LTspice – An Introduction

Version: 0.3

http://kom.aau.dk/∼hmi/Teaching/LTspice/

October 12, 2005

Jan Hvolgaard Mikkelsen, M.Sc.E.E.,
E-mail: jhm@kom.aau.dk

RF Integrated Systems & Circuits (RISC) Division,
Institute of Electronic Systems,
Aalborg University,
Fredrik Bajers Vej 7-A6,
DK-9220 Aalborg,
Denmark.

LTspice – An Introduction

Version: 0.3

http://kom.aau.dk/~hmi/Teaching/LTspice/

Jan Hvolgaard Mikkelsen, M.Sc.E.E.,
E-mail: jhm@kom.aau.dk

CONTENTS

1 INTRODUCTION

When looking for simulation software a host of different candidates are available. Some of these software packages are unique in the sence that they offer something special to the user. This could for instance be an extensive library of standard components that are readily found at the local electronics store. Some of the packages are free and of course some require registration and payment to run. A great number of the free versions are limited and may hence only support analysis of simple circuits. One example is the free student version of `Pspice` which originally only supported 20 some nodes.

Most of the free simulation packages are based on a SPICE (*Simulation Program with Integrated Circuit Emphasis*) simulation engine. This implies that all simulations take place in the time-domain. In contrast to this some tools make use of the frequency-domain for the circuit analysis. As always there are pros and cons for either approach.

This report provides a short introduction to one of the many free simulation software options, `LTspice`. The software is provided by Linear Technology[1] and it comes without any limitations to its use. It should be noted that the graphical user interface (GUI) does not offer access to the complete range of functionalities available in `LTspice`. Despite this fact `LTspice` does offer the complete range of SPICE functionalities. The only thing to rememer is that some functions are available through the `'SPICE directives'` option only. It is also worth noticing that the help functionality of `LTspice` is quite extensive and it provides for good documentation of the program.

Updated information on where to find the software and how to install it is available at the following www location:

http://kom.aau.dk/~hmi/Teaching/LTspice/LTspice_Intro.html

There is also an LTspice faq available. Here known issues and different solutions and work-arounds are presented. The page is constantly updated and it is found at the following www location:

http://kom.aau.dk/faqomatic/cache/198.html

This version of the `LTspice` introduction document referrers to `LTspice` version 2.15z, August 15, 2005

Jan H. Mikkelsen, Aalborg, October 2005

---

[1]http://www.linear.com

2   SCHEMATIC ENTRY AND SIMPLE ANALYSIS

After installing LTspice according to the web instructions a 'SwCAD III' icon is found on the desktop. Double-clicking this will launch LTspice and the window illustrated in Figure 1 pops up.
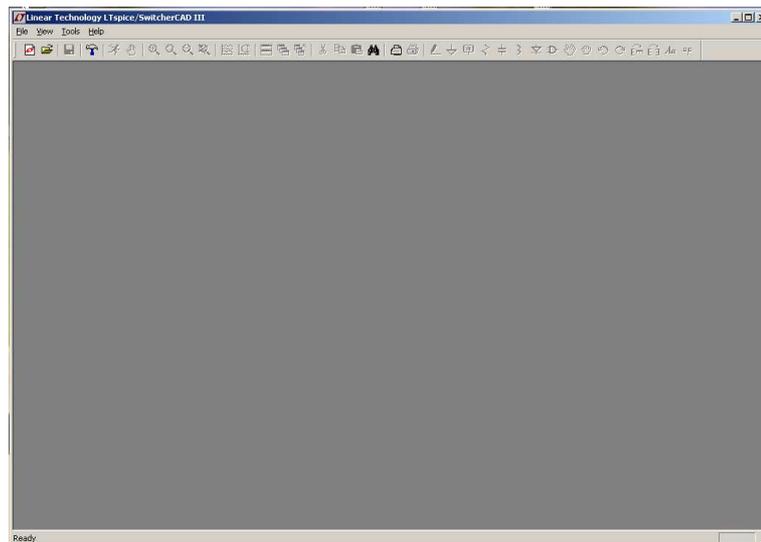
FIGURE 1: *Start-up window when launching LTspice.*

The window shown in Figure 1 is the LTspice main window and it contains almost all the functionalities of the software. At first, the majority of the buttons are non-selectable but as soon as the 'New Schematic' button is pressed this changes. All the functionalities that are available via the buttons are of course also available through the different menues. To avoid any confusion only the menu items are presented here and the use of the icon-buttons is left for the user to explore.

