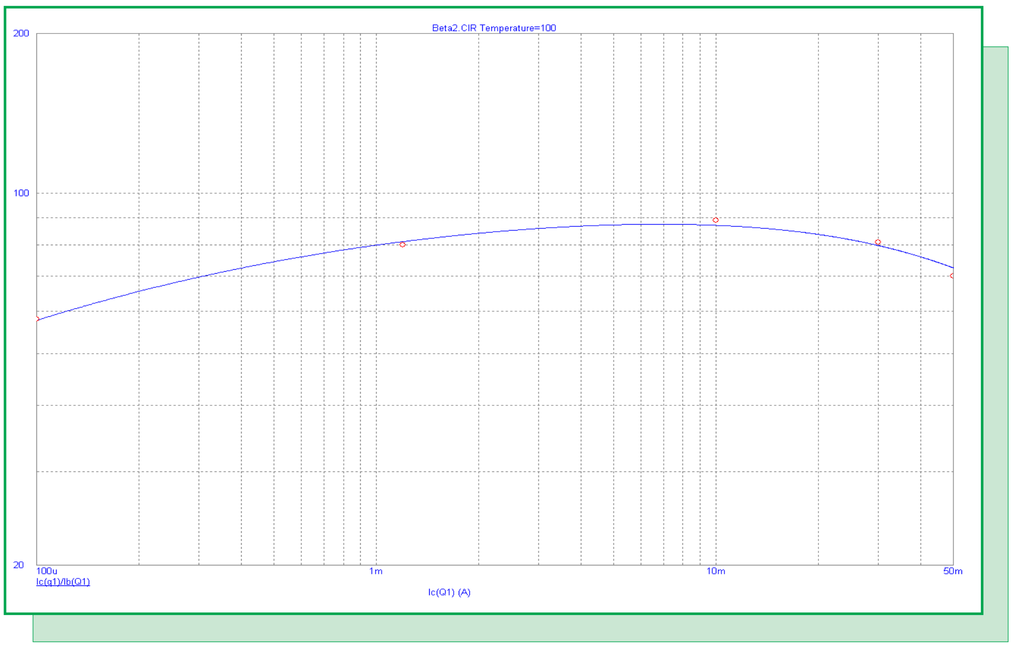
spectrum news

Applications for Micro-Cap™ Users

Spring 2006

News



BJT Temperature Modeling

Featuring:

- Network Installation Troubleshooting
- Modeling Impedance with Tabular Data in AC Analysis
- BJT Temperature Modeling

News In Preview

This newsletter's Q and A section describes how to disable the Text Increment feature where text node names are incremented during a copy and paste operation. The Easily Overlooked Feature section describes the Reduce Data Points feature available in the analysis mode which lets the user reduce the amount of data that Micro-Cap stores for analysis plots which will increase the speed of plot redraws.

The first article describes the steps that can be taken to debug a security issue with a network installation of Micro-Cap.

The second article describes how to model an impedance element in an AC simulation when the impedance information available is tabular data that describes the magnitude and phase of the impedance versus frequency.

The third article describes how to optimize the XTB and XTI parameters for a BJT model in order to more accurately model the beta curves when the temperature of an analysis varies.

Contents

Book Recommendations

General SPICE

• *Computer-Aided Circuit Analysis Using SPICE*, Walter Banzhaf, Prentice Hall 1989. ISBN# 0-13-162579-9

• *Macromodeling with SPICE*, Connelly and Choi, Prentice Hall 1992. ISBN# 0-13-544941-3

• *Inside SPICE-Overcoming the Obstacles of Circuit Simulation*, Ron Kielkowski, McGraw-Hill, First Edition, 1993. ISBN# 0-07-911525-X

• *The SPICE Book,* Andrei Vladimirescu, John Wiley & Sons, Inc., First Edition, 1994. ISBN# 0-471-60926-9

MOSFET Modeling

• *MOSFET Models for SPICE Simulation, William Liu, Including BSIM3v3 and BSIM4*, Wiley-Interscience, First Edition, ISBN# 0-471-39697-4

VLSI Design

• *Introduction to VLSI Circuits and Systems,* John P. Uyemura, John Wiley & Sons Inc, First Edition, 2002 ISBN# 0-471-12704-3

Micro-Cap - Czech

• *Resime Elektronicke Obvody,* Dalibor Biolek, BEN, First Edition, 2004. ISBN# 80-7300-125-X

Micro-Cap - German

• *Schaltungen erfolgreich simulieren mit Micro-Cap V,* Walter Gunther, Franzis', First Edition, 1997. ISBN# 3-7723-4662-6

Micro-Cap - Finnish

• *Elektroniikkasimulaattori,* Timo Haiko, Werner Soderstrom Osakeyhtio, 2002. ISBN# ISBN 951-0-25672-2

Design

• *Microelectronic Circuits High Performance Audio Power Amplifiers,* Ben Duncan, Newnes, First Edition, 1996. ISBN# 0-7506-2629-1

• *Microelectronic Circuits.,* Adel Sedra, Kenneth Smith, Fourth Edition, Oxford, 1998

High Power Electronics

• *Power Electronics,* Mohan, Undeland, Robbins, Second Edition, 1995. ISBN# 0-471-58408-8

• *Modern Power Electronics,* Trzynadlowski, 1998. ISBN# 0-471-15303-6

Switched-Mode Power Supply Simulation

• *SMPS Simulation with SPICE 3,* Steven M. Sandler, McGraw Hill, First Edition, 1997. ISBN# 0-07-913227-8

• *Switch-Mode Power Supply SPICE Simulation Cookbook*, Christophe Basso, McGraw-Hill 2001. This book describes many of the SMPS models supplied with Micro-Cap.

