resistance-temperature characteristics table listing resistance value, resistance tolerance, temperature tolerance and temperature coefficient for the selected thermistor can be given. The data points representing the characteristics of a sensor can be obtained from measurements or device simulations (or from existing datasheet, as is done in this paper for testing purpose).

The interface of the EPCOS NTC R/T Calculation tool is shown in Figure 1, where the nominal resistance  $R_{nom}$ , maximum resistance  $R_{max}$  and minimal resistance  $R_{min}$  at

rated temperature are calculated by selecting temperature range and resistance tolerance.

The resistance - temperature data acquired from the EPCOS NTC R/T Calculation directly for the B57164 series NTC thermistors, which contain 27 types of resistance, are shown by curves in Figure 2. The temperature ranges from  $-55^{\circ}\text{C}$  to  $125^{\circ}\text{C}$ .

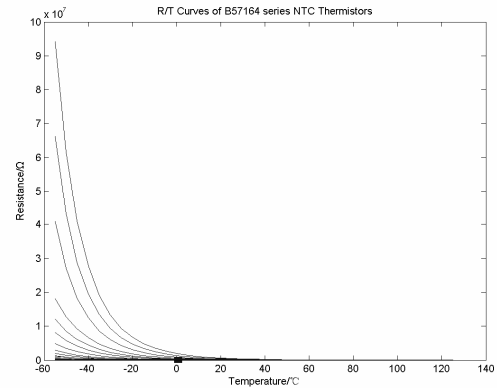


Figure 2. R/T curves of B57164 series NTC thermistors

A single-input and single-output feedforward neural network with one hidden layer was chosen to approximate the characteristic. We used the backpropagation learning algorithm with momentum updating to train the neural network to approximate the curves by programming in the Matlab [14]. To make the Resistance-Temperature data suitable for training, the output resistance data  $R_T$  for the neural network must be normalized, since the activation function of the output layer is the logarithmic sigmoid function, whose output amplitude range is the open interval (0, 1). To speed up the training of the neural network, resistance for each sensor was normalized by dividing by ten times of its maximum  $R_{Tmax}$ , and the temperature was normalized by dividing by 50 as well, by which the training proceeded near the steepest part of the activation function. That is, the input of neural network  $T_{nn}$  should be,

$$T_{nn} = \frac{T}{50} \quad (2)$$

The desired output of neural network,  $R_{nn}$ , should be

$$R_{nn} = \frac{RT}{10 \times RT_{max}} \quad (3)$$

The normalized resistance-temperature curves of B57164 series NTC thermistors are shown in figure 3. The simulation experiments proved that this normalization can accelerate the convergence and improve the approximation accuracy.

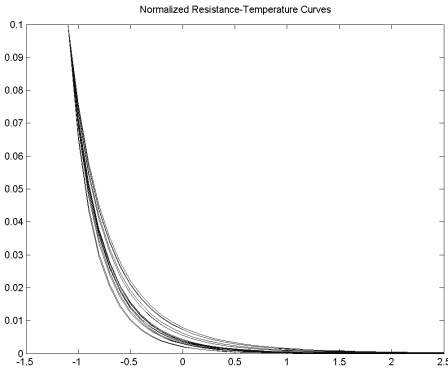


Figure 3. Normalized R/T curves

The error goal for the neural network training can be set by the tolerance assigned to the model. According to the equation (3), the minimum error goal should satisfy equation (4) below, where RTmin is the minimum of all the resistances for a specific thermistor to be modeled.

$$error\_goal = \left( \frac{Tolerance \times RT\ min}{10 \times RT\ max} \right)^2 \quad (4)$$

By decreasing the number of the neurons in the hidden layer and adjusting the learning rate, we obtained an optimum structure finally [14] [15]. The neural network is composed of six neurons in the hidden layer. The signal functions for the hidden layer and the output layer are TANSIG and LOGSIG respectively. The structure achieved is shown in Figure 4. It can implement the error goal for each thermistor in the series. Thus, each thermistor can be represented by a group of weights and biases,  $W_{1i}$ ,  $W_{iR}$ ,  $\theta_i$  ( $i=1, 2, 3, 4, 5, 6$ ) and  $\theta_R$ .