To start a schematic design all the different components have to be inserted. For instance, to insert a capacitor use 'Edit -> Capacitor' and place the component anywhere in the schematic window. In this way all the different components may be placed to form the wanted schemantic. Using 'Edit -> Draw Wire' the connections are formed and the final circuit is completed. Once the circuit is completed it is always a good idear to add lables to the different connections and nodes that are of special interest. This is done using 'Edit -> Label Net'. Once the different nodes have been labled it is much easier to plot the correct voltage and currents as the lables are used as node Id's. If no labels are used all the signals are simply numbered and telling what node has what number is very difficult. It is important to remember that a ground node must always be provided in any schematic for the simulator to work.

After adding lables and appropriate ground node(s) the circuit is in principle ready for simulation. As a simple example the RC circuit illustrated in Figure 2 is used. Here two components, a resistor and a capacitor, have been added in a lowpass configuration. As seen from the figure both input and output nodes have been assigned a label.
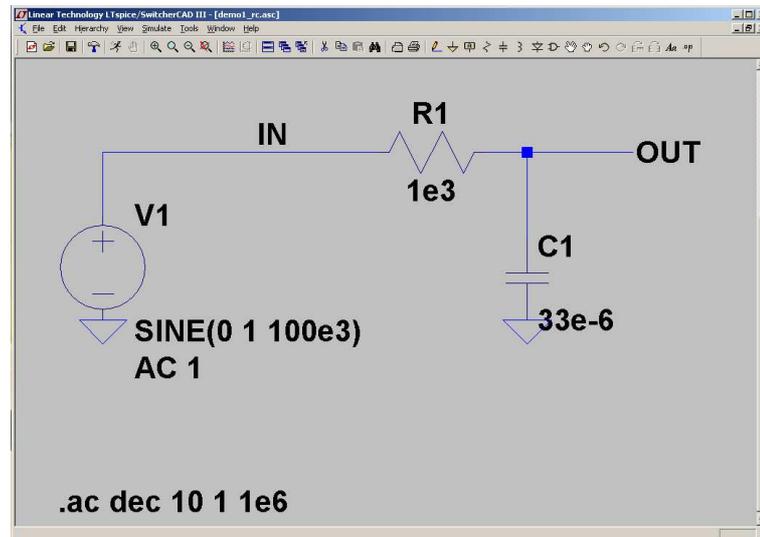
FIGURE 2: *Simple schematic entry in LTspice (demo1_rc.asc).*

A wide range of components and models are included with the LTspice installation. All the user needs to do to get access is to use the 'Edit -> Component' menu option. This produces an extra window where all the different libraries may be accessed. To add the voltage source shown in Figure 2 choose the 'voltage' component and place it in the schematic. To change properties for components right-click on the component to get access to the properties form. Use these forms to set parameters as indicated in Figure 2. It is important to note that the 'voltage' component properties form has different areas where parameters may be specified. Depending on the type of simulation that the user wants to run, different areas need to be specified. For AC simulations only the 'Small signal AC analysis' (.AC) field needs to be completed. To enable a transient analysis the values under 'Functions' need to be specified.

Further, as a general rule users should always be careful using SI units when entering values in the different simulation tools. Often 'M' is interpreted as equal to 'm' and insted of getting a value in Mega-Ω a resistor value in milli-Ω results.

2-1   AC Analysis

With all parameters set, the next step is to run a simulation on the circuit. For LTspice, and any other simulation tool for that matter, a number of specialized simulation types are available. If no specific analysis is specifided the user is prompted for this once the simulation is started. Therefore, the first step is to start the simulation using 'Simulate -> Run'. This produces a new window where the different simulations may be chosen. For this example a simple ac simulation is used. Select 'AC analysis' and set the sweep type to 'Decade', the number of points per decade to 10, the start frequency to 1Hz, and the stop frequency to 100MHz. After pressing 'OK' the simulation continues with the entered simulation data. The next thing that happens is that the user is asked to select what waveforms that are to be plotted. Voltages are

given as the voltage in a specific point while currents are given as the current through a specific component.

