ExtractModel: Examples

Derk Reefman

January 29, 2016

1 Introduction

The user interface to ExtractModelis, as a friend explained, quite Spartan. This document tries to explain the interface and modes of operation with a number of examples that hopefully clarify the operation of ExtractModel. In the below, a triode, a penthode without secundary emission, and a penthode with secundary emission will be modelled as examples.

This document describes ExtractModel version 3.0.

1.1 Installation

ExtractModelis a straightforward executable that needs to be run in a cmd window, or an XTerm under Linux. Hence, installation is easy and is simply copying the executable in a directory from which it can be run. It is advisory to have a PATH variable to that directory such that ExtractModel can be run from any directory. In the examples below, it will be assumed this is the case. It is most convenient to run ExtractModel in the directory that also contains the .utd data files. This will cause the output files produced by ExtractModel to also end up in that same directory.

As ExtractModel uses Gnuplot as its plotting interface, it is important to have Gnuplot installed. Gnuplot can be downloaded fromwww.gnuplot.info for many platforms; see this website for installation of the package. ExtractModel needs to know how Gnuplot can be invoked as a command; for a Windows7 OS this is typically C:\Program Files (x86)\gnuplot\bin\wgnuplot.exe. Under most Linux variants gnuplot is located in /usr/bin/gnuplot.

2 The program `ExtractModel`

A program called ExtractModel has been written to fit the models discussed in the previous sections to a dataset obtained by the uTracer of Ronald Dekker. The uTracer can generate a file with extension `.utd` that contains a table of all measured values, which serves as input to `ExtractModel`. The uTracer can perform different kinds of parameter sweeps. For diodes there are multiple sweeps that can be used; the primary requirement is that I_a is measured as function of V_a. To remain consistent with sweeps used for other tubes it is advised to stick to the 'I(Va,Vg) with Vs, Vh constant' sweep. It is sufficient to enter only one V_g value (the value of which is irrelevant as V_g is not used; as long as $V_g < 0$ to meet the `uTracer` requirements).

For triodes, the sweep that is used by `ExtractModel` is the 'I(Va,Vg) with Vs, Vh constant' or `Vs=cst` sweep for triodes or penthodes. This sweep performs a measurement of the anode current as a function of anode voltage, and does so for various grid voltages V_{g1}. While this sweep is sufficient to fit all parameters for a triode model, for a penthode model different values of V_s are required, too. This is done by repeating the 'I(Va,Vg) with Vs, Vh constant' sweep for various screen voltages V_s, and storing the data files separately. In addition the 'I(Va=Vs,Vg) with Vh constant' or `Vs=Va` sweep is used for penthodes to generate the triode model of the penthode, to generate the initial parameter values (see Sec. ??). The program is written in fortran; microsoft fortran is used for compilation under Windows, and gfortran for compilation under Linux. The user interface is simple text based, aiding in easy portability over different platforms. No attempt is made for any algorithmic optimization for speed as focus has been on both robustness and ease of model development.

2.1 Initialization file description

An initialization file (by default 'Model.ini'; different names can be used as command line arguments) is required to provide all necessary inputs to `ExtractModel`:

```
C:\Program Files (x86)\gnuplot\bin\wgnuplot.exe !location of gnuplot executable
3 !number of .utd files
data1 !as many .utd files as the number above indicates
data2 !as many .utd files as the number above indicates
data3 !as many .utd files as the number above indicates
P !D(Diode),T(Triode), X(Penthode/triode mode), P(Penthode), B(beam tetrode), H(Heptode), F(Pentode wit
Derk !Model used; either Koren, Derk or DerkE
2. !Pmax in fit !fitting only data points below P=Pmax curve
0 !Vg Offset !Offset to grid voltage
```

Optional is an entry

```
Icmax = xxx
RdStart = <StartParfilename>
```

where `xxx` is the maximum cathode current in the data file that is used for fitting. If this entry is left out, there is no limit to the cathode current used. `StartParfilename` is the name of a file that contains initial parameter estimates that will be used - rather than the values that would be generated automatically by `ExtractModel`. The format of the file should be identical to the format that the `.par` output files have.
Following is a line-by-line description of the input required in the initialization file:

1. ExtractModeluses gnuplot [Gnu13] as the utility to create plots of the data and the curves that have been fitted to the data. Hence, ExtractModel needs to know where the executable of Gnuplot is located.

2. The number of .utd datafiles, see Sec. 2.2 for input filename conventions. For penthodes and beam tetrodes, multiple files with sweeps using different values for Vs are required. The maximum number of files is set to 15. For triodes the number of files can be larger than 1 if multiple tubes have been measured, and the fit needs to represent an average of the different tubes. As tubes can differ easily by 10% in current for identical voltages, this is a nice way to get a representation of the ‘average tube’.

3. The names of the data files. The extension .utd must be left out. If fitting a penthode / beam tetrode, the last file can be the file as measured in ‘triode mode’, i.e. in a Vs=Va sweep. If a Vs=Va data file is listed, but *not* as the last file, this will stop ExtractModel.

4. The type of tube. Currently the choice of tubes is

   (a) D - for a diode

   (b) T - for a triode

   (c) X - for a penthode, but measured with the 'I(Va=Vs, Vg) with Vh constant' sweep. This signals ExtractModelto sum the anode and screen currents, and fit a triode model

   (d) Y - for a hepthode, measured with the 'I(Va=Vs, Vg) with Vh constant' sweep (and for a specific Vg3). The use of this tube designator signals ExtractModel to create a parameter file Triode_g3.par that is used as input for starting parameters in the heptode model fitting (see Sec. ??)

