USER-DEFINED ELEMENT IN ZMAN™

How to add a new user-defined element?

Introduced by ZMAN 2.2, the "SIMPLE" category provides you with a functionality to easily make and edit your own circuit elements.

"BASIC" and "SIMPLE" categories are the sets of circuit elements. Each element is written in a form of analytical formula. Table 1 shows the elements of BASIC category. Its formula cannot be edited by a user. On the other hand, you can add and edit your own circuit elements under the category of SIMPLE. The other categories are for equivalent circuit models.

In this note, you can learn how to add a new user-defined element in SIMPLE category.

Preparations

Let's think about the following model:

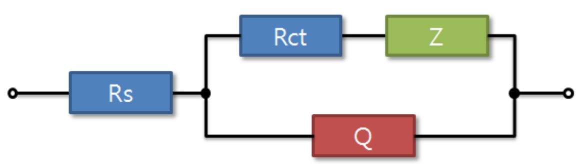


Figure 1. The model to be considered in this note is as shown here.

In the model shown in Figure 1, diffusion impedance element Z is defined as the following equation:

$$Z = \frac{R_W \coth[(T_W j\omega)^{W_P}]}{(T_W j\omega)^{W_P}} \qquad (1)$$

Whilst you will easily find Rs, Rct, and Q(Constant Phase Element) in the BASIC category, there is no such element describing the analytical function Z. Therefore, you need to make a new circuit element of your own as realizing eq. (1).

First, you may symbolize Z as an arbitrary but unused symbol; i.e. V.

Second, you must substitute suitable parameters for Rw, Tw, and Wp elements from eq (1):

Vr = Rw
Vt = Tw
Vp = Wp

It is due to the specific rule of naming parameters which is as the following in the box:

Parameters for circuit Element

ZMAN accepts only the following parameters:

A, AA, AB, AC, … , Aa, Ab, Ac, …
B, BA, BB, BC, … , Ba, Bb, Bc, …
…
Z, ZA, ZB, ZC, …, Za, Zb, Zc, …

- A parameter must consist of symbol and (or) an alphabetical character.
- A parameter name is case sensitive; i.e. AA and Aa are considered to be distinct parameters.

Then, the next step is to rephrase eq (1) into a formula acceptable in ZMAN, i.e. V equals to:

Vr * Coth(pow(Vt*s, Vp)) / pow(Vt*s, Vp)

where s is $j\omega$, Coth(x) is the hyperbolic cotangent of x, and pow(x, y) is the x raised to the power of y. Table 2 shows the list of functions you can use in formula.

Now, the model depicted in the Figure 1 can be described as the following expression.

Rs–(Rct–V)|Q

The rule for naming elements is as the following in the box.

Elements in Model

Variables are distinguished by names. There are the following rules of naming variables:

- The first character should be a Symbol character(A to Z) and followed by alphabetical characters (capital and small letters), decimal digits, and underscore (_)
- A name is case sensitive; i.e. lowercase and uppercase letters are considered to be distinct characters
- There is no limit of name length; herewith all characters are significant

Now, let's start to register V element to Model Editor.

Make a new element in SIMPLE category

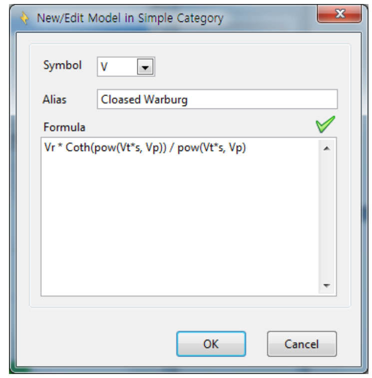
Complete the following steps to register a user-defined element in the model editor.

1. Launch ZMAN.
2. In the 'Getting Started' window, click the 'Model Editor'.

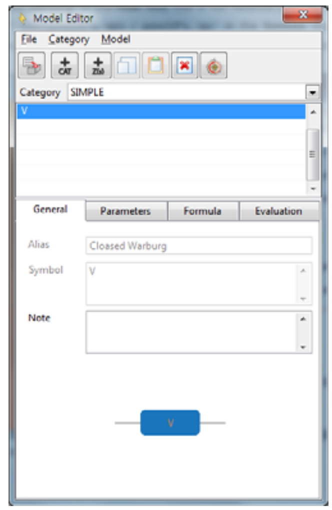
Note Use one of the following methods to access the Model Editor dialog box:

- Click the Model Editor link from the Getting Started Window after you launch ZMAN; or
- Select **Tools >> Model Editor...** from the menu bar of the ZMAN main window.