It is important to realize that all (almost) simulation tools use a syntax that has positive currents flowing INTO a component. Since a (positive) current flowing into a resistor eventually has to leave the resestion again, then this output current has to be negative which confuses everything. To prevent this many simulation tools have some kind of indication of which component terminal is the positive one. This is not the case in LTspice and it can therefore be very difficult to determine the direction of a current. This has absolutely no effect on the accuracy of LTspice, it is only when a current curve is plottet that the phase may change by 180 degrees depending on the orientation of the component.

Due to the use of lables it is easy to see what waveform corresponds to the output and to plot this just choose 'V(OUT)' and then press 'OK'. This produces a new window, see Figure 3, where amplitude and phase of the output signal is illustrated.
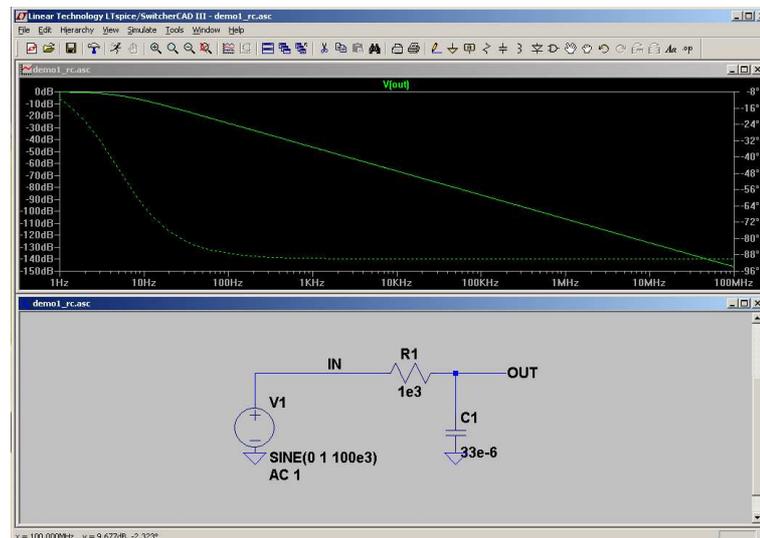
FIGURE 3: *Result of a simple AC analysis in LTspice (demo1_rc.asc).*

As Figure 3 illustrates the output of an AC simulation is a frequency response plot – much like the well known bode plot actually. By looking at the amplitude plot it is easy to see that the circuit implements a lowpass functionality. It is possible to add and delete waveforms at will and since LTspice stores all node voltages and currents per default, adding a curve does not require a re-simulation. If single-clicking on a node or wire the curve is added to the existing curves but if the user double-clicks then the plot window is reset and only the just chosen curve is plotted.

To prevent the phase from being plotted double-click on the axis to the right of the graph. This allows the user to specify axis information and by ticking the 'Don't plot phase' field it is possible to have only the magnitude plot visible.

To determine the 3dB corner frequency of the filter double-click the 'V(out)' text. This produces a marker window and by dragging the marker it is possible to have an accurate reading

of the performance. In this example the magnitude curve is found to equal -3dB at a frequency of 4.82Hz. It is of course possible to use the zoom functionality to scale the plot some.

An AC simulation is a simple small-signal analysis. This basically meens that the simulator forms a linear model for all non-linear components and then use this for the analysis. To understand why this is important consider the case where an amplifier runs on a ±12V. When a circuit runs on ±12V it generally cannot support output voltage swings that exeecs the supply voltages. If the amplifier provides for a gain of 10 then any input signal larger than ±1.2V is going to cause signal clipping on the output. Because of the small-signal linearization this is not possible to see from an AC simulation. Here, a 100V input signal is going to result in a 1kV output signal. This is clearly not possible and to reveal such effects a transient analysis is needed.

