

## “Advanced Programmable Load” VTB Model

Author: Eric Vilar

Date: 8/16/2003

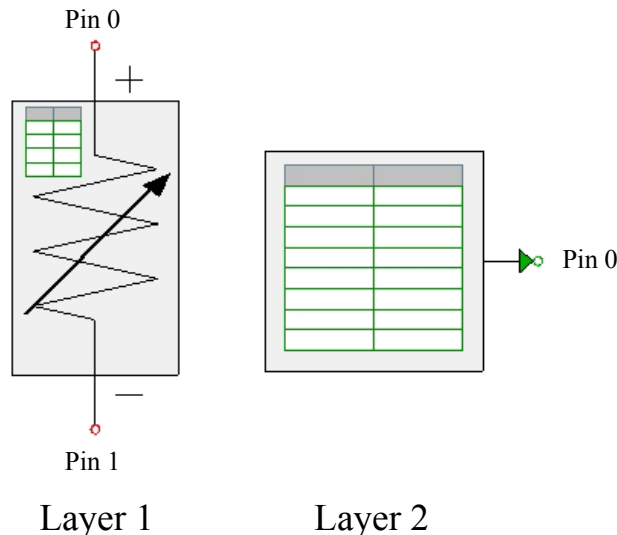
Model name: Advanced Programmable Load

DLL name: AdvProgLoad\_030816.vtm

Version number: 1.1

Report errors or changes to: vilar@engr.sc.edu

### Pictorial Representation of Model



### Brief Description of Model

- When natural coupled (layer 1) this model functions as a voltage (pressure, speed, ect.), current (mass flow, torque, ect.), resistive, or power load.
- When signal coupled (layer 2) this model functions as a signal generator that can be used for controlling signal dependent source models.
- The load is time dependent time dependent, and it is characterized by a look up table in an Excel spread sheet.
- The model reads values directly off the Excel spread sheet rather than from a saved data file.
- Optionally, the model can communicate directly with VBA functions in Excel to customize the loads behavior.

### Model Validity Range and Limitations

- When the model is used as a power load, the system that the model is attached to must be able to supply the power that the model is trying to dissipate. If the system cannot supply the specified amount of power, the simulation will fail because the numerical solution will not converge.
- This VTB model will only operate in a Microsoft Windows environment and Excel 2000 or Excel Xp (Excel 2002) must be installed.
- The model cannot read from the Excel spreadsheet if the Excel application is busy (i.e. a menu is open or a cell is being updated and the cursor is active in the function window). If the Excel application is busy and the model attempts to access it, an error will occur and force VTB to shut down.

## List of Model Pins with Connectivity Information

### Layer 1

Pin Designation	Description
0	Natural coupled, positive node
1	Natural coupled, negative node

### Layer 2

Pin Designation	Description
0	Signal node, output

## List of Parameters and Output Variables

This is a complete list of all parameters of the model. All models use SI units.

Parameter Name	Description	Default Value	Units
Excel workbook and location	The name and location of the Excel workbook file containing the load profile.	“C:\”	none
Worksheet	Excel worksheet within the Excel workbook that the model reads from.	“Profile”	none
Time Position	The starting column and row that the model begins reading time data from when the simulation begins.	“A4”	none
Load Position	The starting column and row that the model begins reading operating point data from when the simulation begins.	“B4”	none
Enable Initialize Function	If selected, the model will execute a macro in the workbook called “VTBtoVBA_Init” each time a simulation is executed. See below about Excel security settings.	False (unchecked)	none
Enable Step Function	If selected, the model will execute a macro in the workbook called “VTBtoVBA_Step” for each discrete time step in the simulation. See below about Excel security settings.	False (unchecked)	none
Load Type	Specifies the operating point the model will control: voltage, current, power, power (dual), resistance, or signal.	“Power”	V, A, W, $\Omega$
Time Constant	Specifies the first order transient response time between different load levels. Approximately five time constants elapse before the new load level is reached. If zero the load switches to the steady state value in one time step.	0	seconds
Initial Branch	When operating as a power load, the model’s numerical	$10 \times 10^6$	$\Omega$

Resistance (power loads only)	method uses past history values to calculate new values. On the first time step, the model behaves as a simple resistor, and this initial resistance is specified by this parameter. Typically large values work well because the load to appear as an open circuit in the system.		
Branch Resistance (current and power (mod.) loads only).	For current load operation, the branch resistance should be much, much larger than the current value. For the dual power load, a small resistance values causes the load to draw more current, and a high resistance causes the load to draw less current.	10e6	$\Omega$
Stop simulation if the load supplies power (voltage and current loads only).	When operating as a voltage or current load, the model is really just a voltage or current source. Check this option to stop the simulation if the load enters the active region.	False (unchecked)	none

This is a list of output variables.

Variable Name	Description	Units
Voltage	Voltage across the model, natural coupled only	volts
Current	Current through the model, natural coupled only	amperes
Power	Power dissipated in the model, natural coupled only	watts
Resistance	Proportionality between voltage and current, natural coupled only	ohms
Signal	Output value (real number), signal coupled only	none (real number)
Target steady state load level	The operating point (it the system can support it) that the model will eventually arrive at.	volts, amperes, watts, ohms, real number
Target transient load level	The operating point that the model should converge at for each corresponding time step. At steady state the target load level and transient load level are equal.	volts, amperes, watts, ohms, real number

## Mathematical Description of Model

### ***Current (Mass Flow, Torque, ect.) Load***

In current mode, the model is simply a current sink, which dissipates power rather than produces power like a current source. Using the resistive companion method, the current sink,  $I_s$ , is attached to a branch in parallel with the branches resistance,  $R_b$ , as shown in Figure 1.