Micro-Cap Questions and Answers

Question: I have created a circuit section that I would like to copy and paste a few times in the schematic. Since each of these sections are operating in parallel with each other, I have labelled the nodes on each side of the section as PortA and PortB. However, when I copy and paste the section, the text labels for the nodes in the new pasted section have numbers appended to them such as PortA1 and PortB1. I then have to manually edit each of the text labels to maintain the parallel connectivity. Is there a way to prevent Micro-Cap from automatically changing these text labels?

Answer: Micro-Cap does have an option available that can disable this feature. Select Preferences under the Options menu. In the Preferences dialog box that comes up, click on the Options tab. In the Circuit section of the Options page, there is an option called Text Increment. Disable this option. Now when text is pasted into the schematic, the text label will maintain its original string.

The Text Increment option is designed to increment any grid text that is being pasted into the circuit if there is existing text within the circuit that matches the string. This option is extremely useful when labelling nodes in the following manner:

Out1
Out2
...
Out16

such as with the output of a 16 bit AtoD component. In this case, simply place a text string called Out1 into the schematic. Select the Out1 text string. Hold the CTRL key down and drag on the Out1 text. This CTRL+Drag procedure performs an operation called drag copying. Drag copying copies the selected objects and drags the copy along with the mouse, leaving the original objects undisturbed. When Text Increment is enabled, the drag copying will also automatically increment the text so it is a very simple procedure to quickly create and place node labels such as these.

If you do want to maintain the connectivity of sections via node names that you are pasting into the schematic, then this option should be disabled.

Easily Overlooked Features

This section is designed to highlight one or two features per issue that may be overlooked because they are not made visually obvious with a toolbar button.

Reduce Data Points Dialog Box
For some simulations, a very small timestep is used to obtain a high degree of accuracy. However, all of the data points calculated may not be necessary for the visual inspection of the waveform. Having a large amount of data points in the simulation will increase the redraw times of the plot when going between cursor and scope modes, adding text, adding tags, or any other operation that requires a redraw. The Reduce Data Points option provides a method to delete a portion of the calculated data points in order to help speed up the redraw operation. This option is available under the Transient, AC, or DC menus, and the corresponding dialog box appears below.

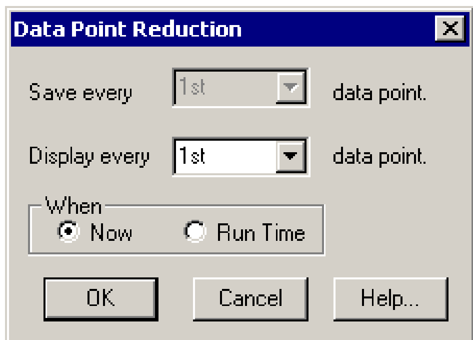


Fig. 1 - Reduce Data Points dialog box

This dialog box lets you discard every n'th data point in a save or display operation. Note that if you reduce data points for a save operation, it will also use the display reduce figure when displaying it during a retrieve operation. For example, if both the Save and Display are set to keep every tenth point, performing a save and then retrieving the file without changing either of these will produce an analysis showing every hundredth point. If a DSP operator is used in conjunction with either of these methods, then they are forced to use every data point, and any changes in this dialog box will not have an affect. An FFT window will use the reduced data point set though.

Save every n'th data point: This list box controls the percentage of data points saved in the file when the Run Options is set to Save or a probe analysis is run.

Display every n'th data point: This list box controls the percentage of data points that are plotted when the Run Options is set to Normal or Retrieve. This will be disabled if Now is selected and a DSP waveform is being plotted.

Now: This uses the Display settings to reduce the number of data points in the plot that is presently in memory. The deleted data points cannot be retrieved.

Run Time: This will use the Save and Display settings the next time a simulation is run with the circuit file.

While the Reduce Data Points option is also available in AC and DC analysis, it typically is not required as those simulations are usually not data point intensive.

Network Installation Troubleshooting

The difference between the stand alone and the network version of Micro-Cap is how the license security is installed and accessed. Rather than connecting a dongle to the local computer, the network version of Micro-Cap lets the dongle be placed on the server. This dongle, in conjunction with the license manager software which is also installed on the server, controls the number of simultaneous users of Micro-Cap. For example, if a 3 seat LAN version was purchased, then a maximum of three users can access Micro-Cap at the same time. While installing the latest HASP device driver will fix nearly every security issue on the stand alone version, the network version has more points at which a security issue can arise. This article will provide a guide on how to troubleshoot possible security issues on a network installation. Since the most common network protocol in use currently is TCP/IP, the focus of this article will be on troubleshooting a network with such a protocol.

When Micro-Cap is launched on the client, the first thing the program will check is to see which protocols are installed on the system. The client will then send out a login request whose format will be dependent upon which protocols are available. Once the license manager on the server receives the login request, it will check to see if the correct dongle is attached to the server. If the correct dongle is present, the license manager will make sure that there is at least one free license available for Micro-Cap. If the login can not be validated, such as when all the licenses are currently in use or the dongle is not available on the server, the license manager will return an error. Otherwise, the license manager will record the user that logged in, and Micro-Cap will be able to start on the client system.

For the network version, the security problem may arise on either the server or the client. While there are no hard rules to determine which computer is causing the problem, there are a couple of ways to indicate where the problem most likely resides. First of all, if any of the clients can run the Micro-Cap software then the problem most likely is occurring on the client system that is unable to run Micro-Cap. If all clients are unable to run the program, it is a good chance that the server is the origin of the problem. The second indication is through the error message that is returned when Micro-Cap is unable to start. One of two error messages are likely to appear. Either, the "Can't find the NetHASP Server" error, shown in Figure 2, or the "Security key missing. Please replace the key" error will be invoked. While not as good of an indicator as the first method, generally the NetHASP error means that the client is unable to find the license manager software, and the security key error means that the HASP license manager is found but there is not a valid dongle associated with it.