2-2   *Transient Analysis*

To determine if signal clipping tages place and to evaluate any non-linearities a transient analysis is nesessary. The transient analysis is a large-signal analysis and it is able to take non-linearities into consideration. It is fairly easy to set up the transient analysis but for the purly passive RC circuit example the transient analysis makes no sense. To illustrate how the transient analysis works in practic, take the RC example from earlier and replace the resistor by a diode as shown in Figure 4. Note that the capacitor has also been reduced in size.
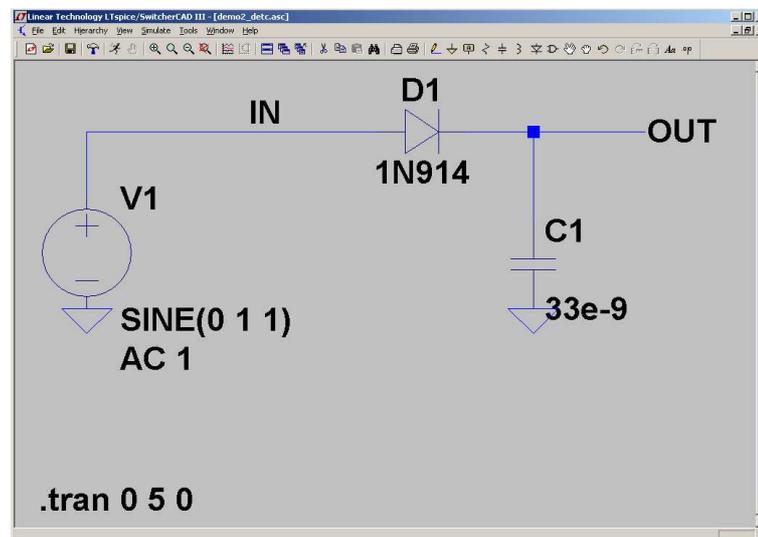
FIGURE 4: *Simple schematic entry in LTspice (demo2_detc.asc).*

When right-clicking on the diode symbol it is possible to specify some basic diode parameters or pick a specific diode using the 'Pick New Diode'. When choosing to select a new diode a window with all the diode models implemente in the LTspice component library is shown. After replacing the resistor with a 1N914 diode and reducing the size of the capacitor the schematic and the resulting AC simulation result is as shown in Figure 5.
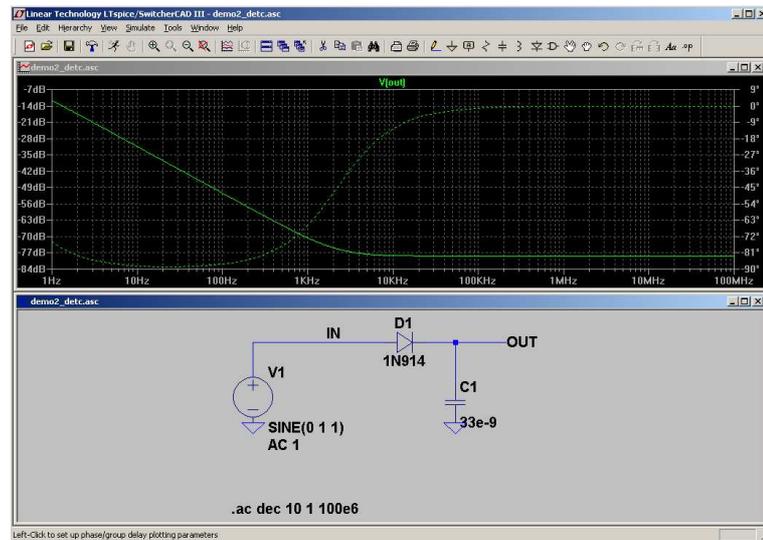
FIGURE 5: *Result of an AC analysis in LTspice (demo2_detc.asc).*

Clearly, an AC analysis does not produce any relevant[2] information for the new circuit. To have some usefull information a transient analysis is required. To do this the voltage source needs to be configured. In this example the source has been set to 1V (peak voltage) and a frequency of 1Hz. When the transient analysis is specified it is important to consider the number of signal periods that the simulation is to cover. Here five periods have been selected and this corresponds to 5s. Right-clicking on the '.ac' text in the schematic window allows the user to change analysis type. Like that the 5s is inserted as stop time for the transient analysis. The result of the simulation is shown in Figure 6 where both input and output signals have been included.