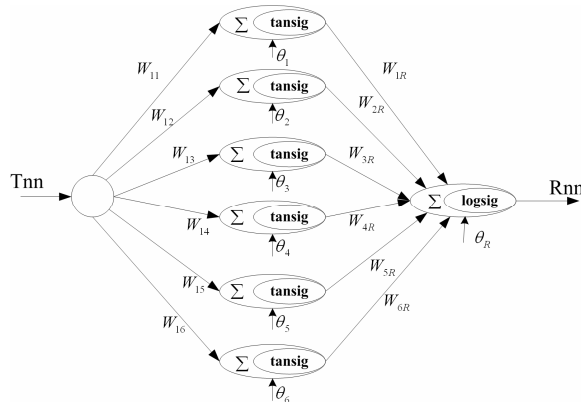


Figure 4. Neural Network structure achieved

### III. MODEL SUBCIRCUIT DESCRIPTION

To make the model applicable to circuit simulation in PSPICE, the structure of the neural network should be described in PSPICE language [19]. In PSPICE, there are four kinds of dependent sources, which are Voltage Controlled Voltage Sources (E), Current Controlled Current Sources (F),

Voltage Controlled Current Sources (G), and Current Controlled Voltage Sources (H). The controlling relationship of these dependent sources can be expressed as an arithmetic expression (VALUE), a polynomial (POLY), a lookup table (TABLE) or a Laplace transform expression (LAPLACE) etc.

Each neuron in the hidden-layer and the output layer can be formed by two VCVSs, according to the expression of input-output relationship.

The expression for the neurons in the hidden layer is:

$$X_i = \tan\ sig(W_{1i}T_m + \theta_i) \quad (5)$$

Where:  $X_i$  is the output of the  $i$ th neuron, the function of  $\tan\ sig$  is:

$$\tan\ sig(\alpha) = \frac{2}{1 + e^{-2\alpha}} - 1 \quad (6)$$

The expression for the neuron in the output layer is:

$$R_m = \log\ sig\left(\sum_i X_i W_{iR} + \theta_R\right) \quad (7)$$

The function of  $\log\ sig$  is:

$$\log\ sig(\alpha) = \frac{1}{1 + e^{-\alpha}} \quad (8)$$

Therefore, each neuron in the hidden layer can be implemented with the circuit in Figure 5.

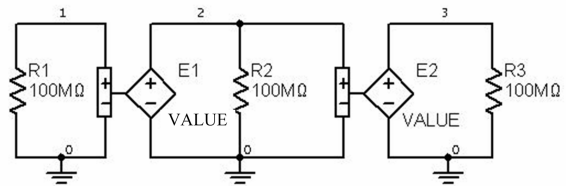


Figure 5. A hidden layer neuron model circuit

The subcircuit description for the  $i$ th neuron in the hidden layer is given below.

```
.SUBCKT NNi 1 3 PARAMS: B1i=1 W1i=1
R1 1 0 1000M
R2 2 0 1000M
R3 3 0 1000M
E1 2 0 VALUE= {B1i+W1i*V(1, 0)}
E2 3 0 VALUE= {2/(1+EXP(-2*V(2, 0)))-1}
.ENDS
```

The neuron in the output layer can be described as below.

\* The description of the output neuron

```
.SUBCKT NN7 1 2 3 4 5 6 8 PARAMS: B2=1 W21=1
W22=1 W23=1 W24=1 W25=1 W26=1
R5 7 0 1000M
R6 8 0 1000M
E1 7 0 VALUE= {B2+W21*V (1, 0)+W22*V (2, 0)
+W23*V(3,0)+W24*V(4,0)+W25*V(5,0)+W26*V(6,0)}
E2 8 0 VALUE= {1/ (1+EXP (-V (7, 0)))}
.ENDS
```

Till now, the neural network has been described in PSPICE language. Unfortunately, the output physical quantity of the neural network is a voltage. To make it behave as a resistor, an additional transition circuit is needed. A variable resistor can be modeled in PSPICE as below according to Ohm's Law [16] [17] [18] [19].

```
.SUBCKT VARIRES 1 2 CTRL
R1 1 2 1E10
G1 1 2 Value = {V (1, 2)/ (V (CTRL) +1u)}
.ENDS
```

The very small number 1u is used to avoid division by zero. The quantity of the controlling voltage V (CTRL) is equal to the resistance to be obtained. Figure 6 shows the ultimate structure of the NTC thermistor model based on neural network.

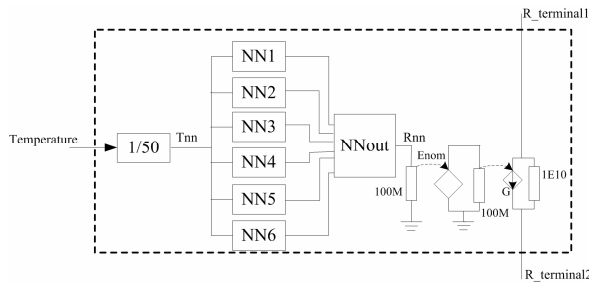


Figure 6. NTC thermistor model based on Neural Network

Since the resistance for each sensor has been divided by ten times of its maximum for normalization, the Enom is used for denormalization.