   (e) P - for a penthode. For historical (programming) reasons, this tube designation is used if secundary emission is not playing any role. It can be used with both the Derk and DerkE models.

   (f) V - for a variable-mu penthode. For historical (programming) reasons, this tube designation is used if secundary emission is not playing any role. It can be used with both the Derk and DerkE models.

   (g) B - for a beam tetrode. For historical (programming) reasons, this is the tube designation that includes secundary emission effects. It can be used with both the Derk and DerkE models.

   (h) H - for a Hepthode.

   (i) F - for a penthode with g3 modulation

5. Model used for fitting. Known model names are 'Koren', 'Derk' (see Sec. ??) ('DerkI' is also possible - see App. ??) and 'DerkE' (see Sec. ??). For diodes and triodes, the only available model is 'Koren'; whenever a different name is given, it is still fitted according to the Koren model. For hepthodes, the only available model is ‘Derk’.

6. The maximum anode dissipation Pmax used in fitting the data; only data for which $I_a V_a < P_{\max}$ is used to fit the data to.

7. The uTracer can only generate grid voltages between -49 and 0 volts. VgOffset is any constant voltage that is added to that by an external source to create grid voltages below -49V.

2.2 Input file naming

The input filename conventions are not very strict for diodes and triodes, except for the fact that the filename extension should be `.utd`. A choice the author makes is to just call the file to the tube naming, *e.g.*, `ECC81.utd` or `PCF80C.utd`. For penthodes, more files are needed. Again, there are no strict rules to naming. A choice the author makes is to have a naming that indicates whether the tube has been measured in triode mode or with a fixed V_{g2} bias, *e.g.* `PF86_triode.utd`, which would be a PF86 measured in triode mode, and `PF86_200.utd` which would be a PF86 measured at $V_{g2} = 200V$. In all cases, the exact parameter values will be read from the data file itself.

For heptodes the naming is no longer free. Heptode data files need to be generated with both V_{g1} as a variable, and with V_{g3} as a variable, and the uTracer does not allow storing these as separate variables. The trick that needs to be played is to run several 'I(Va=Vs, Vg) with Vh constant' sweeps, where either V_{g1} or V_{g3} is connected to the V_g terminal. Naming the datafile has therefore has to be according to the following convention:

filename = `abcdefgh_gn_xx.x.utd`

where n denotes the grid (either 1 or 3); and xx.x denotes the constant voltage that has been applied to that grid during the scan. An example is `ECH81_100_g3_10.0.utd`. This is a dataset where V_{g2} is kept at a constant voltage of 100V ('I(Va, Vg) with Vs, Vh constant' sweep), which will be reflected in the data it self. The first grid $g1$ is connected to the V_g terminal, and V_{g3} has been kept constant by an external voltage source to $-10.0V$. This voltage is not represented in the dataset, and the only way for `ExtractModel` to know this voltage is the filename. Another example is `ECH81_triode_g1_1.5.utd`. This represents a 'I(Va=Vs, Vg) with Vh constant' sweep, where an external source has been used to set $V_{g1} = -1.5V$. This implies that $g3$ of the heptode has been connected to the V_g terminal of the uTracer, and, hence, that the data tabulated in the file, has a variable V_{g3}.

2.3 Output files

The most important outputfile is 'data1.cir', which is a concatenation of the first data file name in the initialization file, and the '.cir' extension. This is the LTSpice file containing the model of the tube that has been fitted.

```
**************************************************
 .SUBCKT ECC85 1 2 3; A G C;
* ExtractModel V .995
* Model created: 29-Nov-13
X1 1 2 3 TriodeK MU= 94.0 EX=1.148 kG1=  59.9 KP= 230.9 KVB=3805. RGI=2000
+ CCG=0.0P  CGP=0.0P CCP=0.0P  ;
.ENDS

**************************************************
.SUBCKT TriodeK 1 2 3; A G C
E1 7 0 VALUE=
+{V(1,3)/KP*LOG(1+EXP(KP*(1/MU+V(2,3)/SQRT(KVB+V(1,3)*V(1,3)))))}
RE1 7 0 1G
G1 1 3 VALUE={0.5*(PWR(V(7),EX)+PWRS(V(7),EX))/kG1}
RCP 1 3 1G    ; TO AVOID FLOATING NODES IN MU-FOLLOWER
C1 2 3 {CCG}  ; CATHODE-GRID
```

```
C2 2 1 {CGP}  ; GRID-PLATE
C3 1 3 {CCP}  ; CATHODE-PLATE
D3 5 3 DX     ; FOR GRID CURRENT
R1 2 5 {RGI}  ; FOR GRID CURRENT
.MODEL DX D(IS=1N RS=1 CJO=10PF TT=1N)
.ENDS TriodeK
```

The `.cir` file contains the Spice definition of the model, DiodeK, TriodeK, PenthodeD, PentodeDE, PenthodeB, PenthodeBE, or HeptodeD, and the tube specific description of the parameterset feeding the model. Also note that the signum function (see Eq. (**??**)) is implemented consistenly for all models. When a library of tubemodels is made, the Spice definitions of the models need to be copied only once, thereafter only the tube specific parameter description needs to be copied. The Spice code generated for a Triode is based on the code as proposed by Koren ([Kor01]). The code contains parametric descriptions of the inter-electrode capacitances, that are all set to zero in the output of ExtractModel. The user will have to add these based on datasheet reported values. Other files that are created are a `Model.par` file, which contains the refined parameters, and, if a triode type 'X' was fitted, a `Triode.par` file which can be used as input for fitting a penthode model later. Further output files are the files that are fed to `Gnuplot` to do the plotting. The plot file for display of the anode currents is the `.plt` file; the plot file with extension `.spl` loads and displays the screen currents. Note that this extension is also used by Shockwave. These files can be loaded in Gnuplot any time later to review the fitting results, independent of `ExtractModel`.