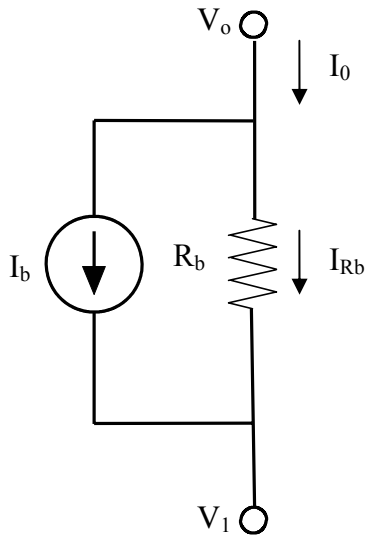


Figure 1. Topology of a current sink using the resistive companion method.

The user specified parameters are  $I_b$  and  $R_b$ . The amount of current into the load is

$$I_0 = I_b + I_{Rb} = I_s + \frac{V_0 - V_1}{R_b} \quad (1)$$

To make the model behave ideally such that  $I_0$  equals  $I_b$ , make  $R_b$  very large so that  $I_{Rb}$  is very small.

### **Voltage (Pressure, Speed, ect.) Load**

In voltage mode, the model is an ideal voltage source using modified nodal analysis. The model will force the specified voltage across its terminals. The voltage mode topology is shown below in Figure 2.

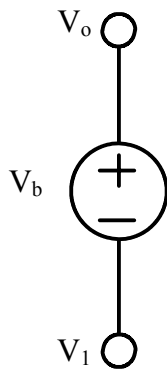


Figure 2. Topology of a voltage source using modified nodal analysis.

The user specifies the value of the voltage source,  $V_b$ , which specifies the voltage difference between node voltages  $V_0$  and  $V_1$ :

$$V_b = V_0 - V_1. \quad (2)$$

#### **Warning!**

This model should not be connected directly in parallel with other voltage source models; instead, a resistance should be attached between them. Many voltage source models include a series resistance inside the voltage model, and when this is the case a series resistance does not need to be inserted.

## Resistive Load

As a resistive load, the model behaves according to Ohm's Law.

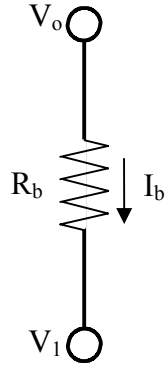


Figure 3. Topology of a resistor.

The current through the resistive load is related to the voltage across it by

$$I_b = \frac{V_0 - V_1}{R_b}. \quad (3)$$

## Power Load

In power mode, the model behaves as a non-linear resistor where the current into the model is a function of voltage:

$$I_b = \frac{P}{V_b}. \quad (4)$$

The current into the load is inversely proportional to the voltage across the load by an amount of power that is specified by the user. Because the model is non-linear, several iterative calculations may be needed by the VTB solver in order to get the system level solution to converge. The Newton-Raphson iteration method is used to model the above relationship as

$$I_b^{n+1} = -\left(\frac{P}{(V_b^n)^2}\right)V_b^{n+1} + 2\frac{P}{V_b^n}, \quad (5)$$

where  $n$  denotes the  $n^{\text{th}}$  iteration of the solution process. Note that as the system converges  $V_b^{n+1} = V_b^n$ , and the relationship simplifies back into (4).

### Initial conditions.

On the very first iteration of the first time step  $V_b^n$  is unknown; therefore, the model behaves as a simple resistor. If a *relatively* large value of resistance is used, which is recommended, the load will appear as an open circuit and dissipate an insignificant amount of power. After the first iteration, the model functions as a power load.

### Warning!

This version of the power load will cause the system to appear to fail when it is connected in series with a current source. The current output by the current source will not equal the amount specified. This apparent failure occurs because the load model forces a reverse current into the current source through the current sources branch resistance. The dual power load will work when connected in series with a current load.

The amount of power dissipated in the load must be greater than zero. If a value equal to or less than zero is specified, the model will default to a power of 1mW.

## Power Load (dual)

This version of the power load uses modified nodal analysis to design the dual implementation of the original power load above. The internal voltage of the voltage source is

$$V_i = \frac{P}{I_b}, \quad (6)$$

where the user defined parameter P is the total power dissipated in the branch resistance and in the internal voltage source combined. Using the Newton-Raphson iteration method, the relationship becomes

$$V_b^{n+1} = -\left(\frac{P}{(I_b^n)^2}\right)I_b^{n+1} + 2\frac{P}{I_b^n}, \quad (7)$$

where n denotes the n<sup>th</sup> iteration of the solution process. Note that as the system converges  $I_b^{n+1} = I_b^n$ , and the relationship simplifies back into (6).

*Initial conditions.*

On the very first iteration of the first time step  $V_b^n$  is unknown; therefore, the model behaves as a simple resistor. If a *relatively* large value of resistance is used, which is recommended, the load will appear as an open circuit and dissipate an insignificant amount of power. After the first iteration, the model functions as a power load.

*Warning!*

This version of the power load will cause the system to not converge when connected in parallel with an ideal voltage source. The failure occurs because the load and the voltage source will try and specify two different load levels across the same set of nodes.