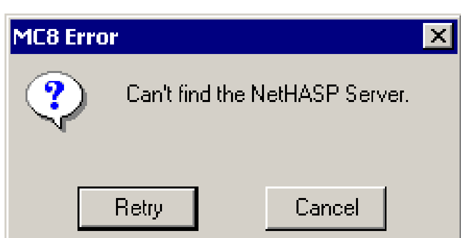


Fig. 2 - Security error message

Server
The security installation on the server is rather simple. The hardware dongle should plug into either the parallel port or the USB port depending on the type of dongle that was purchased. If the server is running an older Windows operating system such as Win9X or WinNT, the HASP device driver should be installed. Finally, the license manager must be installed. If the server is the suspect in the security problem, try the following steps:

1) Make sure that the license manager software (also the HASP device driver if used) is the latest version. The latest version can always be found on Aladdin's website at http://www.aladdin.com. Install the newest version if necessary.

2) It is quite rare for the dongle to fail, but it is a possibility. Try attaching the network dongle directly to one of the client systems that has Micro-Cap installed on it. The client system must have the HASP device driver installed for this test. Try to run Micro-Cap. This will run Micro-Cap as a stand alone version rather than a network version. It is possible that the portion of the key that just controls the network security capability has failed. However, we have never encountered this situation.

3) One other issue that can occur with a Windows server is that the clients have sporadic access to the license manager. At times, Micro-Cap will launch without a problem and at other times, attempts to start the program will return one of the two security login errors. This is usually caused because the license manager has been installed as an application on the server. If the user whose profile the license manager was installed into logs out, the clients will not have access to the license manager as it will no longer be running. The way to solve this problem is to install the license manager as a service instead. The best method to do this is to run the LMSETUP.EXE file which installs the license manager. During the installation process, one of the screens will ask whether the license manager should be installed as an application or as a service. Installing the license manager as a service ensures that the license manager is available to clients even when no one has logged onto the server.

Client
On the client system, the most common configuration is installing the Micro-Cap software on the local drive. The only other component that should be installed on the client is the HASP device driver which will be placed on the system during the course of the normal Micro-Cap installation. The client is typically the failure point for the security problem. There are a number of possible reasons for a failure. Some of them are that the client is unable to see the server that the license manager is located on, the network login request times out before the license manager can be found, or the communication path between the client and server is blocked. If the client is the suspect in the security problem, try the following steps:

1) Try to ping the server. Ping is a network utility that provides a basic test to see if the system being pinged is both operating and accessible from the client. In order to use this utility, the IP address of the server would need to be known. On the client system, open a DOS Command Prompt window. Once the prompt appears, type in the following:

ping <address>

where <address> would be an IP address such as 243.212.54.41. When successful, the ping utility will return a reply from the pinged system which includes the round trip time and packet loss rate. Should the pinged system not be found, the response will say that the request has timed out. If the system with the license manager can not be pinged from the client, Micro-Cap will be unable to run on that client.

2) Install and run the Monitor program from Aladdin. The installer for the Monitor program can be found on the Micro-Cap CD under the LAN Monitor folder. The Monitor program allows centralized administration of any NetHASP license managers that are installed on the network. A sample screen shot of the Monitor program is displayed in Figure 3. In this figure, a single NetHASP license manager has been found and is displayed underneath the NetHASP LM category. Selecting a license manager item in this category displays the name and address of the computer that the license manager is located on. If no items appear under the NetHASP LM category, then the client is currently unable to see the license manager.

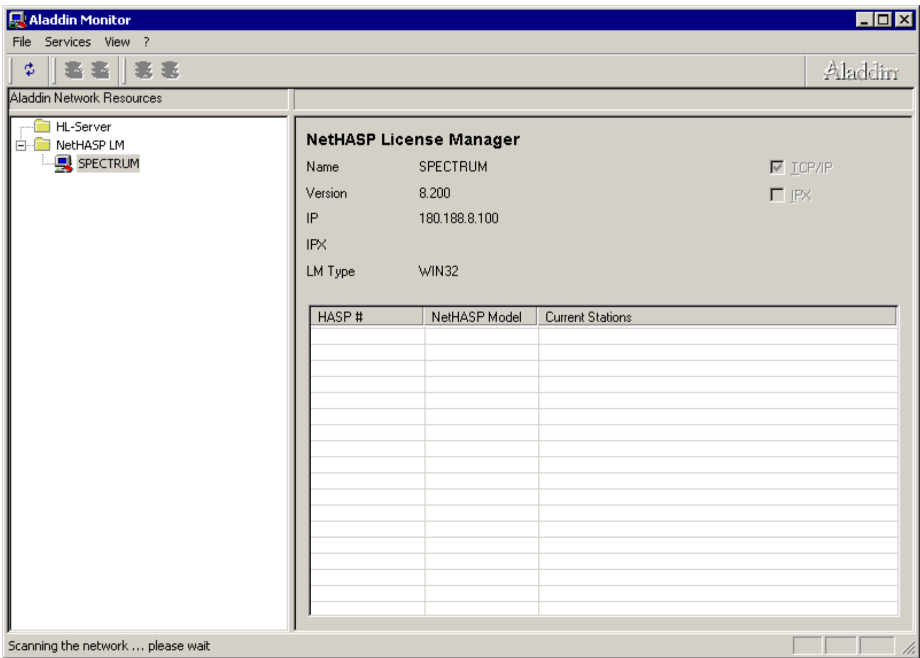


Fig. 3 - Aladdin Monitor program

The Monitor program is also useful if Micro-Cap returns an error about an old license manager being installed on the network. This program would show the location of the other license manager that Micro-Cap is finding. In this case, the older license manager should either be updated or the Nethasp.ini file, which will be described below, should be used with Micro-Cap.