From the plots the circuit is seen to have a rectifying effect on the input signal. When Figures 5 and 6 are compared it is clearly seen that the two analysis types offer different information.

If a specific diode is not listed it needs to be added to the component library. This is the case if a user wants to simulate the performance of the circuit using for instance the 1N4148 diode.

3 ADDING COMPONENTS TO THE LTspice LIBRARY

Assuming that a specific diode is needed, the 1N4148 for instance, and this is not found in the component library the user needs to add this. The file structure for LTspice is very simple and this makes it easy to add any components to its library structure. If the software is installed in the default location it is located at:

C:/Program Files/LTC/SwCADIII

---

[2]This is not quite true but the AC analysis does not reveal all characteristics of a circuit.
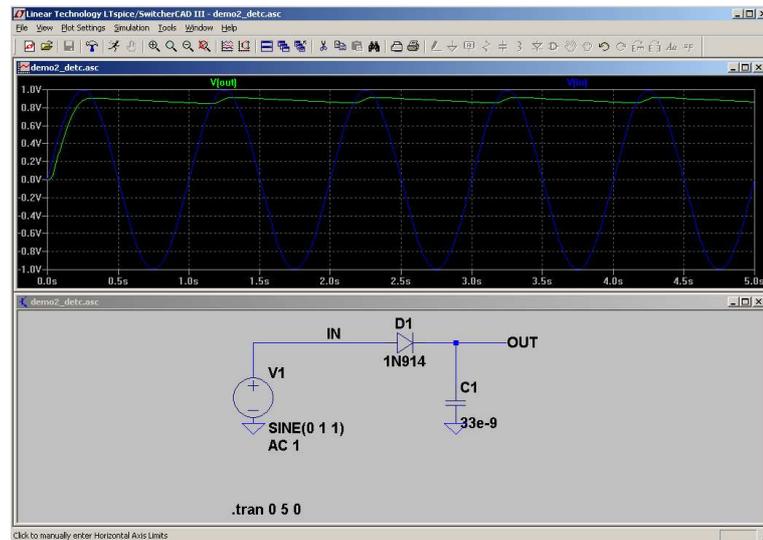
FIGURE 6: *Result of a transient analysis in LTspice (demo2_detc.asc).*

LTspice has its entire component library placed in the 'lib' sub-folder. This library is nicely seperated into a 'cmp', a 'sch', a 'sub', and a 'sym' folder. As the names suggest the component library is structured as

- cmp - Basic components and their models

- sch - Larger structures consisting of schematics

- sub - Sub-circuit descriptions to model the structures in the 'sch' folder

- sym - Symbol views of the different components.

When drawing the circuit schematic the user actually makes use of the symbols in the 'sym' folder. A very nice feature in LTspice is that several models may be added to any symbol. This meens that there does not have to be a symbolic view for every model included in the library. This makes it very easy to add new device models to the component library and it also makes it very easy to adjust model parameters if needed.

The first task is to find information on the specific diode and more importantly to find a spice model for it. One way to do this is to search the web using Google or even better the different component manufactures webpages. A Google search using the terms 1N4148 and diode gives a lot of hits, some contain data sheets and some even spice models. When downloading models from non-manufacture pages it is important to always verify the model by comparing it to a data sheet. Once the model has been verified it is a simple task to add this to the library. Since the diode is a basic component it is to be placed in the 'cmp' folder. Here a number of files are located:

- standard.bjt - BJT transistor models

- standard.cap - Capacitor models

- standard.dio - Diode models

- standard.ind - Inductor models

- standard.jfet - JFET transistor models

- standard.mos - MOSFET transistor models

- standard.res - Resistor models

Since all these are simple ASCII files the user only has to open the 'standard.dio' file in any editor and cut and paste the 1N4148 spice model into this file. When this has been done the new model should appear when the user tries to pick a new diode again.