## Transient Response

There is no transient response if the time constant is zero. Otherwise, the model's operating load,  $L$ , is

$$L(t) = L(t_0) + (L(t_f) - L(t_0)) \left(1 - e^{-\frac{t-t_0}{\tau}}\right). \quad (8)$$

The load has units of A, V, W,  $\Omega$ , or equivalent. In signal mode, the load output is just a numerical value.

$L(t_0)$  is the load's value at the time when the load begins a transition to a new steady state load value

$L(t_f)$ .

## Example of Model Use

### Creating the Load Profile Look Up Table in Excel

The 'load profile' is a look up table written in Excel that expresses the load's operating point as a function of time. An example load profile spreadsheet is given in Figure 4 below. In the example time (in seconds) is entered in column "A" of the Excel spreadsheet, and the corresponding load values are entered in column "B". By default, the model will read data beginning at row 4, which leaves room to give some information about the load profile. The example in figure 2 below uses row 1 to identify the load profile as an "example input spreadsheet for the Advanced Programmable Load model." The third row is used for labeling the column headings 'Time(s)' and 'Load(W).'

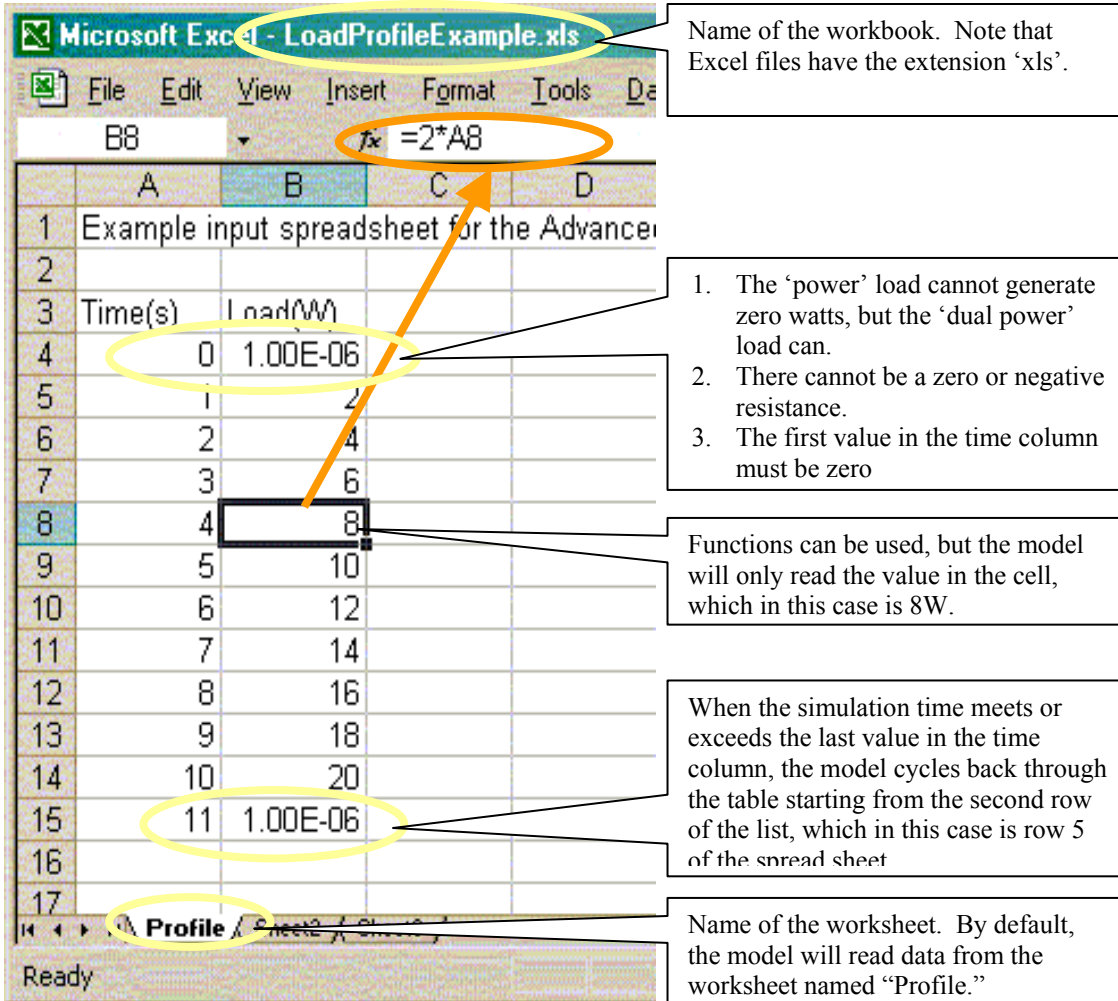


Figure 4. The load profile example written in Excel that is referenced many times throughout this document.

### Interpreting the Load Profile Lookup Table

Let the values in the time column be labeled  $T_i$  and the values in the load column be labeled  $L_i$ . For a given simulation time  $T$  such that  $T_a < T < T_b$ , the model determines the load's operating point  $O$  to be  $L_a$ . For the example profile in figure 2 above, if  $T$  is 2.5 seconds, which is greater than  $T_1$  with a value of 1 second and less than  $T_2$  with a value of 2 seconds, then the load's operating point is  $L_1$ , which equals 2W. If the simulation time exceeds the greatest value in the time column,  $T_n$ , then the model adjusts the simulation time by dividing the simulation time  $T$  by  $T_n$ , and uses the remaining time  $T_{adj}$  to find the load's operating point,  $O$ . For example, if  $T$  is 13.5 seconds, then  $T_{adj}$  is 2.5 seconds and the operating point is 2W. If  $T$  is equal to  $T_n$ , then the load's operating point is  $L_n$ .