We may find that the model of a 2-terminal sensor has three terminals. The additional terminal is the input port of a non-electronic quantity which will be substituted with an electronic quantity in PSPICE simulation.

#### IV. MODEL LIBRARY CREATION

PSPICE allow a subcircuit to accept parameter values passed from the subcircuit call line. This characteristic can be used to create a model library. We can form a subcircuit with some general parameters to represent the neural network firstly. Then each sensor model can be created by calling the subcircuit with real parameters. Finally, a sensor model library can be formed by packing all the subcircuits into a file.

The structure of the library is shown below. The model for B57891M0102 NTC thermistor is given as an example. The

models of other sensors can be created by simply calling the NTC subcircuit with corresponding parameters.

#### NTC THERMISTOR LIBRARY

\* The general structure description of a sensor with \*parameters to be passed

```
.SUBCKT NTC 14 11 12 PARAMS: W11=1, W12=1,
+W13=1, W14=1, W15=1, W16=1, B11=1, B12=1, B13=1,
+B14=1, B15=1, B16=1, W21=1, W22=1, W23=1, W24=1,
+W25=1, W26=1, B2=1, INGAIN=1, OUTGAIN=1,
+RTmax=1
```

```
E1 10 0 value= {INGAIN*V (14, 0)}
R7 14 0 1000M
X1 10 1 NN1 PARAMS: B11= {B11}, W11= {W11}
X2 10 2 NN2 PARAMS: B12= {B12}, W12= {W12}
X3 10 3 NN3 PARAMS: B13= {B13}, W13= {W13}
X4 10 4 NN4 PARAMS: B14= {B14}, W14= {W14}
X5 10 5 NN5 PARAMS: B15= {B15}, W15= {W15}
X6 10 6 NN6 PARAMS: B16= {B16}, W16= {W16}
X7 1 2 3 4 5 6 8 NN7 PARAMS: B2= {B2} W21= {W21}
+W22= {W22} W23= {W23} W24= {W24} W25= {W25}
+W26= {W26}
```

```
R1 1 0 1000M
R2 2 0 1000M
R3 3 0 1000M
R4 4 0 1000M
R5 5 0 1000M
R6 5 0 1000M
E2 9 0 value= {OUTGAIN*RTmax*V (8, 0)}
R9 9 0 1000M
R8 8 0 1000M
R10 11 12 1E10
G1 11 12 Value= {V (11, 12)/ (V (9) +1u)}
.ENDS
```

\*\*\*\*\*model of B57164K0101 \*\*\*\*\*

```
.SUBCKT B57164K0101 1 2 3
Xt 1 2 3 NTC PARAMS: W11=0.478301 W12=0.010728
+W13=0.766984 W14=-0.638282 W15=0.398080
+W16=-0.706663 B11=1.108261 B12=-0.426180
+B13=1.192934 B14=0.386648 B15=0.339571
+B16=-1.120134 W21=-1.067615 W22=0.024264
+W23=-1.623335 W24=1.239622 W25=-1.045151
+W26=1.694760 B2=-1.618273 INGAIN=0.02 OUTGAIN=10
+RTmax=4213.000000
.ENDS
```

## V. MODEL VERIFICATION

The validity of the achieved model is confirmed by a simulation experiment. A testing circuit for NTC thermistor is shown in Figure 7. The shadowed part with three terminals is the NTC thermistor model. The electronic quantity  $V_t$  is used to simulate temperature. We use DC Sweep Analysis in PSPICE to make the  $V_t$  sweep from -60V to 140V with 0.1V interval. This corresponds to the changes of temperature from -60°C to 140°C. When the current source IIN is set to 1A, the value of the voltage between node 2 and node 0 should be equal to the resistance varying with temperature according to the Ohm's law.

The description of the circuit is shown as below.

```
TestNTC
```

```
.Lib selfntc.lib
```

```
Xsensor 1 2 0 B57164K0101
```

```
Vt      1  0  1
```

```
IIN     0  2  1A
```

```
.DC     Vt -60 140 0.1
```

```
.PROBE
```

```
.END
```

The voltage curve of V (2) is shown in figure 8. The data points in the curve fit the characteristic of B57164K0101 below tolerance.