3) Add the Nethasp.ini file to the folder where the Micro-Cap executable is located. This file is a configuration file for the client that specifies how it will communicate with the license manager on the server. Rather than sending the login request throughout the network, this text file lets a specific server address be defined to which the login request will be made. This file is also useful if the load time for Micro-Cap is adversely affected by the search for the license manager. One technique that Aladdin recommends is using the Nethasp.ini file in the same folder as the Monitor program and then make the appropriate edits in the file. As soon as the Monitor program can see the license manager, then just copy the Nethasp.ini file over to the folder that the Micro-Cap executable is located in. The standard Nethasp.ini file for TCP/IP communication is shown as follows. Simply fill in the server address for the NH_SERVER_ADDR entry. Basic templates for the IPX and NetBios protocols can be found on Aladdin's website.

```
[NH_COMMON]
NH_TCPIP = Enabled;                   ; Use the TCP/IP protocol

[NH_TCPIP]
NH_SERVER_ADDR = xx.xx.xx.xx;         ; IP addresses of all the NetHASP
                                      ; License Managers you want to search.
                                      ; Unlimited addresses and multiple
                                      ; lines are possible.
                                      ; Possible address format examples:
                                      ; IP address:      192.114.176.65
                                      ; Local Hostname:   ftp.aladdin.co.il

NH_TCPIP_METHOD = UDP                 ; Send a TCP packet or UDP packet
                                      ; Default:  UDP

NH_USE_BROADCAST = Enabled            ; Use TCP/IP Broadcast mechanism.
                                      ; Default:  Enabled
```

4) In the Nethasp.ini file, set the NH_USE_BROADCAST entry to Disabled. This setting disables the TCP/IP broadcast mechanism. This is most helpful if the client is on a different subnet or domain then the server.

5) Verify that the communication port (475) to the license manager is open and not blocked by a firewall. On the client system, open a DOS Command Prompt window. Once the prompt appears, type in the following:

telnet <ip address> 475

where <ip address> is the IP address of the server where the license manager is located. If working, the DOS window will be cleared. Pressing any key will return the message "Connection to host lost". If port 475 is blocked, the message “Could not open a connection to host on port 475 : Connect failed” will appear. The port would then need to be opened to enable communication between the client and the license manager.

6) Increase the timout value in the Nethasp.ini file. The default initial search period for the NetHASP algorithm is two seconds. This search period can be extended by adding the following text into the [NH_COMMON] section of the Nethasp.ini file:

NH_SESSION=<seconds>

where <seconds> is the new value for the initial search.

7) Reboot the client. We have found in a couple of cases that after making some of the above changes, a reboot of the client system solved the security problem. While it typically should not be necessary to do this, apparently the HASP software may keep some of the communication settings in memory so a reboot will clear those.

Modeling Impedance with Tabular Data in AC Analysis

Quite a few components can be modeled in Micro-Cap by simply simulating their impedance characteristics. One method to model a complex impedance is to combine resistors, inductors, and capacitors together until the impedance versus frequency curve matches the expected results. This method can be very time consuming if done by hand. If the model is to be simulated in an AC analysis, an easier method is described in this article that will use data triplets of the frequency, impedance magnitude, and impedance phase to represent the complex impedance.

One device where modeling the complex impedance is crucial for an AC simulation is a loudspeaker. A typical speaker will specify a nominal impedance such as eight ohms. This only describes the standard impedance of the speaker. Over different frequencies, the speaker impedance may vary drastically from this nominal value. At this point, it will be assumed that the impedance magnitude and phase versus frequency for the loudspeaker has already been obtained through measurement of a physical device, from a data sheet, or through another program. For this example, the impedance curve defines the impedance characteristics for a fifteen inch, eight ohm loudspeaker from 1Hz to 20kHz. This curve was derived from data in the book "High Performance Audio Power Amplifiers" by Ben Duncan. Once the data is available, it should be converted into a .define statement and placed in the Text page of a schematic. Part of the define statement for the specified loudspeaker is as follows:

```
.define Speaker
+        1.000000,      6.642448,       7.075845
+        1.050764,      6.648317,       7.075845
+        1.104105,      6.654794,       7.430443
+        1.160153,      6.661941,       7.802316
+        1.219047,      6.669827,       8.192229
.           .              .               .
.           .              .               .
.           .              .               .
+    16406.252669,     70.373163,      85.429172
+    17239.098286,     73.939376,      85.649411
+    18114.222406,     77.689242,      85.859090
+    19033.771253,     81.632330,      86.058708
+    20000.000000,     85.778745,      86.248742
```

This define statement creates the symbolic variable Speaker. The + signs are used as a continuation character to tell Micro-Cap that the following lines are still associated with the Speaker variable. Each data triplet is defined as:

Frequency, Impedance magnitude, Impedance phase

Alternatively, the real and imaginary values of the impedance can also be used in this method by defining the triplet as:

Frequency, Real impedance, Imaginary impedance

The components within Micro-Cap that can read in data in this type of format are the Laplace table sources. The Laplace table source comes in four varieties: LTIofI - current controlled current source, LTIofV - voltage controlled current source, LTVofV - voltage controlled voltage source, and LTVofI - current controlled voltage source. For modeling impedance, only the LTVofI and the LTIofV sources are applicable. Due to the data types specified in the Speaker define statement, the Laplace source which will be used in this instance is the LTVofI. This Laplace source measures the current through its input, looks up the transfer function in the defined table, and produces the resultant voltage output. For impedance, the source should be wired so that the measured input current is the current through its own voltage output. This creates the function:

V = I * Table transfer function

where I and V are both assigned to the Laplace source output so that the table transfer function is the direct equivalent of the complex impedance. This method models the voltage drop that would exist if the impedance was present in the circuit. Should a LTIofV source be used instead, the voltage inputs need to be wired to measure the voltage across the current source output as this method would model the current flow that would exist if the impedance was present. The table data would also have to be transformed into the following:

Frequency, 1/Impedance magnitude, - Impedance phase

since the table transfer function should now be the equivalent of the conductance. An example circuit displaying the use of the LTVofI impedance method is displayed in Figure 4.

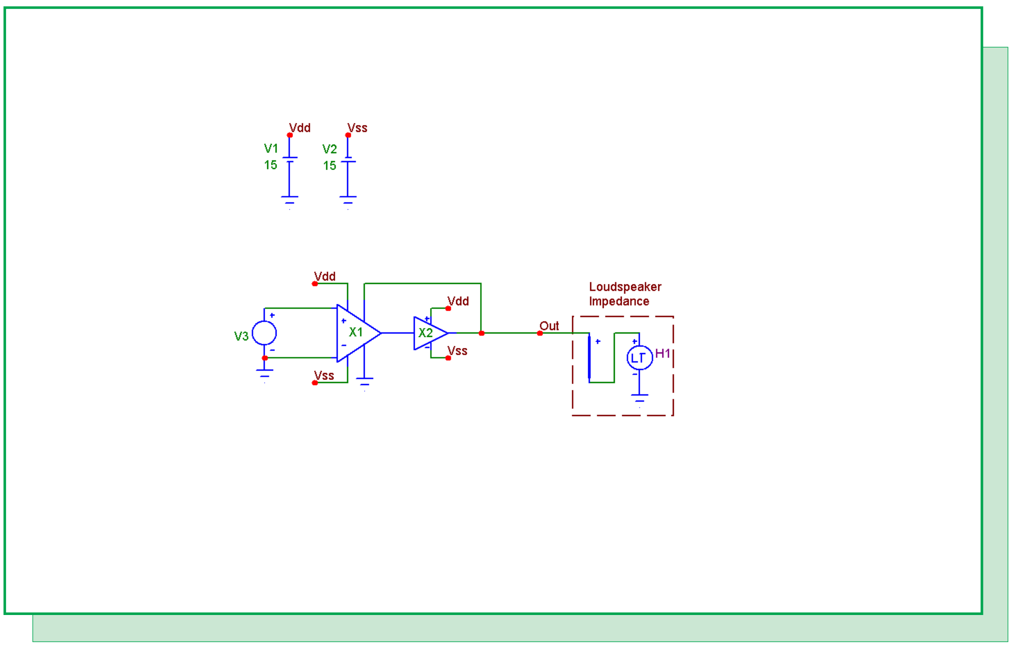


Fig. 4 - Loudspeaker impedance example

In the schematic, an AC source of magnitude 1 is the input into an INA134 audio differential line receiver. The output of the INA134 is then fed into the input of a BUF634 high speed buffer. The Sense pin of the INA134 is connected to the output of the BUF634. This configuration provides an output current boost to the circuit. The LTVofI component that models the loud-speaker impedance is connected to the output of the BUF634 device. Note that the Laplace source has been wired so that it is measuring the current through itself. The Laplace source has its attributes defined as:

FREQ = Speaker
KEYWORD = Mag Deg

The FREQ attribute defines the table values for the source. In this case, the Speaker symbolic variable has been entered which will use the define statement that was created previously and stored in the Text page of the schematic. The KEYWORD attribute defines the type of data that is specified in the table. Mag indicates that the magnitude value is true magnitude and Deg specifies that the phase value is in degrees.

The AC analysis for this circuit is displayed in Figure 5. The impedance of the speaker output is calculated by using the expression V(Out)/I(H1). The top plot shows the magnitude of the impedance and the middle plot shows the phase of the impedance. The impedance curves show the general impedance characteristics of a loudspeaker. These two waveforms match precisely with the data that was specified within the Speaker table definition. The bottom plot displays the output current that is generated through the LTVofI source.