### Excel Security: Enable Macros and Access to the Visual Basic Project (Optional)

By default, security on Excel is set to 'high' to keep Excel from executing potentially malicious macros that could compromise your computer. In 'high' security mode, only certified macros from sources that you approve will be executed. Currently, none of the macros associated with this program are certified. If you want to use custom macros that can be accessed by the Advanced Programmable Load model, then the security mode must be set no higher than 'medium.' To adjust the security setting, go to *Tools/Macro/Security* and select 'medium.' If you are using Excel Xp, access to the Visual Basic Project

must also be enabled: on the menu bar under *Tools/Macro/Security/Trusted Sources* select *Trust access to Visual Basic Project*.

## Using Custom Sub Procedures and Functions in VBA (Optional)

Optionally, the Advanced Programmable Load can initiate VBA sub procedures and functions linked to the Excel spread sheet. There are two options available: *VTBtoVBA\_Init* and *VTBtoVBA\_Step*. The first option, *VTBtoVBA\_Init*, is a sub procedure so it does not return a value. The model calls it whenever a VTB simulation begins so that with each new simulation the spread sheet can be programmed to generate a new set of data. The syntax for this function is:

```
Public Sub VTBtoVBA_Init()  
  
    Add VBA code here  
  
End Sub
```

Figure 5. The proper syntax for the *VTBtoVBA\_Init* sub procedure.

The second option, *VTBtoVBA\_Step*, is a function that receives state variable parameters from the model and returns to the model the target steady state load level. The syntax for this function is

```
Function VTBtoVBA_Step ( _  
    dCurrentEventTime As Double, _  
    dNextEventTime As Double, _  
    dCurrentEventValue As Double, _  
    dNextEventValue As Double, _  
    dSimTime As Double, _  
    dTimeStep As Double) As Double  
  
    Add VBA code here  
  
End Function
```

Figure 6. The proper syntax for the *VTBtoVBA\_Step* sub function.

The event variables in the function parameter list correspond exactly to the rows in the spreadsheet look up table. For example, the lookup table in Figure 4 above has 11 different events that occur at specific times and cause specific operating point values (Watts in this case). The current event time (*dCurrentEventTime*) is the closest time value in the look up table that is less than or equal to the current simulation time (*dSimTime*). The current event value (*dCurrentEventValue*) is the load's operating point specified on the same row of the table. The time value on the next row in the table is the next event time (*dNextEventTime*) and the operating point value is the next event value (*dNextEventValue*).

Whenever the last event time in the table is exceeded, the model restarts the event list beginning with the second row in the table, which is row five in the spreadsheet given in Figure 4. When a new cycle starts, the event times are offset by the amount of simulation time that has elapsed. To make programming easier inside *VTBtoVBA\_Step* function, the current event and next event parameter values are already adjusted by the offset before they are passed through the parameter list.

## Custom Function Tutorial: Linear Interpolation

Open the Excel workbook *LoadProfileExample.xls*. If you do not have the workbook, create Inside Excel go to *Tools/Macro/Visual Basic Editor*, the Visual Basic Editor will open. In the project window on the left hand side of the Visual Basic Editor right-click the label *VBAProject("LoadProfileExample.xls")* and choose *insert/module*. Next, copy the block of code in Figure 6 and paste it into the new blank module



window that opened in the previous step. Copy the block of code in Figure 7 and paste it into the function over the text 'Add VBA code here.' Before running the VTB simulation, open the Advanced Programmable Load model and select (check) the option to enable the *Step* function.

```
Dim dSlope As Double
dSlope = (dNextEventValue - dLastEventValue) / (dNextEventTime -
dLastEventTime)
VTBtoVBA_Step = dLastEventValue + _
```

Figure 7. This block of code can be used in the *VTBtoVBA\_Step* function to linearly interpolate the operating point of the load between events.

Graphical results

## Model Validation

In below Figure 8, the Advanced Programmable Load's load profile is identical to the profile shown in Figure 4 above. According to the profile, the load should increase stepwise by 2W for every second from 0 (1uW) to 20W. However, the maximum power that can be delivered to the load is 12.5W, which occurs when the input resistance,  $R_0$ , and the load resistance are equal. Therefore, the model's power demand can only be met for the first seven seconds. After seven seconds, the power demand goes up to 14W, which exceeds the amount of power that can be delivered to the load by the voltage source. Figure 9 shows the parameter setup for the Advanced Programmable Load model and Figure 10 shows the plot power dissipated in the load versus time.

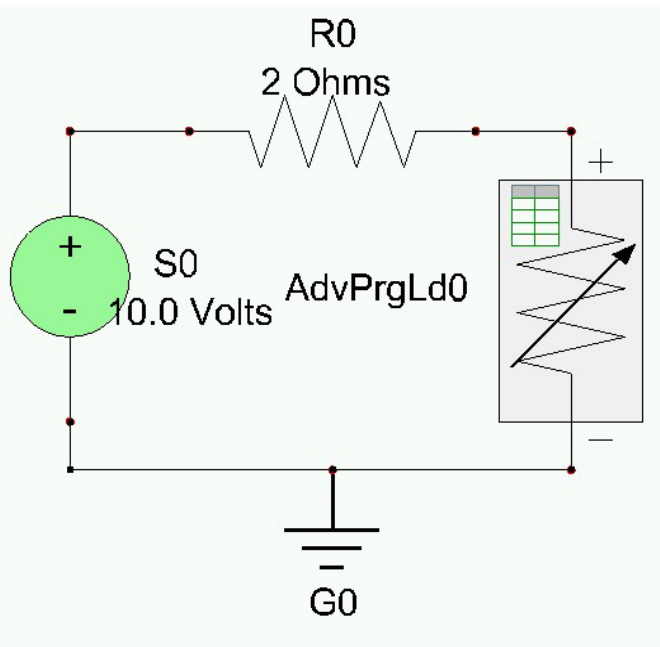


Figure 8. Circuit used to demonstrate the model.