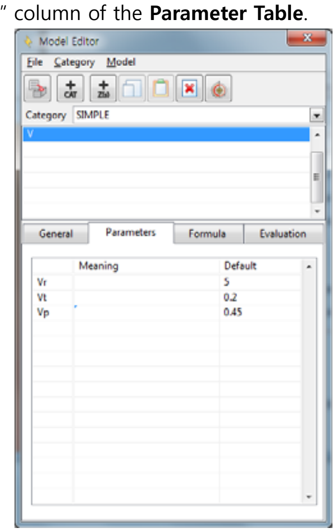
3. From the **Category** list, select SIMPLE. This displays its elements in the below list box.
4. Select **Model >> New/Edit...** to display "New/Edit Model in Simple Category" dialog.
5. From the **Symbol** list, select **V**. Disabled items means that they are used in BASIC category.
6. Type "Closed Warburg" in the **Alias** field. This is not necessary but for your convenience.
7. Type "**Vr * Coth(pow(Vt*s, Vp)) / pow(Vt*s, Vp)**" in the **formula** box. If the formula is acceptable, you can see green **Validation** checkmark and the **OK** button that is enabled.



8. Click the **OK** button to complete the steps. You can see V element is added in the **Element** list.



9. Click **Parameters** tab and type the suitable values in the "Default" column of the **Parameter Table**.



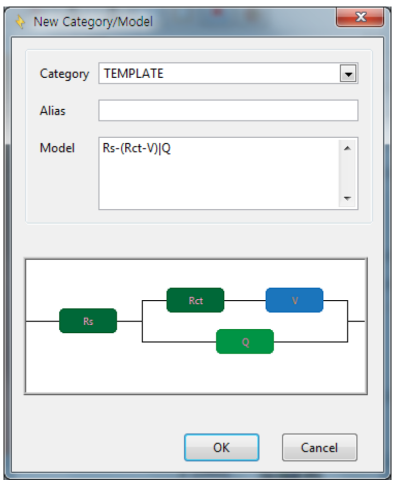
10. See the Nyquist and Bode Plot by clicking **Evaluation** tab.

Now we are ready to handle the model depicted in Figure 1.

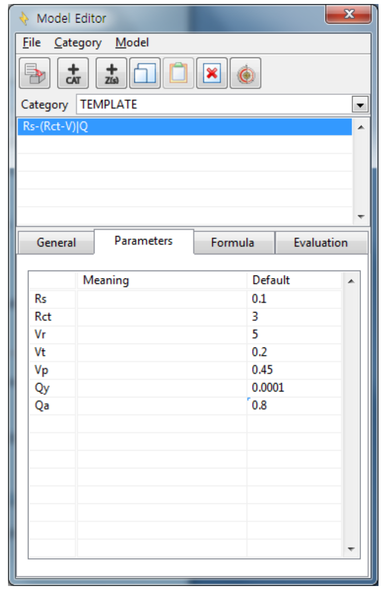
Add a new model in user category

Complete the following steps to make a user-defined model in the Model Editor.

1. From the **Category** list, select TEMPLATE. You can see its models in the below list. Select **Model >> New...** and the New Category / Model dialog is showed. If the TEMPLATE category is not existing, you can add it to the list by selecting **Category >> New...**. In this case, you can see the same dialog box.
2. Select "TEMPLATE" or type it in the **Category** box and "Rs–(Rct–V)|Q" in the **Model** box. The OK button should be enabled.



3. Click the **OK** button to come back to Model Editor. You may see the model is added in the Model Editor. To observe the behavior of the model, select **Parameters** tab and type suitable values in the table.



4. Check its Nyquist plot by clicking **Evaluation** tab. In order to have a close look, change frequency range from 0.1 Hz to 1 MHz and select **Matching Scales** in the right-click menu of the plot.

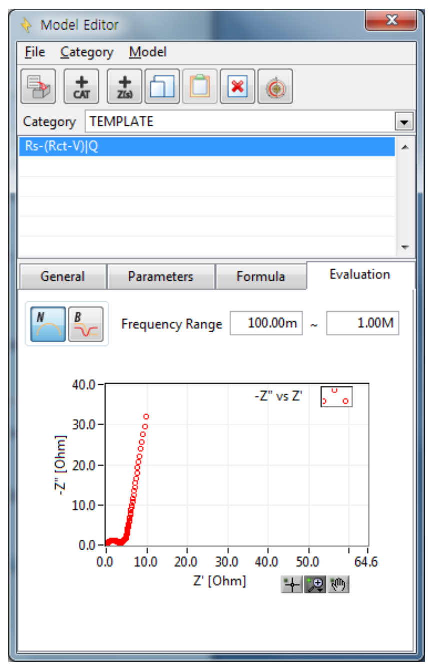


Table 1. Circuit elements defined in BASIC category are summarized.