Note that all the information contained in the files located in the 'cmp' folder is read during start-up of LTspice. This implies that the program has to be restarted whenever a new component model has been added to one of these seven files.

4 INCLUDING COMPLICATED MODELS TO THE LTspice LIBRARY

When more complicated models are required the steps are not as easy as for the simpler component models. The use of OPAMPs is an example where a complicated model is used to describe the performance. In principle, an OPAMP is a circuit in it self and contains both transistors, resistors, and capacitors. Since LTspice is maintained by Linear Technology most of their OPAMS are included in the library. One example is shown in Figure 7 where a simple inverting topology is build using the LT1001 OPAMP.
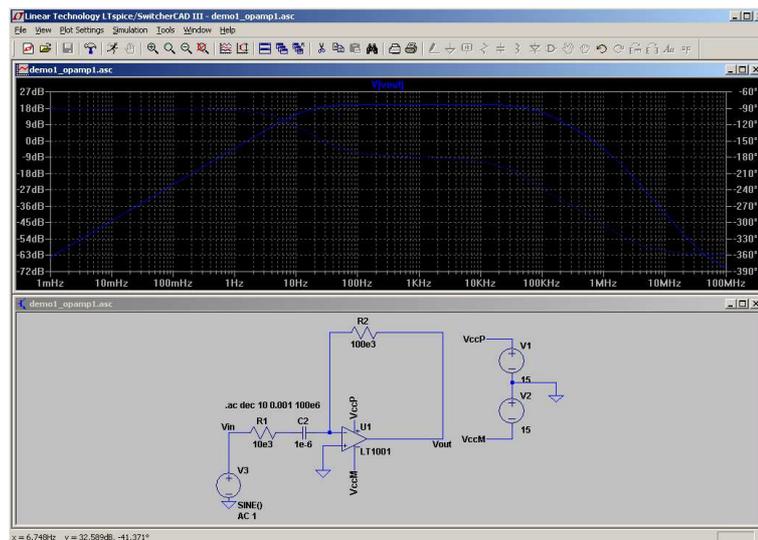
FIGURE 7: *Schematic entry using an OPAMP (demo3_opamp1.asp).*

The series capacitor on the input side blocks DC and provides for a highpass response with a corner frequency around 16Hz. At approximately 75kHz the respons has another corner frequency. Here it is the performance of the OPAMP that limits performance which then results in a lowpass effect.

With an input voltage of 1V and a gain of 10 the output voltage swing is +/- 10V which is easily accomodated by the supply voltages used in the example. If the input voltage is increased to 10V the AC simulation will show no changes at all despite having an output signal swing of +/- 100V.
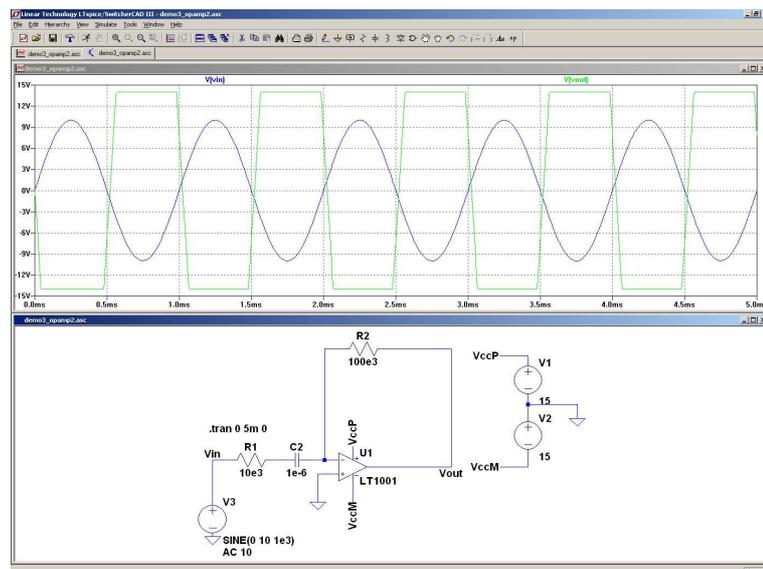
FIGURE 8: *Schematic entry using an OPAMP (demo3_opamp2.asp).*

If a transient analysis is conducted using the same input signal conditions the result is clearly visible as Figure 8 shows. Here it is clearly seen that the circuit enters compression.