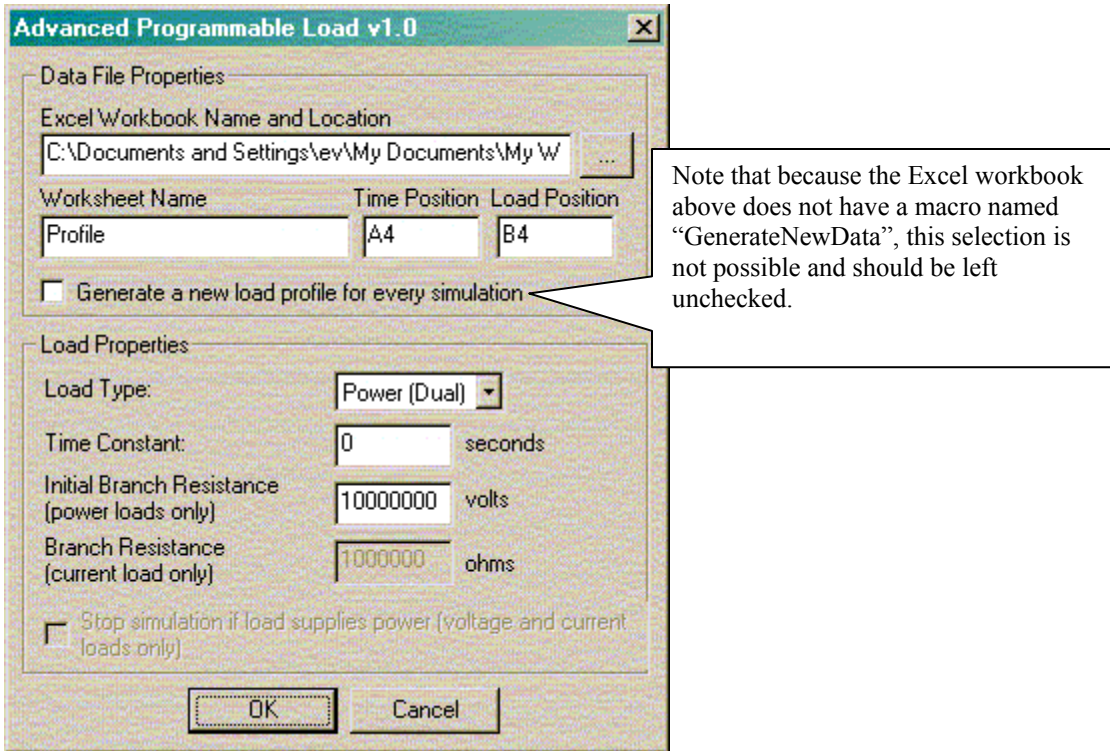


Figure 9. The model's properties used for validating the model as a power load.

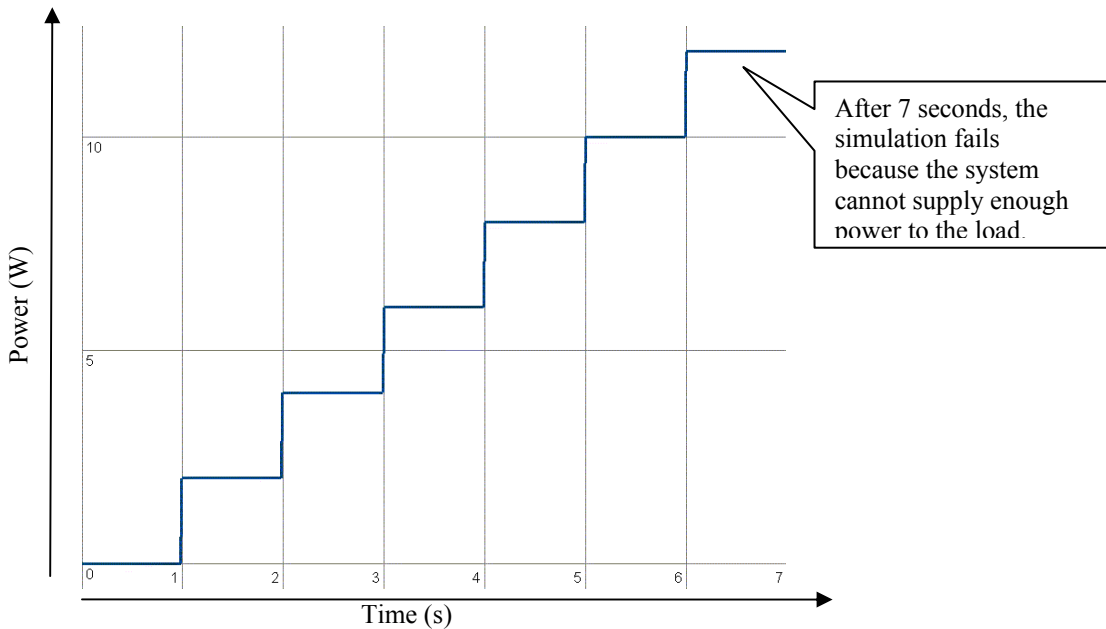


Figure 10. Plot of power delivered to the load versus time.