2.4 Model fitting tips

Feedback from users working with `ExtractModel` has generated some practical tips for a good fit, and listed in the following. `ExtractModel`already incorporates, or will incorporate, some of the tips below and generate warnings when the data to be fitted is insufficient (sec. 2.4.1), when the tube displays saturation effects (sec. 2.4.2), or when the dynamic range is too large for a good fit (sec. 2.4.3). `ExtractModel` will leave it to the discretion of the user to continue - but results may not be correct.

2.4.1 Number of different values for V_g

For a triode fit, different values for V_g are required to get a good estimate of especially the parameter k_p. Depending on how accurate the measurements are, at least 5 different values for V_g are required, and at least a few values are needed that give an accurate description of the cut-off behaviour, which helps in good estimates of k_{VB}. This means that several V_g values should be present that have a region where the anode current is zero within the precision of the uTracer. Preferably (see also Sec. 2.4.3 one would have a range of V_g values which, at the maximum anode voltage and highest V_g, result in an anode current of about 150% of the maximum anode current, and at the maximum anode voltage and lowest V_g about 20% of that. Fig. **??** gives an idea for two different triodes. For regular penthodes, at least 3 different values of V_{g2} are required, and per V_{g2} value at least 3 different values for V_g. Variable-mu pentodes - on the contrary - require a large number of different values for v_{g1}, preferably evenly spread over the high-mu and low-mu characteristic of the pentode.

2.4.2 Saturation

Saturation of a tube is a phenomenon where the cathode is not capable of delivering the current that theoretically would flow, see also [Spa48] for a description of this phenomenon. For triodes, the anode current as function of anode voltage show a positive curvature (the curve becomes ever steeper with increasing anode voltage). Severe saturation will cause the anode current to show negative curvature (*i.e.*, the steepness of current increase as a function of voltage becomes less). It is important to remove curves that display this behaviour while fitting. `ExtractModel` will try to fit the measurements (even though in very severe cases, it gives a warning); and as a result the fitted parameter values may not represent the true behaviour.

For penthodes, saturation may occur either due to excessive screen current at low anode voltage, or, similar as with a triode, due to excessive anode current at very high anode voltages. Saturation at low anode voltages can be detected by inspection of the screen current, which, at low anode voltages, should be a concave curve. When this curve becomes convex, saturation plays a role. Especially when secundary emission is included in the model fitting, `ExtractModel` may easily be fooled by the convex part in the screen current, and mistakenly interprets it as secundary emission (which also causes convex curvature of the screen current!).

As a rule of thumb, one should never have data with cathode currents that largely exceed the maximum values as published in the datasheets. For new tubes of reknowned brands, the cathode can typically deliver 150% of the tabulated maximum value. This is often not true for some russian NOS stubes, and old, used tubes that already spent a good deal of their life in heavy duty.

2.4.3 Dynamic range

The previous section Sec. 2.4.2 already implicitly gave an upper boundary for the current measurements. Curves reaching lower current values are required for covering the full space that is needed for a good parameter fit - however, if currents are very close to zero over the full anode voltage range, this adds no information (and actually detracts `ExtractModel`). Typically, a good dynamic range is between 5 and 10, meaning that the highest current reached for the V_g, V_{g2} combination that gives the lowest overall current, should be no less than about 1/10 to 1/5 of the maximum current over all parameters. Good examples for this are the curves presented throughout this document, where the dynamic range never exceeds 7.

For vaiable-mu pentodes, care should be taken that all curves cover the high-mu and low-mu part of the characteristic evenly. A dataset with many values for the low-mu part, and few for the high-mu part will give an untrustworthy result. Note that this is a mistake easily made, since an evenly distributed set of grid values V_{g1} will likely correspond to a few high-mu values, and many more low-mu values.

2.4.4 Data range

In particular with heptodes, it is important to play a bit with the maximum anode dissipation and/or maximum cathode current of the data used for fitting. The heptode model can easily get caught in a false minimum if the maximum power/current used in fitting is too large. This leads to the interesting effect that the fit can sometimes (even for high currents) be better if the data used in the fit is restricted to values of 0.5W or even a bit below for the anode dissipation. Until initial parameter estimation is improved, this inconvenience has to be accepted.

3 Fitting a diode

Fitting a diode is theoretically the easiest application for `ExtractModel`. It requires a date file that has the measured Ia-Va curve of the diode; see Sec. 2 how to generate such a file. To fit the data in the file, an input file will have to be made as shown below:

```
C:\Program Files (x86)\gnuplot\bin\wgnuplot.exe
1   ! we only have 1 file
EZ80 ! the name of the file (without extension)
D  ! We fit a diode
Koren ! We use the Koren model
2 !Only data resulting in Pa < 2W are used in the fit
0 !Vg Offset
```

To execute `ExtractModel`, open a command window (cmd under Windows, xterm under Linux) and execute `ExtractModel`. A Gnuplot window pops up that shows the measured data points, and the curve based on the refined parameters. Under Windows, there is a second pop-up with a Pause button - press OK and it will disappear. The purpose of this pop-up is to show the fit results for any other data files (we will explore this later).