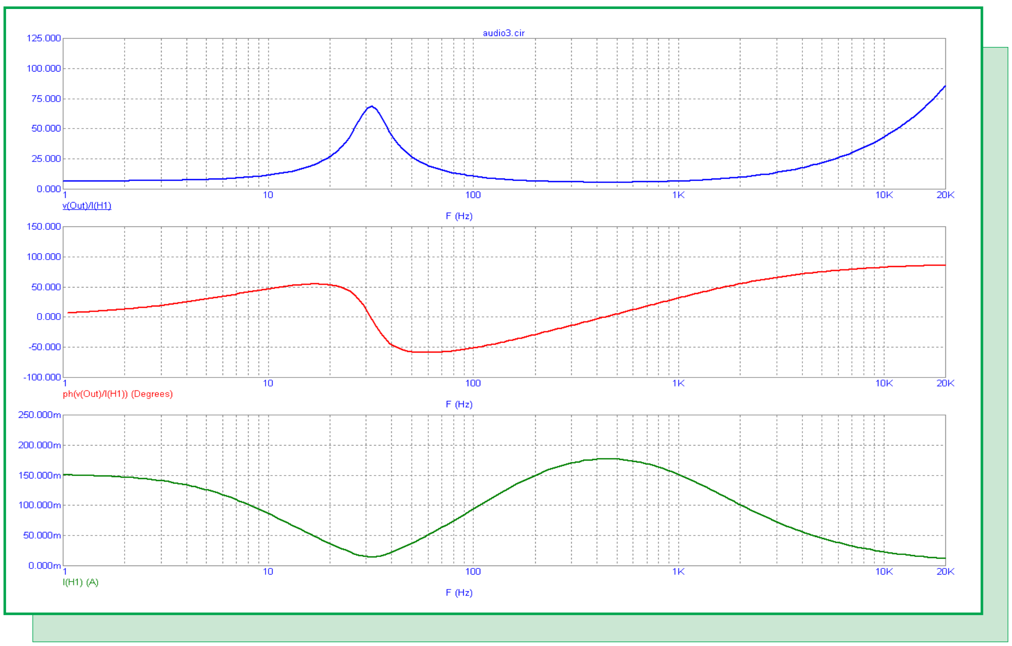


Fig. 5 - Loudspeaker impedance AC simulation

BJT Temperature Modeling

For many SPICE models, temperature modeling has not been optimized. The temperature parameters that affect the characteristic curves of the model are typically left at their default values which may not be accurate for simulations whose temperature setting deviates from the temperature that the model's parameters were optimized at. Most device models are optimized at either 25 or 27 degrees Celsius which is considered room temperature. The circuit optimizer that is available in transient, AC, or DC analysis for Micro-Cap can be used to derive the temperature parameter values for more accurate simulation results.

For the BJT model, the two primary temperature parameters in the model are XTI and XTB. XTI is the saturation current temperature exponent and is used to change the transistor global saturation current sensitivity. The global saturation current is used in the computation of the current through the diodes that connect from the base to the collector and emitter. Modifying XTI generally affects the low current region of the beta curve more than the high current region. XTB defines the temperature coefficient for betas. Modifying XTB vertically shifts the nonlinear beta curve.

The circuit in Figure 6 is used to plot the nonlinear beta curve in transient analysis. The transistor model used in the circuit is a 2N3946 general purpose NPN transistor. The data for this transistor model was derived from the Motorola "Small-Signal Transistors, FETs, and Diodes" data book. In the data book, the VCE for the current gain characteristic is specified at one volt so the battery at the collector in the circuit has been given the same value. The voltage source at the base of the transistor has been defined as follows:

DC 0 AC 1 0 Pulse .15 1.5 0 .5 1 .5 5

which will provide a voltage ramp from .15V to 1.5V during the first .5s which is the length of the simulation that will be run.

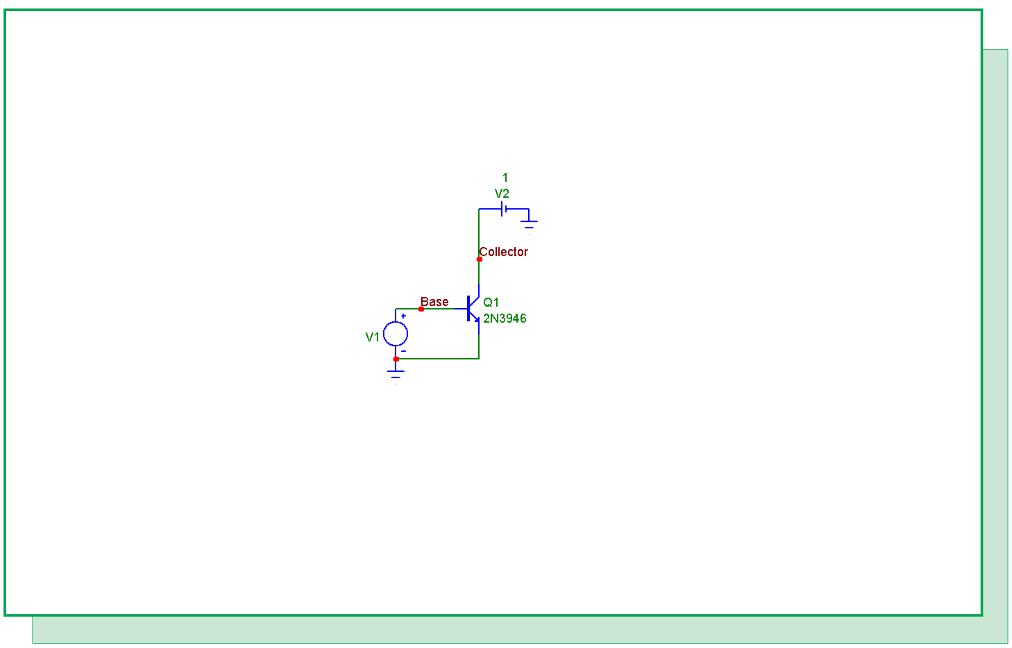


Fig. 6 - Forward beta measurement circuit

The results of a .5s transient simulation of this schematic when the temperature is set to the nominal temperature of 27C produces the nonlinear beta curve shown in Figure 7. This curve matches closely with the curve shown in the Motorola data book.