## Transient Response

For this demonstration, the system remains identical to the system given in Figure 8, except that the time constant parameter is changed from 0 seconds to 0.1 seconds. Figure 12 is a plot of the power dissipated in

the load versus time. It shows that the load has a first order time dependence, and that it requires approximately five time constants (0.5 seconds) to reach its steady state operating point. Figure 11 shows the parameters used to setup the load model.

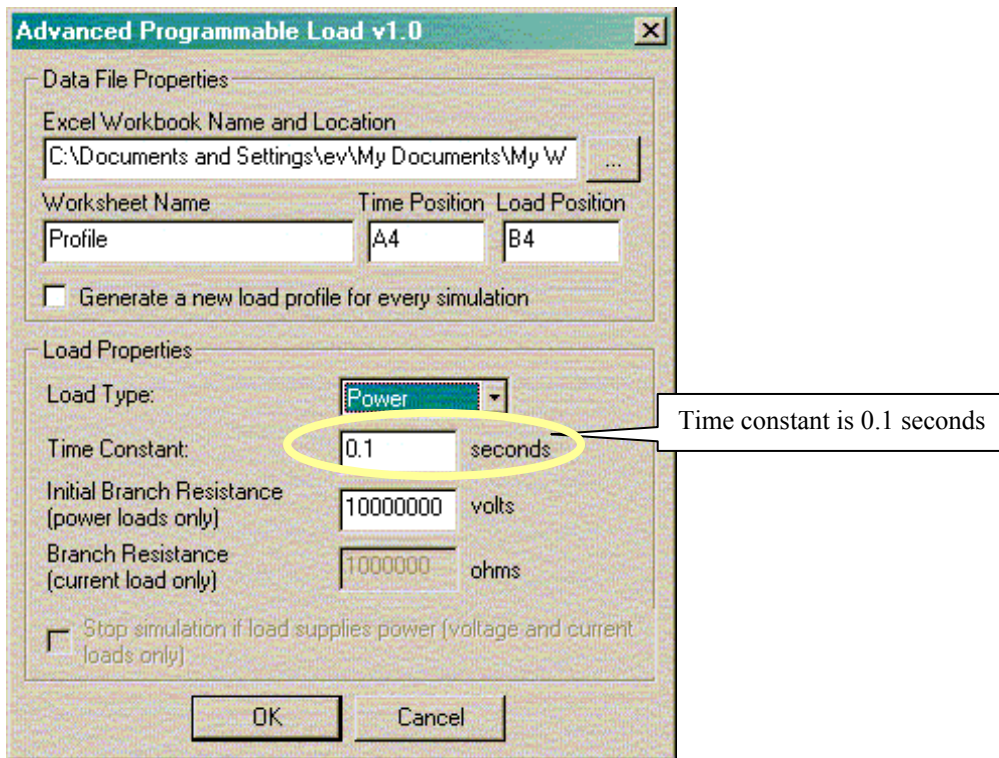


Figure 11. The model's load properties for the transient response demonstration.

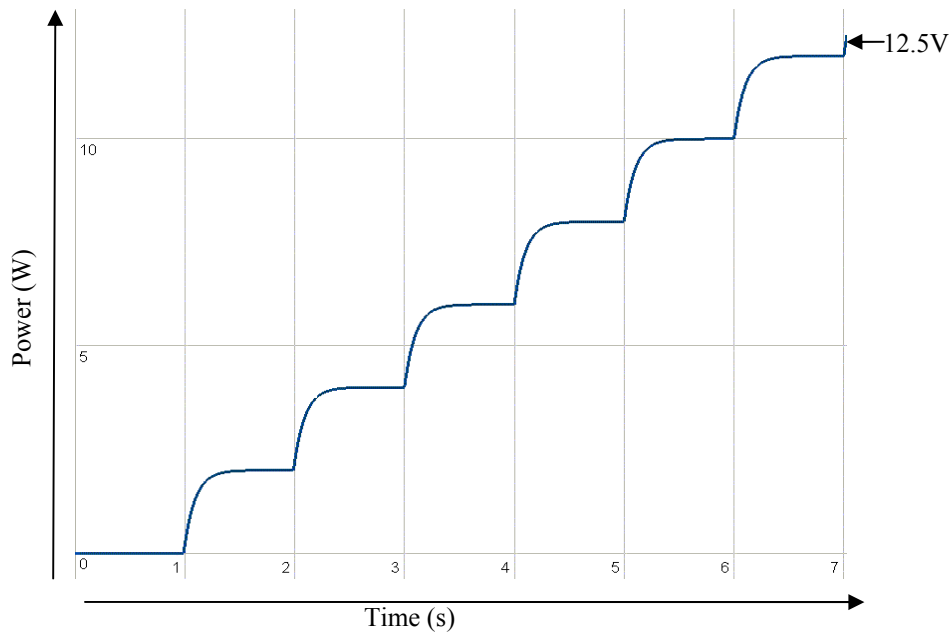


Figure 12. Plot of power dissipated in the load versus time that demonstrates the transient time response.