Symbol	Description	Parameters	Formula	Note
R	Resistive Element	R	R	
C	Capacitive Element	C	$\frac{1}{sC}$	*1
L	Inductive Element	L	sL	
W	Warburg Diffusion	W	$\frac{1}{W\sqrt{s}}$	
Q	Constant Phase Element	Qy Qa	$\frac{1}{Q_y}\frac{1}{s^{Q_a}}$	
O	Nernst Impedance	Oy Ob	$\frac{1}{O_y\sqrt{s}}\tanh(O_b\sqrt{s})$	
T	Finite Diffusion	Ty Tb	$\frac{1}{T_y\sqrt{s}}\coth(T_b\sqrt{s})$	
G	Homogeneous Reaction (Gerischer)	Gy Gk	$\frac{1}{G_y\sqrt{G_k+s}}$	
S	Spherical Diffusion	Sy Sk	$\frac{1}{S_y}\frac{1}{\sqrt{S_k}+\sqrt{s}}$	
X	Finite-Length diffusion at planar particles	Xr Xc	$\sqrt{\frac{3X_r}{X_c s}}\tanh(\sqrt{3X_r X_c s})$	*2
Y	Finite-Length diffusion at spherical particles	Yr Yc	$\frac{\tanh(\sqrt{3Y_r Y_c s})}{\sqrt{\frac{3Y_c s}{Y_r}}-\frac{1}{Y_r}\tanh(\sqrt{3Y_r Y_c s})}$	*2
Z	Finite-Length diffusion at cylindrical particles	Zr Zc	$\frac{I_0(\sqrt{2Z_r Z_c s})}{\sqrt{2Z_r Z_c s}\cdot I_1(\sqrt{2Z_r Z_c s})}Z_r$	*2, *3

***1. Complex argument, $s = j\omega$, where imaginary unit, j equals $\sqrt{-1}$ and ω is angular frequency.**

***2. Impedance Spectroscopy: Theory, Experiment, and Applications, 2nd edition, Ed. E. Barsoukov, and J. R. Macdonal, John Wiley & Sons, Inc., Hoboken, New Jersey, 2005**

***3. $I_0(x)$ and $I_1(x)$ are modified Bessel-functions of the first kind, with 0 and 1 order correspondingly.**

Table 2. List of functions.

Constants	
pi	Returns pi
Functions	
abs(x)	computes the absolute value of x
acos(x)	inverse cosine of x
acosh(x)	inverse hyperbolic cosine of x
acot(x)	inverse cotangent of x
acoth(x)	inverse hyperbolic cotangent of x
asin(x)	inverse sine of x
asinh(x)	inverse hyperbolic sine of x
atan(x)	inverse tangent of x
atan2(x,y)	inverse tangent of x/y
atanh(x)	inverse hyperbolic tangent of x
ceil(x)	computes the smallest integer greater than or equal to x
cos(x)	cosine of x
cosh(x)	hyperbolic cosine of x
cot(x)	cotangent of x
coth(x) **Coth(x)**	hyperbolic cotangent of x Coth(x) is modified to avoid failure at limit condition of x
csc(x)	cosecant of x
csch(x)	hyperbolic cosecant of x
deg(x)	converts radians to degrees (x*180/p)
e(x)	returns e or the argument multiplied by e
erf(x)	Error function
erfc(x)	complementary Error function
exp(x)	e raised to the x power (exponential function)
factr(x)	Factorial of x
floor(x)	computes the largest integer less than or equal to x
fract(x)	computes the fractional part of x
gamma(x)	Gamma function
gammai(a,x)	Incomplete Gamma function
getexp(x)	computes the exponent of a floating-point value
getman(x)	computes the mantissa of a floating-point value
int(x)	computes the integer part of x
ldexp(m,e)	computes a floating-point number from mantissa and exponent
ln(x)	natural logarithm of x (logarithm to the base e)
log(x,y)	logarithm of y to the base x
log10(x)	logarithm of x to the base 10
log2(x)	logarithm of x to the base 2
pi(x)	returns pi or the argument multiplied by pi
pow(x,y)	x raised to the y power
pow10(x)	10 raised to the x power
pow2(x)	2 raised to the x power

rad(x)	converts degrees to radians (x*p/180)
random(x,y)	generates random numbers within the specified range
sec(x)	secant of x
sech(x)	hyperbolic secant of x
sign(x)	returns the sign of x
sin(x)	sine of x
sinc(x)	sin(x)/x
sinh(x)	hyperbolic sine of x
spike(x)	Spike function
sqrt(x)	computes the square root of x
square(x)	square function
step(x)	step function
tan(x)	tangent of x
tanh(x) **Tanh(x)**	hyperbolic tangent of x Tanh(x) is modified to avoid failure at limit condition of x
Special Functions	
BesselJ(n,x)	Bessel function of the first kind, denoted as $J_n(x)$
BesselY(n,x)	Bessel function of the second kind, denoted as $Y_n(x)$
BesselI(n,x)	modified Bessel function of the first kind, denoted as $I_n(x)$
BesselK(n,x)	modified Bessel function of the second kind, denoted as $K_n(x)$