4-1   Procedures for adding complicated models to `LTspice`

To add another OPAMP model to the library the first step is once again to search for the device. If for instance an TLE2037 is required a search is going to reveal that this is produced by Texas Instruments. On their webpages it is possible to download a spice based macro model for the device. Once this file has been retreived it often contains two different models, a level 1 and a level 2 model. The level 2 model is generally more accurate so this is the one to choose.

Once the model has been retreived one of two approaches may be used to add the device to the model library: 1) the quick and dirty approach, and 2) the nice and neatly structured approach.

4-1-1   Adding models the dirty way

Once the model is found it is a simple matter of copying the description to a file and store this in anyone of the sub-folders under the 'lib' folder. In this case the 'lib/cmp' folder has been used to store the file using a resonable name - which in this case might be 'tle2031.sub'. The content of this file is as follows:

```
* TLE2037 OPERATIONAL AMPLIFIER "MACROMODEL" SUBCIRCUIT
* CREATED USING PARTS RELEASE 4.03 ON 06/25/90 AT 13:17
* REV (N/A)        SUPPLY VOLTAGE: +/-15V
* CONNECTIONS:   NON-INVERTING INPUT
*                  | INVERTING INPUT
*                  | | POSITIVE POWER SUPPLY
*                  | | | NEGATIVE POWER SUPPLY
*                  | | | | OUTPUT
*                  | | | | |
.SUBCKT TLE2037  1 2 3 4 5
*
  C1   11 12 75E-12
  C2    6  7 50.00E-12
  C3   87  0 134E-9
  CPSR 85 86 1.99E-6
  DCM+ 81 82 DX
  DCM- 83 81 DX
  DC    5 53 DX
  DE   54  5 DX
  DLP  90 91 DX
  DLN  92 90 DX
  DP    4  3 DX
  ECMR 84 99 (2,99) 1
  EGND 99  0 POLY(2) (3,0) (4,0) 0 .5 .5
  EPSR 85 0 POLY(1) (3,4) -60E-6 2E-6
  ENSE 89 2 POLY(1) (88,0) 20E-6  1
  FB 7 99 POLY(6) VB VC VE VLP VLN VPSR 0 68.25E6 -60E6 60E6 60E6 ...
  GA    6  0 11 12 26.08E-3
  GCM   0  6 10 99 7.349E-9
  GPSR 85 86 (85,86) 100E-6
  GRC1 3 11 (3,11) 26.08E-3
  GRC2 3 12 (3,12) 26.08E-3
  GRE1 13 10 (10,13) 10.04E-3
  GRE2 14 10 (10,14) 10.04E-3
  HLIM 90  0 VLIM 1K
  HCMR 80 1 POLY(2) VCM+ VCM- 0 1E2 1E2
  IRP 3 4 3.43E-3
  IEE  10  4 DC 375.0E-6
  IIO  2 0 6E-9
```

```
 I1 88  0 1E-21
 Q1   11   89 13 QX
 Q2   12   80 14 QX
 R2    6   9 100.0E3
 RCM  84 81 1K
 REE  10 99 533.3E3
 RN1  87  0 8.75E7
 RN2  87 88 235
 RO1   8   5 25
 RO2   7 99 26
 VCM+ 82 99 12.4
 VCM- 83 99 -12.4
 VB    9   0 DC 0
 VC    3 53 DC 2.500
 VE   54   4 DC 2.200
 VLIM  7   8 DC 0
 VLP  91   0 DC 40
 VLN   0 92 DC 40
 VPSR 0 86 DC 0
.MODEL DX D(IS=800.0E-18)
.MODEL QX NPN(IS=800.0E-18 BF=12.50E3)
.ENDS
```

This is clearly a more complicated model than the diode model. Now to include this specific model in a simulation just choose an OPAMP as usual but instead of selecting a LT1001 device as previously, choose the 'opamp2' device. After placing the device in the schematic right-click it and enter 'tle2037' in the 'value' field. The basic OPAMP symbol now links to the TLE2037 description and everything is almost ready for simulation. The only thing remaining to do is to add the 'tle2031.sub' file to the library search path of LTspice. This is done by adding the SPICE directive '.LIB tle2031.sub' to the schematic window. Once this has been done the same simulation set-up as for the LT1001 device may be used to get the new result using the TLE2037. This is shown in Figure 9.