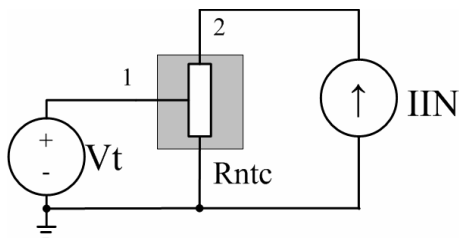


Figure 7. Testing circuit for NTC thermistor

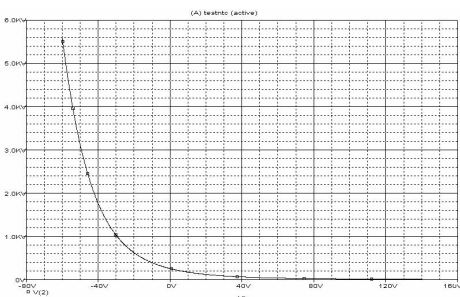


Figure 8. Voltage curve of V (2)

## VI. CONCLUSIONS AND DISCUSSIONS

An approach for modeling sensors based on neural network in PSPICE is presented in this paper. The approach can be general because it is applicable to all the sensors whose characteristics are continuous. The model structures achieved

by this approach are clear and comprehensible. A sensor model library for a series of NTC thermistors has been created to show the validity of this method.

When the output of a sensor is related to several input quantities, a multi-input and single output multi-layer feed-forward may be used to create the model, and such a modification can be the inclusion of self heating parameters into the model. On the other hand, to make the model structures simpler, effective algorithms for generating simple neural network structures are needed. Such works can be the subjects of future studies.

## Acknowledgment

The research is supported by Science Foundation for Young Teachers of Northeast Normal University (NO20080202)

## References

- [1] J. Alvin Connelly, pyung choi, *Macromodeling with SPICE*. New Jersey: Englewood Cliffs Prentice-Hall, 1992, ch.1-ch.4.
- [2] *MicroSim PSpice A/D Reference Manual*, version 8.0, MicroSim Co., Irvine, CA, 1997, ch.2
- [3] NTC thermistors: standardized R/T characteristics, EPCOS Co., Munich, GERMANY, 2006
- [4] *Motorola Pressure Sensor Macromodel Library V1.0*, Motorola Inc. Semiconductor Products Sector, Sensor Products Division, Phoenix, USA, 2000
- [5] *Compound Coefficient Pressure Sensor PSPICE Models*, Freescale Semiconductor Co., Austin, TX, 2005
- [6] Meijer, "Fast and Smooth Highly Nonlinear Multidimensional Table Models for Device Modeling," *IEEE Transactions on Circuits and Systems*, 1990; 37, pp.335-346
- [7] Meijer, *Neural Network Applications in Device and Subcircuit Modelling for Circuit Simulation*, Ph.D. thesis, Eindhoven University of Technology, May 2, 1996
- [8] J. Martins, M. Fernandes, A. Fantoni, M. Vieira, "Spice model for a laser scanned photodiode tricolor image sensor," *Journal of Non-Crystalline Solids*, 2006; 352, pp.1813-1817
- [9] Bruce C. S. Chou, Yeong-Maw Chen, Mang Ou-Yang, Jin-Shown Shie, "A sensitive Pirani vacuum sensor and the electrothermal SPICE modeling," *Sensors and Actuators*, 1996; 53: pp. 273-277
- [10] J. Martins, M. Fernandes, M. Vieira, "Electrical simulation of a p-i-n image sensor," *Vacuum*, 2002;64, pp. 307-313.
- [11] Ali Ümit Keskin, Tolga Meriç Yanar, "Steady-state solution of loaded thermistor problems using an electrical equivalent circuit model," *Measurement Science and Technology*, 2004; 15, pp. 2163-2169.
- [12] Ali Ümit Keskin, "A simple Analog Behavioral Model for NTC thermistors including selfheating effect," *Sensors and Actuators-A*, 2005; 118-2, pp: 244-247
- [13] Timo Veijola, "Electrothermal Simulation Models for NTC and PTC Thermistors," *Proceedings of CSC'98*, 1998:2, pp 950-955
- [14] Satish Kumar, *Neural Networks*, New York: McGraw-Hill, 2006, pp.164-214
- [15] Ying Huang, Lian-ming Wang, "Neural Network Method for Modeling Nonlinear Electron Devices," *Chinese Journal of Electron Devices* 2005; 28: pp.890-892
- [16] Christophe Basso, "SPICE analog behavioral modeling of variable R-L-C," *Electronic Products China*, 2005; 9, Available at: <http://www.epc.com.cn/magazine/20060105/5486.asp>
- [17] István Petrás, Marco Gilli, "Complex dynamics in one-dimensional CNNs," *International Journal of Circuit Theory and Applications*, 2006; 34, pp: 3-20
- [18] Wang, Lian Ming; Ma, Lin Yun; Huang, Ying, "A General Neural Network-Based Approach to Modeling Sensors in PSPICE Simulation," *Third International Conference on Natural Computation*, 2007; 3, pp: 226-2