The directory will now contain the following files:

```
C:\Users\nlv13348\Dropbox\Spice\ExtractModel\ECC81>dir
 Directory of C:\Users\nlv13348\Dropbox\spice\extractmodel\EZ80

04/13/2014  02:04 PM    <DIR>          .
04/13/2014  02:04 PM    <DIR>          ..
11/08/2013  03:27 PM               891 EZ80.cir
11/08/2013  03:27 PM               992 EZ80.dat
11/08/2013  03:27 PM             1,500 EZ80.fit
11/08/2013  03:27 PM               106 EZ80.log
11/08/2013  03:27 PM               525 EZ80.plt
11/02/2013  10:15 AM             3,264 EZ80.utd
11/08/2013  03:26 PM               175 Model.ini
11/08/2013  03:27 PM                69 Model.Par
               8 File(s)          7,522 bytes
               2 Dir(s)  93,593,587,712 bytes free
```

The most important file created is the `EZ80.cir` file, which is the Spice circuit description:

```
****************************************************

.SUBCKT EZ80 1 2; A C;
* ExtractModel V .92
* Model created: 08-Nov-13
X1 1 2 DiodeK  EX=1.410 k = 975.4 eps=   -.1
+ CCP=0.0P  ;
```

```
.ENDS
*****************************************************

.SUBCKT DiodeK 1 2; A C
G1   1 2  VALUE = {0.5/k*(PWR(V(1,2)+eps,EX)+PWRS(V(1,2)+eps,EX))}
C3   1 2  {CCP}   ; CATHODE-PLATE       A  - C
.ENDS DiodeK
```

This file contains the generic description of a diode (a lot of this description is copied from [Kor01]) as `.SUBCKT DiodeK`; and the model which is specific to the EZ80. The definitions of the capacitances is for the user; please edit these yourself based on values that are reported in the handbooks (see [Phi13] for an almost exhaustive collections of datasheets).

EZ80.plt is the `Gnuplot` command file that generates the plot. This file can always be loaded by Gnuplot, independent of `ExtractModel`, at a later moment. The two files `EZ80.dat` and `EZ80.fit` contain the original data (as presented in `EZ80.utd` and the model curve, respectively, that are used by the command file `EZ80.plt`. Finally, there is a `Model.par` file, which contains the values of the parameters.

Unfortunately, `uTracer` has not been designed for the parameter space most relevant to diodes: high current and low voltage. As a result, the data for a diode will be rather noisy. To make any fitting parameters less dependent on this noise, multiple measurements can be made (with identical settings in the `uTracer`) resulting in multiple `.utd` files. A typical `.ini` file corresponding to this would be:

```
C:\Program Files (x86)\gnuplot\bin\wgnuplot.exe
3   ! we  have 3 files
EZ80_1 ! the name of the file (without extension)
EZ80_2 ! the name of the file (without extension)
EZ80_3 ! the name of the file (without extension)
D  ! We fit a diode
Koren ! We use the Koren model
2 !Only data resulting in Pa < 2W are used in the fit
0 !Vg Offset
```

Here `EZ80_1, EZ80_2` *etc.* are the different files that have been collected. The remainder of the process is identical to fitting only one data file, and results in a file `EZ80_1.cir` with the Spice circuit describing the diode.

4 Fitting a triode

For fitting a triode, we first have to create the file `Model.ini`, which contains the input that `ExtractModel` will use. Below is an example input file, which is based on the assumption that in the directory where `ExtractModel` is called, a file ECC81.utd exists that contains Ia-Va curves for various values Vg of the grid bias:

```
C:\Program Files (x86)\gnuplot\bin\wgnuplot.exe
1   ! we only have 1 file
ECC81 ! the name of the file (without extension)
T  ! We fit a triode
Koren ! We use the Koren model
```

```
2 !Only data resulting in Pa < 2W are used in the fit
0 !Vg Offset
```

To execute `ExtractModel`, open a command window (cmd under Windows, xterm under Linux) and execute `ExtractModel`.