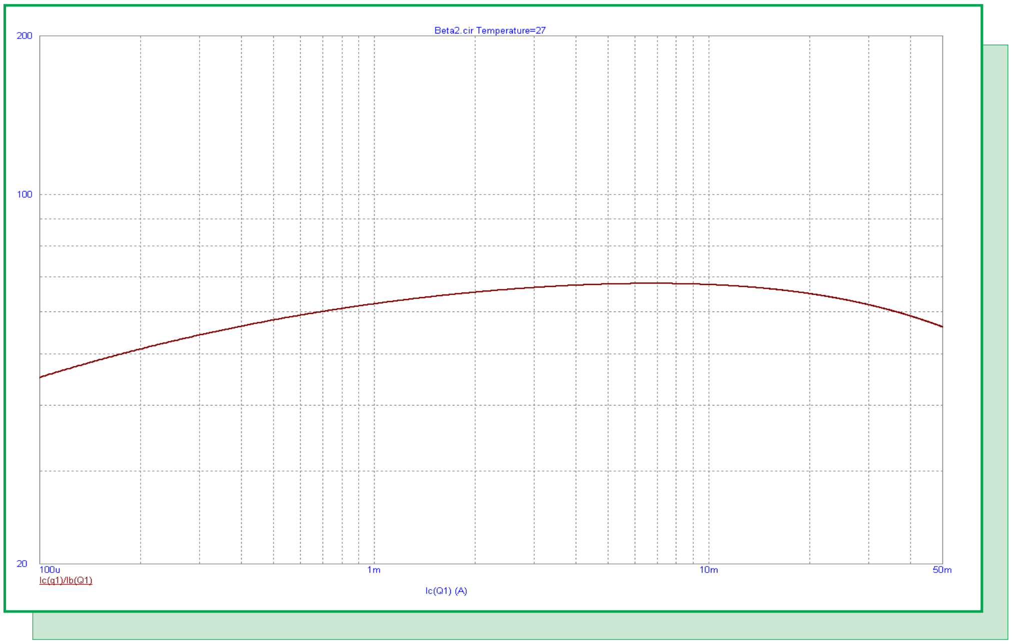


Fig. 7 - 2N3946 current gain curve at 27C

If the Temperature field is set to 100 in the Transient Analysis Limits dialog box, the beta curve that is then produced is a good deal off of the one specified in the data book for that temperature. The reason for this is that the XTI and XTB model parameters are set to their default values. The optimizer available within transient analysis will be used to calculate values for XTI and XTB to better match the beta curve at 100C.

Prior to entering the optimizer, the settings in the Transient Analysis Limits dialog box must be set to the values that the curve is to be optimized at. In this case, the Temperature field is set to 100. To enter the optimizer, select the Optimize option under the Transient menu. For optimizing the XTI and XTB parameters, the optimizer settings for this example are shown in Figure 8.

The Find section specifies the parameters that are to be optimized. For both the XTI and XTB parameters, the Parameter Type option selected was Model so that the 2N3946 model is being optimized directly. If there was more than one transistor in the schematic referencing the 2N3946 model then all instances of this transistor would be optimized with this option rather than just the single specified transistor that would be optimized using the Component type method. The Parameter Type along with the parameter to be optimized can be selected by clicking on the Get button.

Since the object in this optimization is to match the beta curve from the Motorola data book at 100C, the optimizing criteria in the That section for each function is set to Equates. This is the criteria that needs to be used for any curve fitting operation. A sampling of data from the data book for the beta curve at 100C is:

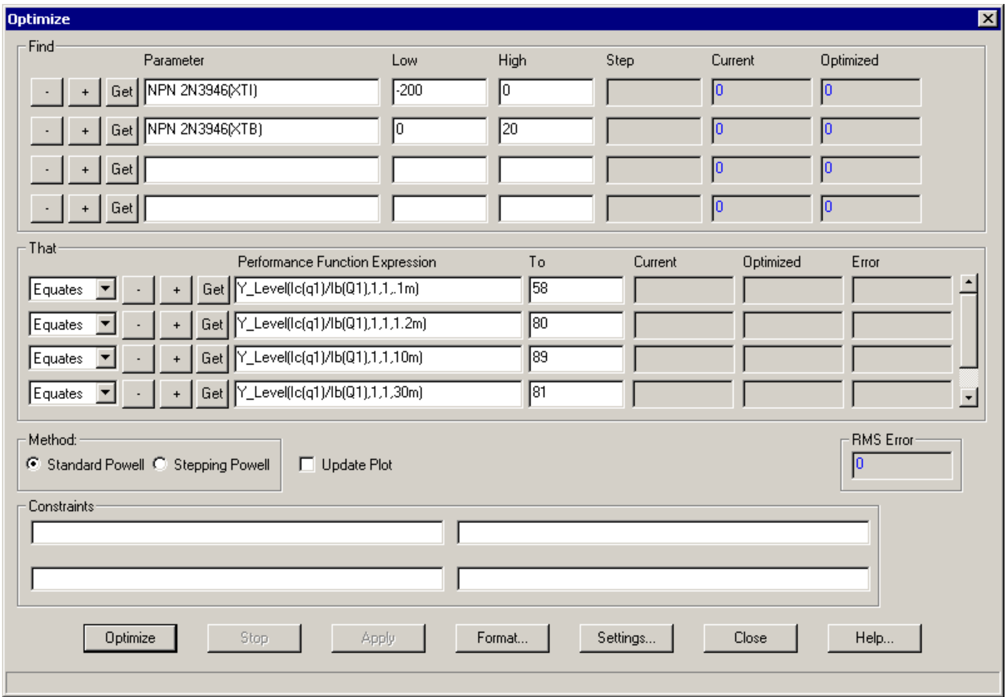


Fig. 8 - Optimizer dialog box settings for XTI and XTB

@ Ic = .1mA, Beta = 58
@ Ic = 1.2mA, Beta = 80
@ Ic = 10mA, Beta = 89
@ Ic = 30mA, Beta = 81
@ Ic = 50mA, Beta = 70