## Custom Functions Results: Linear Interpolation

In this test, the resistance between the source and the load in the demonstration model from Figure 8 is reduced so that all of the power demanded by the load can be delivered. The load profile is identical to the example in Figure 4, and the VBA function is added to the Excel workbook following the procedure in the section *Custom Function Tutorial: Linear Interpolation*. Note that at 14 seconds the operating point of the load is 2 W because the model has cycled back to the second row of the lookup table.

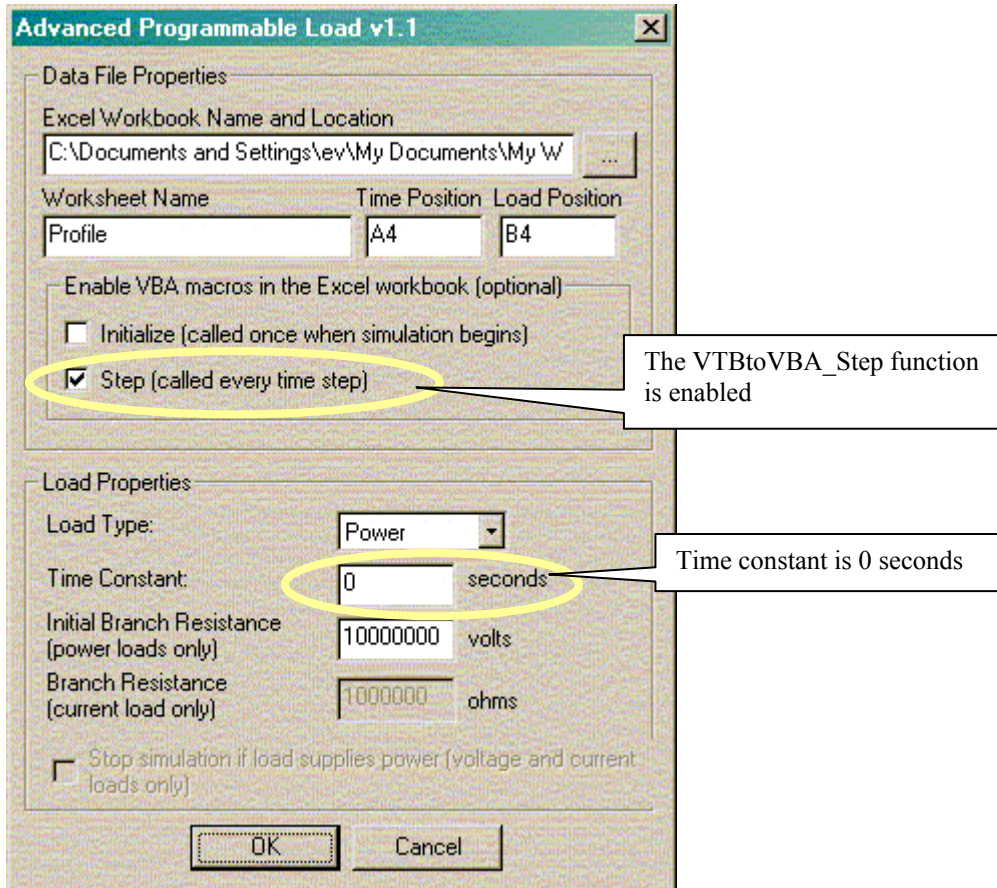


Figure 13. Model properties used when testing the custom linear interpolation function.