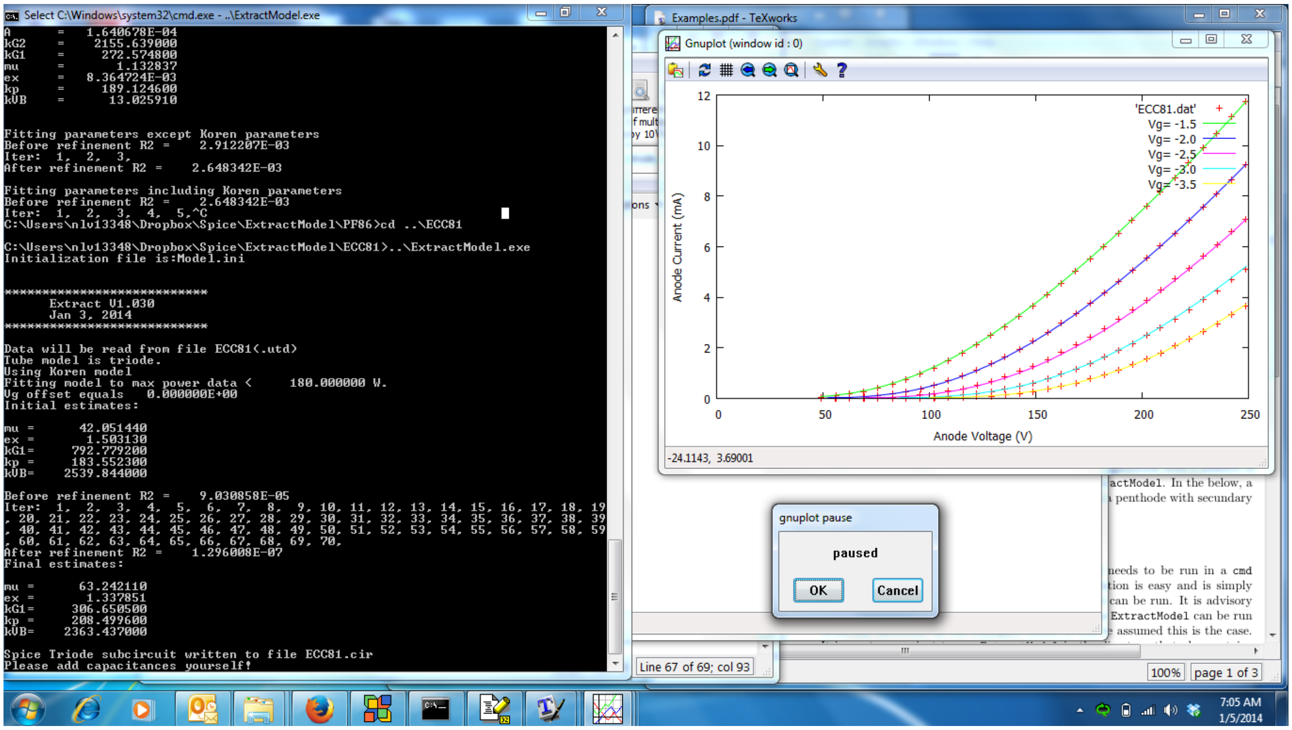

Figure 1: Output of `ExtractModel` for the ECC81 example

In the cmd window we see a lot of output: the version of `ExtractModel`, confirmation of the contents of the inputfile; the parameter values based on the first estimation; the sum of squares R_2 before refinement; the iteration number and final sum of squares. The last bit of information is the parameter values after the refinement, and the fact that a Spice triode subcircuit has been written to file `ECC81.cir`. This file name has been constructed from the input file name, with the addition of the `.cir` extension.

Next to that, a Gnuplot window pops up that shows the measured data points, and the curve based on the refined parameters. Under Windows, there is a second pop-up with a Pause button - press OK and it will disappear. The purpose of this pop-up is to show the fit results for any other data files (we will explore this later).

The directory will now contain the following files:

```
C:\Users\nlv13348\Dropbox\Spice\ExtractModel\ECC81>dir
 Volume in drive C has no label.
 Volume Serial Number is FA5B-D21C
```

```
 Directory of C:\Users\nlv13348\Dropbox\Spice\ExtractModel\ECC81

01/05/2014  07:40 AM    <DIR>          .
01/05/2014  07:40 AM    <DIR>          ..
01/05/2014  07:38 AM               773 ECC81.cir
01/05/2014  07:38 AM             4,960 ECC81.dat
01/05/2014  07:38 AM             4,813 ECC81.fit
01/05/2014  07:38 AM               304 ECC81.plt
11/15/2013  03:13 PM            16,067 ECC81.utd
11/17/2013  11:19 AM               176 Model.ini
01/05/2014  07:38 AM               110 Model.Par
01/05/2014  07:38 AM               110 Triode.par
               8 File(s)         27,313 bytes
               2 Dir(s)  92,434,653,184 bytes free

C:\Users\nlv13348\Dropbox\Spice\ExtractModel\ECC81>
```

The most important file created is the `ECC81.cir` file, which is the Spice circuit description:

```
****************************************************
.SUBCKT ECC81 1 2 3; A G C;
*      Extract V1.030
*  Model created:  5-Jan-2014
X1 1 2 3 TriodeK MU= 63.24 EX=1.338 KG1= 306.7 KP= 208.5 KVB= 2363. RGI=2000
+ CCG=0.0P  CGP=0.0P CCP=0.0P  ;
.ENDS

****************************************************
.SUBCKT TriodeK 1 2 3; A G C
E1 7 0 VALUE=
+{V(1,3)/KP*LOG(1+EXP(KP*(1/MU+V(2,3)/SQRT(KVB+V(1,3)*V(1,3)))))}
RE1 7 0 1G
G1 1 3 VALUE={0.5*(PWR(V(7),EX)+PWRS(V(7),EX))/KG1}
RCP 1 3 1G    ; TO AVOID FLOATING NODES IN MU-FOLLOWER
C1 2 3 {CCG}  ; CATHODE-GRID
C2 2 1 {CGP}  ; GRID-PLATE
C3 1 3 {CCP}  ; CATHODE-PLATE
D3 5 3 DX     ; FOR GRID CURRENT
R1 2 5 {RGI}  ; FOR GRID CURRENT
.MODEL DX D(IS=1N RS=1 CJO=10PF TT=1N)
.ENDS TriodeK
```

This file contains the generic description of a triode (a lot of this description is copied from [Kor01]) as `.SUBCKT Triode`; and the model which is specific to the ECC81. The definitions of the capacitances is for the user; please edit these yourself based on values that are reported in the handbooks (see [Phi13] for an almost exhaustive collections of datasheets).

ECC81.plt is the `Gnuplot` command file that generates the plot. This file can always be loaded by Gnuplot, independent of `ExtractModel`, at a later moment. The two files `ECC81.dat` and `ECC81.fit` contain the original data (as presented in `ECC81.utd` and the model curve, respectively, that are used by the command file `ECC81.plt`.

Finally, there are two `.par` files, which in this case are identical, and contain the values of the parameters.