The equivalent performance functions along with their To values that the optimizer needs to use to match to this data is:

Function: Y_Level(Ic(Q1)/Ib(Q1),1,1,.1m) To: 58
Function: Y_Level(Ic(Q1)/Ib(Q1),1,1,1.2m) To: 80
Function: Y_Level(Ic(Q1)/Ib(Q1),1,1,10m) To: 89
Function: Y_Level(Ic(Q1)/Ib(Q1),1,1,30m) To: 81
Function: Y_Level(Ic(Q1)/Ib(Q1),1,1,50m) To: 70

The Y_Level operator will try to optimize the Y value of the expression Ic(Q1)/Ib(Q1) at the specified X value to the value set in the To field. Since the X Expression in the Transient Analysis Limits dialog box is set to Ic(Q1), the X value is the collector current of the transistor. For example, with the first expression in the list, when the collector current is .1mA, the optimizer will try to determine the values of XTI and XTB in the 2N3946 model so that the output of the Ic(Q1)/Ib(Q1) expression is equal to 58. Since five performance functions have been specified, the optimizer will find the values of XTI and XTB that produce the curve that creates the smallest total RMS error between the target and actual values at each point. Each of the Equates conditions are weighted equally in terms of importance when optimizing.

The Standard Powell optimization method has been selected, and no constraints have been specified. Clicking on the Optimize button initiates the optimization. For these particular settings, the optimizer calculates a value for XTI of -65.051 and a value for XTB of 1.193. The total RMS error for the curve is 2.722 which is a good match.

Clicking on the Apply button will update the 2N3946 model in the schematic so that the new XTI and XTB values are used. Applying the updated parameters from the optimizer does not actually overwrite the 2N3946 model in either the Micro-Cap library or in the schematic. Instead, any transistors that were previously referencing the 2N3946 model in the schematic will now be referencing a model called 2N3946_OPT1 which has been placed in the Model page of the schematic. The updated model for the 2N3946 appears as follows:

```
.MODEL 2N3946_OPT1 NPN (IS=0.629196F BF=78.0026 NF=929.132M VAF=100
+ IKF=145.824M ISE=985.815F NE=1.74045 BR=105.035M IKR=46.8389 ISC=100P RE=2
+ RC=394.919M CJE=7.01663P VJE=907.488M MJE=300M CJC=4.87834P VJC=756.556M
+ MJC=300M TF=570.088P XTF=500.002M VTF=10 ITF=9.94048M TR=3.50347U
+ XTB=1.193 XTI=-65.051)
```

This model is a copy of the 2N3946 model with just different values for XTB and XTI. Running the same schematic with the updated model statement produces the nonlinear beta curve at a simulation temperature of 100C that is shown in Figure 9. This beta curve matches closely with the one shown in the Motorola data book.

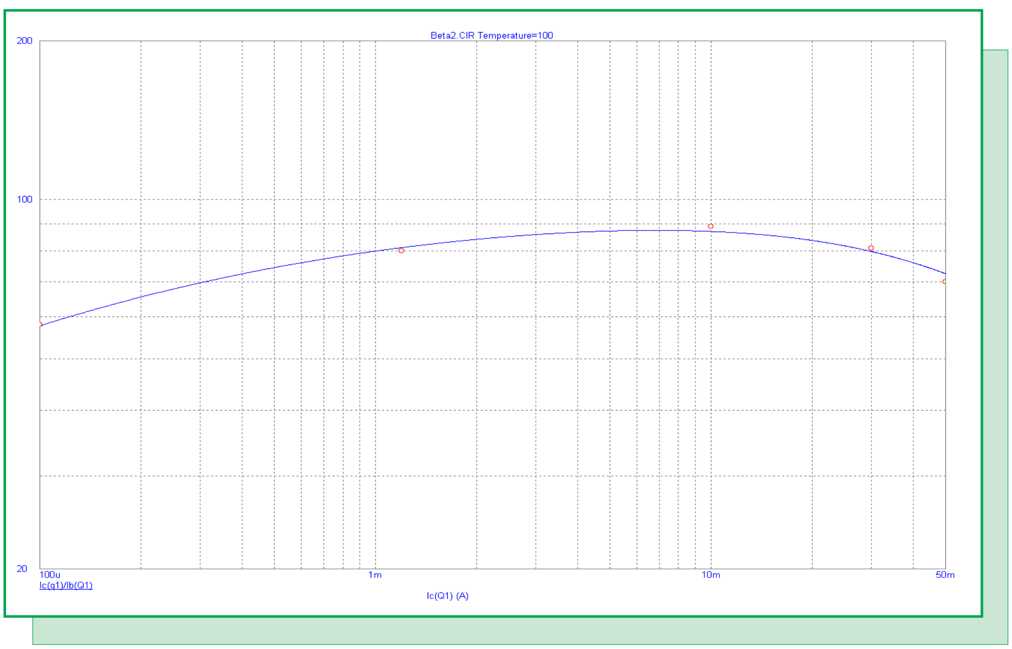


Fig. 9 - 2N3946 optimized current gain curve at 100C

Product Sheet

Latest Version numbers

Micro-Cap 8 .... Version 8.1.1
Micro-Cap 7 .... Version 7.2.4
Micro-Cap 6 .... Version 6.3.3
Micro-Cap V .... Version 2.1.2

Spectrum's numbers

Sales .... (408) 738-4387
Technical Support .... (408) 738-4389
FAX .... (408) 738-4702
Email sales .... sales@spectrum-soft.com
Email support .... support@spectrum-soft.com
Web Site .... http://www.spectrum-soft.com
User Group .... micro-cap-subscribe@yahoogroups.com