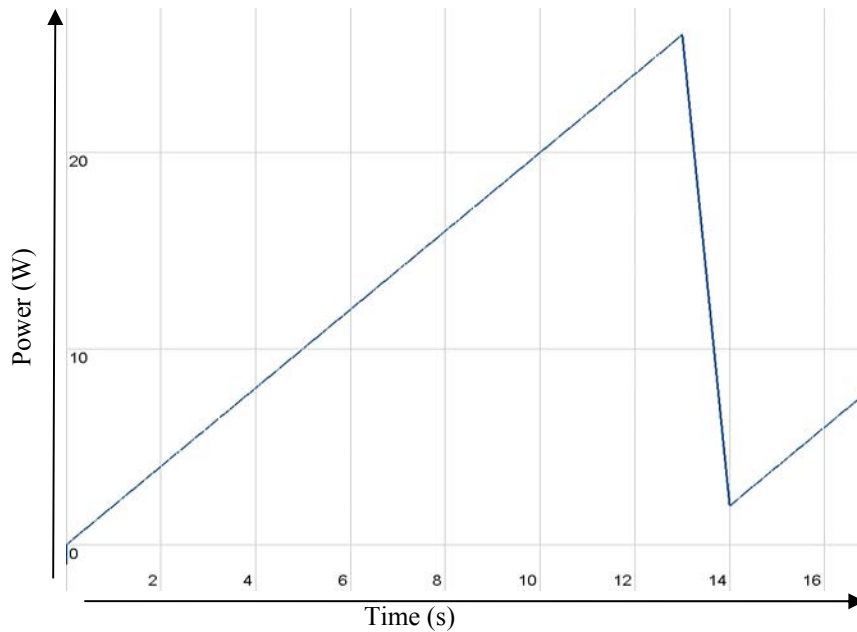


Figure 14. Output plot of power dissipated in the load versus time while using the custom linear interpolation function.

```

Function VTBtoVBA_Step ( _
    dCurrentEventTime As Double, _
    dNextEventTime As Double, _
    dCurrentEventValue As Double, _
    dNextEventValue As Double, _
    dSimTime As Double, _
    dTimeStep As Double) As Double

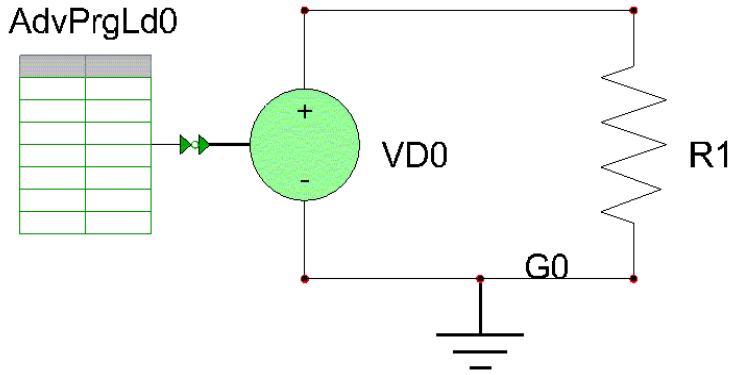
    Dim dSlope As Double
    dSlope = (dNextEventValue - dLastEventValue) /
        (dNextEventTime - dLastEventTime)
    VTBtoVBA_Step = dLastEventValue + _
        dSlope * (dSimTime -
            dLastEventTime) / (dNextEventTime - LastEventTime)

```

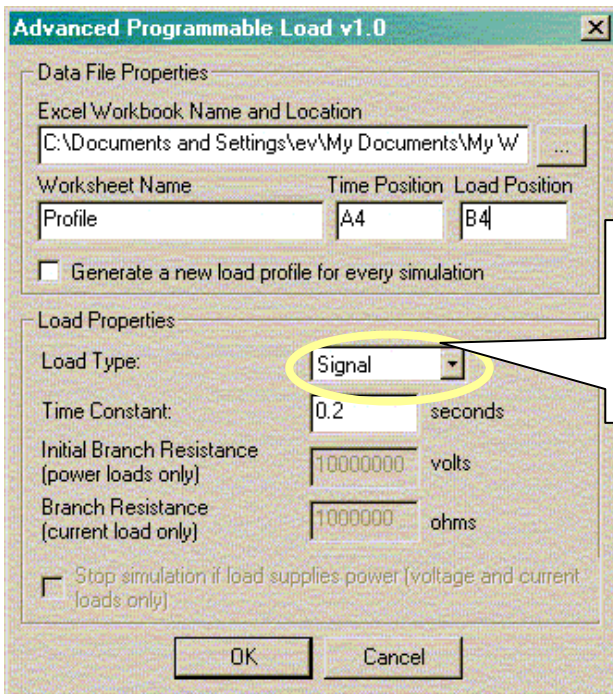
Figure 15. The code for implementing linear interpolation with the VTBtoVBA\_Step function, which is located in the VBA project attached to the Excel workbook.

## Signal Generator

In signal mode the model can be used to drive signal dependent sources. Figure 16 below shows the Advanced Programmable Load driving a signal dependent voltage source. The parameters used to setup the model are given in Figure 17, and the load profile is identical to the one shown in Figure 4 previously. Figure 18 shows the plot of voltage versus time for the signal dependent voltage source. Note that at time zero, the voltage of the signal load is incorrect. The voltage should be 1 $\mu$ V as specified in the load profile, but instead the voltage is 15.9857V. This occurs because the solver first solves the system for all naturally coupled models before allowing the signal coupled models to communicate. In general, mixed coupled models such as the signal dependent voltage source cannot be initialized by a signal stream.



al mode.



Switch the operation parameter to 'signal,' and the load will output a signal stream which can be connected to drive signal dependent source models. The icon changes automatically.

Figure 17. Parameters used to setup the model as a signal generator.

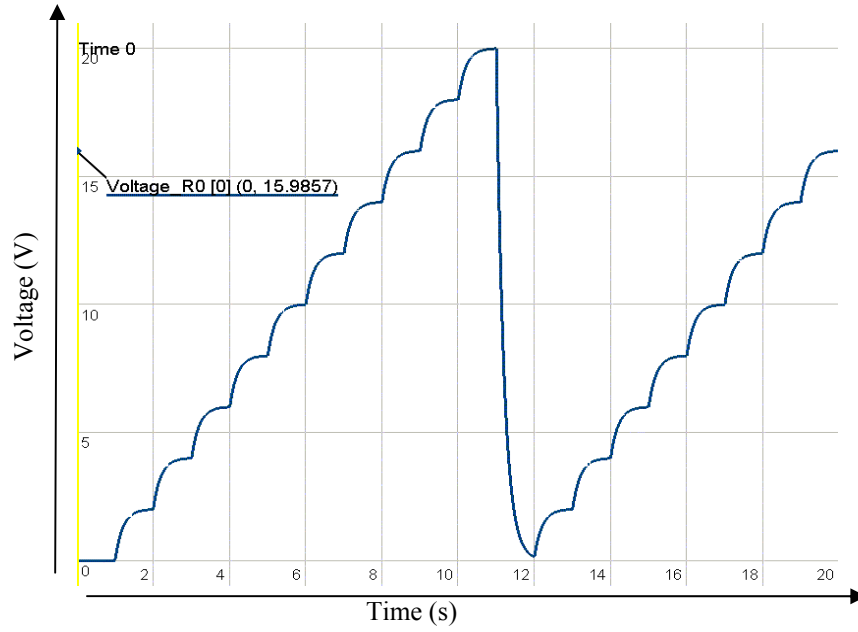


Figure 18. Plot of the voltage output versus time for the signal dependent voltage source.

## References

None.