Compared to the results in Figure 7 the TEL2037 is found to perform much better. Here the high frequency 3dB cut-off frequency is found to approximately 9.5MHz which implies that the TLE2037 is 126 times faster than the LT1001.

4-1-2 Adding models the structured way

If LTspice is used extensively a wide list of different components might have to be added. Assuming that the just mentioned dirty approach is used a wast number of '.sub' files will be stored. A more structures approach is to add specific user libraries to LTspice instead. This is done by first adding two extra folders to the library structure

- /LTC/SwCADIII/lib/sub/myLib
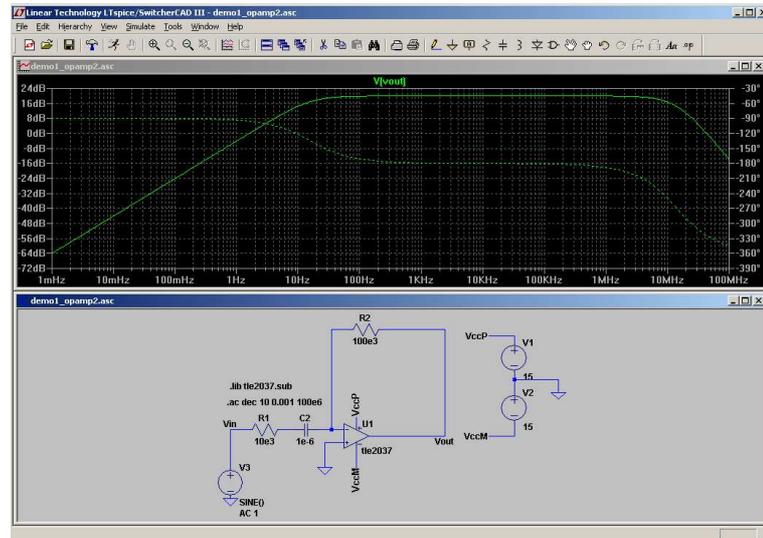
FIGURE 9: *Schematic entry using an OPAMP (demo3_opamp3.asp).*

- /LTC/SwCADIII/lib/sym/myLib

When looking for a specific model to add to a schematic then the 'myLib' folder will now be visible along with the default ones in LTspice. After adding the two folders it is a simple matter of downloading the SPICE model (<mod>) file for the device and copy this into a new file named '<name>.lib'. This file has to be placed in the 'sub/myLib' folder. The next step is to add a symbolic view to the new device. It is possible to draw one from scratch in LTspice but it is far easier to copy an already existing one. So in the case of an OPAMP, find a suitable device from the 'Opamps' folder and copy this. If for instance the TL1001 device has the same symbol as you would need then simply copy '/lib/sub/Opamps/TL1001.asy' as '/lib/sym/myLib/<mod>.asy'.

The last thing to do is to ensure that the new symbol also links to the correct SPICE model. This is done by opening the new symbol and then selecting 'Edit -> Attributes -> Edit Attributes'. In the resulting window all instances of (in this case) 'TL1001' has to be replaced by <mod>. The resulting content should look something like this:

```
Prefix       X
SpiceModel   myLib\<name>.lib
Value        <mod>
Value2       <mod>
Specline
Specline2
Description  Whatever comes to mind
```

If you need to add further components to your own library file simply do the same once again. But remark that the new SPICE file may be added to the '<name>.lib' file. If you have a

number of OPAMPs from for instance Texas Instruments then all of these may be added in a single file called (for instance) `'TI_OPAMP.lib'`. This way it is very easy to keep track of all the SPICE models that you may have to add your self.

5   RUNNING `LTspice` ON LINUX/UNIX TERMINALS

...