5 Fitting a penthode

In order to fit a penthode. several files need to be collected:

1. A data file created with the $V_s = V_a$ sweep
2. Several data files created with the $V_s = cst$ sweep, for various values of V_s.

In order to fit a penthode, first the parameters describing the penthode wired as a triode need to be determined - this is also why we need a data file for a $V_s = V_a$ sweep. To do this, we create an initialization file which we call `Triode.ini` for `ExtractModel` that looks as follows:

```
C:\Program Files (x86)\gnuplot\bin\wgnuplot.exe
1
PF86_triode
X !to sum Ia and Is to fit the triode model to
Derk
2. !Pmax in fit
0 !Vg Offset
```

Note that we use 'Derk' for the model definition - there is no model 'Derk' for a Triode, but we want t osee how `ExtractModel` handles this. The contents of the directory are now:

```
C:\Users\nlv13348\Dropbox\Spice\ExtractModel\PF86>dir
 Volume in drive C has no label.
 Volume Serial Number is FA5B-D21C

 Directory of C:\Users\nlv13348\Dropbox\Spice\ExtractModel\PF86

01/05/2014  09:06 AM    <DIR>          .
01/05/2014  09:06 AM    <DIR>          ..
01/22/2013  08:28 PM            18,888 pf86_200.utd
01/22/2013  08:28 PM            18,888 pf86_250.utd
01/22/2013  08:28 PM            18,888 pf86_300.utd
01/22/2013  08:03 PM            22,019 pf86_triode.utd
01/05/2014  09:04 AM               248 Triode.ini
               7 File(s)         79,406 bytes
               2 Dir(s)  92,425,502,720 bytes free

C:\Users\nlv13348\Dropbox\Spice\ExtractModel\PF86>
```

where the files pf86_xxx.utd are the files with a $Va = cst$ sweep and pf86_triode the file with a $V_s = V_a$ sweep. We now run ExtractModel, with Triode.ini as argument (we could have called the initialization file anything - if we would have called it Model.ini ExtractModel could have been run without argument).

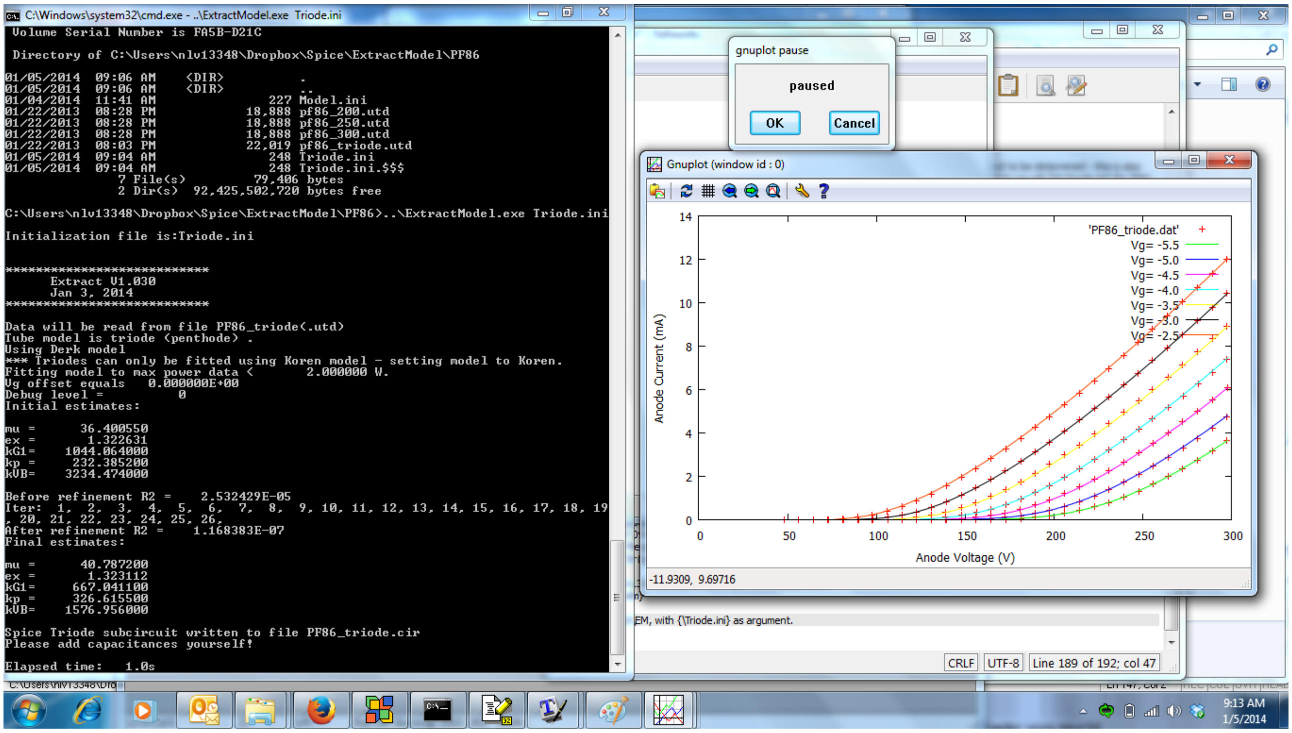

Figure 2: Output of ExtractModel for the PF86 example with the Triode.ini initialization file.

The output is very similar to the output produced for the ECC81 from the previous section, note the remark *** Triodes can only be fitted using Koren model - setting model to Koren.

Now the penthode model has to be created. For this a new initialization file which we call Penthode.ini needs to be created with the following contents:

```
C:\Program Files (x86)\gnuplot\bin\wgnuplot.exe
4
pf86_200
pf86_250
pf86_300
PF86_triode
P
Derk
2. !Pmax in fit
0 !Vg Offset
```

Note that we use 'P' for the model (penthode without secundary emission) and 'Derk' for the model describing the penthode. We could have used the 'DerkE' model as well - but it will not give as good a fit as the 'Derk' model. Also note that the $V_s = V_a$ dataset is *last* in the row of datasets - it will result in an error when it is *not* last. We now run `ExtractModel` with `Penthode.ini` as argument.

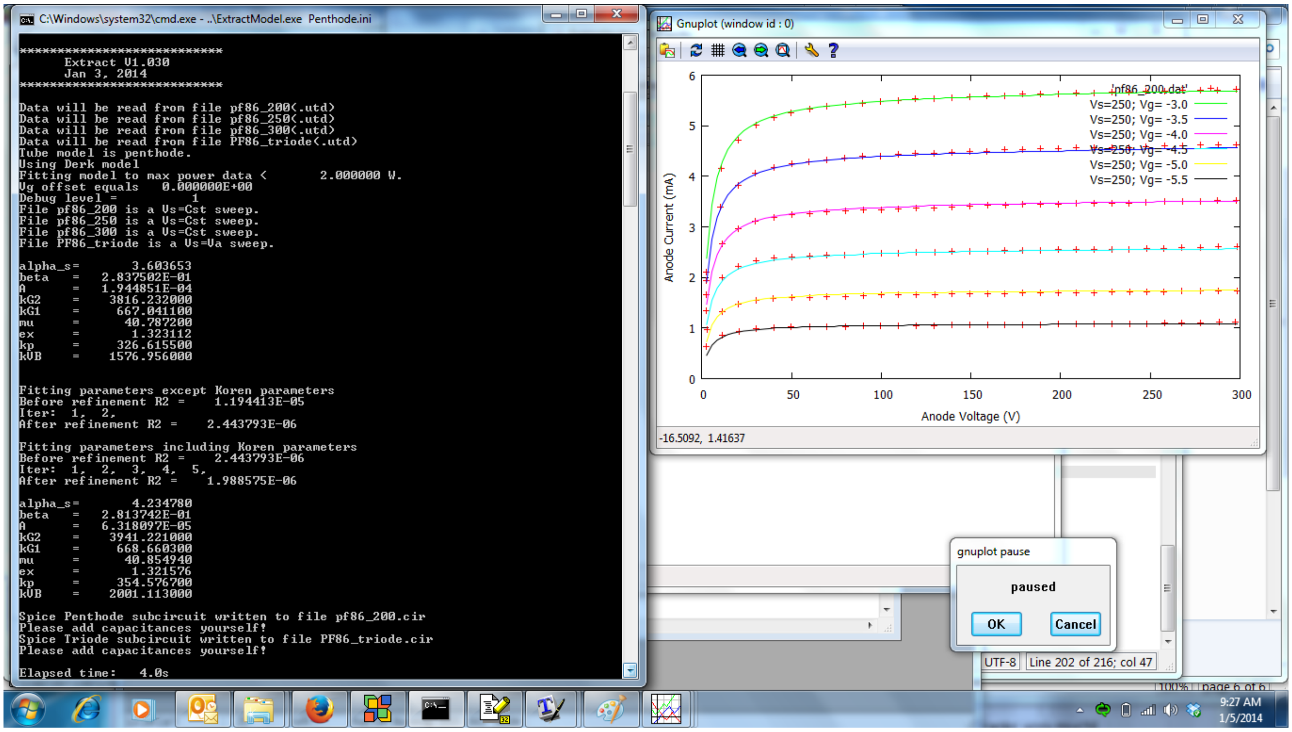

Figure 3: Output of `ExtractModel` for the PF86 example with the Penthode.ini initialization file.

In the output we can identify that `ExtractModel` first does a refinement of all parameters except the Koren parameters - `ExtractModel` reads the Koren parameters from the `Triode.par` file that was generated in the first run of `ExtractModel`. `ExtractModel` also reports that two spice files are generated: a file `pf86_200.cir` which contains the spice model for a penthode, and a file `PF86_triode.cir` which contains the spice circuit for the PF86 strapped as a triode (*i.e.*, the screen is connected to the anode). There is also the `Gnuplot` pop-up - pressing OK will now display the screen currents and fit for the first dataset. Repeatedly pressing 'OK' will display the anode and screen currents of next dataset *etc.*, until we have the last dataset, which is the $V_s = V_a$ dataset.

The directory now contains the following output files:

```
C:\Users\nlv13348\Dropbox\Spice\ExtractModel\PF86>dir
 Volume in drive C has no label.
 Volume Serial Number is FA5B-D21C

 Directory of C:\Users\nlv13348\Dropbox\Spice\ExtractModel\PF86
```

```
01/05/2014  09:26 AM    <DIR>          .
01/05/2014  09:26 AM    <DIR>          ..
01/05/2014  10:10 AM               234 Model.Par
01/05/2014  09:26 AM               278 Penthode.ini
01/05/2014  10:10 AM             1,254 pf86_200.cir
01/05/2014  10:10 AM             5,952 pf86_200.dat
01/05/2014  10:10 AM             5,680 pf86_200.fit
01/05/2014  10:10 AM               367 pf86_200.plt
01/05/2014  10:10 AM            16,626 pf86_200.sft
01/05/2014  10:10 AM               385 pf86_200.spl
01/22/2013  08:28 PM            18,888 pf86_200.utd
01/05/2014  10:10 AM             5,952 pf86_250.dat
01/05/2014  10:10 AM             5,680 pf86_250.fit
01/05/2014  10:10 AM               367 pf86_250.plt
01/05/2014  10:10 AM            16,626 pf86_250.sft
01/05/2014  10:10 AM               385 pf86_250.spl
01/22/2013  08:28 PM            18,888 pf86_250.utd
01/05/2014  10:10 AM             5,952 pf86_300.dat
01/05/2014  10:10 AM             5,680 pf86_300.fit
01/05/2014  10:10 AM               367 pf86_300.plt
01/05/2014  10:10 AM            16,626 pf86_300.sft
01/05/2014  10:10 AM               385 pf86_300.spl
01/22/2013  08:28 PM            18,888 pf86_300.utd
01/05/2014  10:10 AM               763 PF86_triode.cir
01/05/2014  10:10 AM             6,944 PF86_triode.dat
01/05/2014  10:10 AM             6,507 PF86_triode.fit
01/05/2014  10:10 AM               436 PF86_triode.plt
01/05/2014  10:10 AM            19,045 PF86_triode.sft
01/05/2014  10:10 AM               456 PF86_triode.spl
01/22/2013  08:03 PM            22,019 pf86_triode.utd
01/05/2014  09:04 AM               248 Triode.ini
01/05/2014  09:12 AM               110 Triode.par
              32 File(s)        202,463 bytes
               2 Dir(s)  92,413,743,104 bytes free
```

Files with the `.plt` extension are the plot files that can be load in `Gnuplot` and will show the anode measured and fitted current. Files with the `.spl` extension will do the same for the screen current. Files with the `.dat` and `.fit` extension are the data files used by the `.plt` files, and files with the `.sft` extension are used by the `.spl` files.

The `pf86_200.cir` file is the file with the Spice model describing the penthode:

```
****************************************************
.SUBCKT pf86_200 1 2 3 4 ; A G2 G1 C;
*       Extract V1.030
* Model created:  5-Jan-2014
X1 1 2 3 4 PenthodeD  MU= 40.9 EX=1.322 kG1= 668.7 KP= 354.6 kVB =  2001.1 kG2= 3941.2
+ Ookg1mOokG2=.12E-02 Aokg1=.94E-07 alkg1palskg2=.12E-02 be=   .28 als=  4.23 RGI=2000
```

```
+ CCG1=0.0P  CCG2 = 0.0p CPG1 = 0.0p CG1G1 = 0.0p CCP=0.0P  ;
.ENDS

****************************************************
.SUBCKT PenthodeD 1 2 3 4; A G2 G1 C
RE1  7 0  1MEG    ; DUMMY SO NODE 7 HAS 2 CONNECTIONS
E1 7 0 VALUE=
+{V(2,4)/KP*LOG(1+EXP(KP*(1/MU+V(3,4)/SQRT(KVB+V(2,4)*V(2,4)))))}
E2   8 0 VALUE = {Ookg1mOokG2 + Aokg1*V(1,4) - alkg1palskg2/(1 + be*V(1,4))}
G1   1 4  VALUE = {0.5*(PWR(V(7),EX)+PWRS(V(7),EX))*V(8)}
G2   2 4 VALUE = {0.5*(PWR(V(7),EX)+PWRS(V(7),EX))/KG2 * (1+ als/(1+be*V(1,4)))}
RCP  1 4  1G      ; FOR CONVERGENCE A  - C
C1   3 4  {CCG1}   ; CATHODE-GRID 1 C  - G1
C4   2 4  {CCG2}   ; CATHODE-GRID 2 C  - G2
C5   2 3  {CG1G2}   ; GRID 1 -GRID 2 G1  - G2
C2   1 3  {CPG1}  ; GRID 1-PLATE G1 - A
C3   1 4  {CCP}   ; CATHODE-PLATE A  - C
R1   3 5  {RGI}   ; FOR GRID CURRENT G1 - 5
D3   5 4  DX      ; FOR GRID CURRENT 5  - C
.MODEL DX D(IS=1N RS=1 CJO=10PF TT=1N)
.ENDS PenthodeD
```

Note that the circuit is called pf86_200, which is the name of the first data file appearing in the list of datasets.

6 Fitting a penthode with secundary emission

In this section, a model will be fitted to the EL500 tube (which is identical to the PL504). Fitting a penthode with secundary emission is similar to fitting a regular penthode model; the first step is to fit a triode model as described in the previous section. The required initialization file for that is:

```
C:\Program Files (x86)\gnuplot\bin\wgnuplot.exe
1
EL500_triode
X
DerkE
700.    !Pmax in fit
-25     !Vg Offset
```

Note the fact that now the entry for VgOffset equals -25 - the measurements have been done while adding this offset to the first grid in order to keep anode currents below 200mA. For PaMax a value of 700 is used, basically saying that no limitation for anode dissipation is used. After running `ExtractModel` with this initialization file, a new initialization file `Penthode.ini` needs to be made and is shown below:

```
C:\Program Files (x86)\gnuplot\bin\wgnuplot.exe
4
```

```
EL500_300
EL500_250
EL500_200
EL500_triode
B
DerkE
700.     !Pmax in fit
-25      !Vg Offset
```

Note the use of the 'B' designator for using a penthode model with secundary emission, and the fact that the used model is 'DerkE'. Alternatively, the 'Derk' model could have been used, with a worse fit.

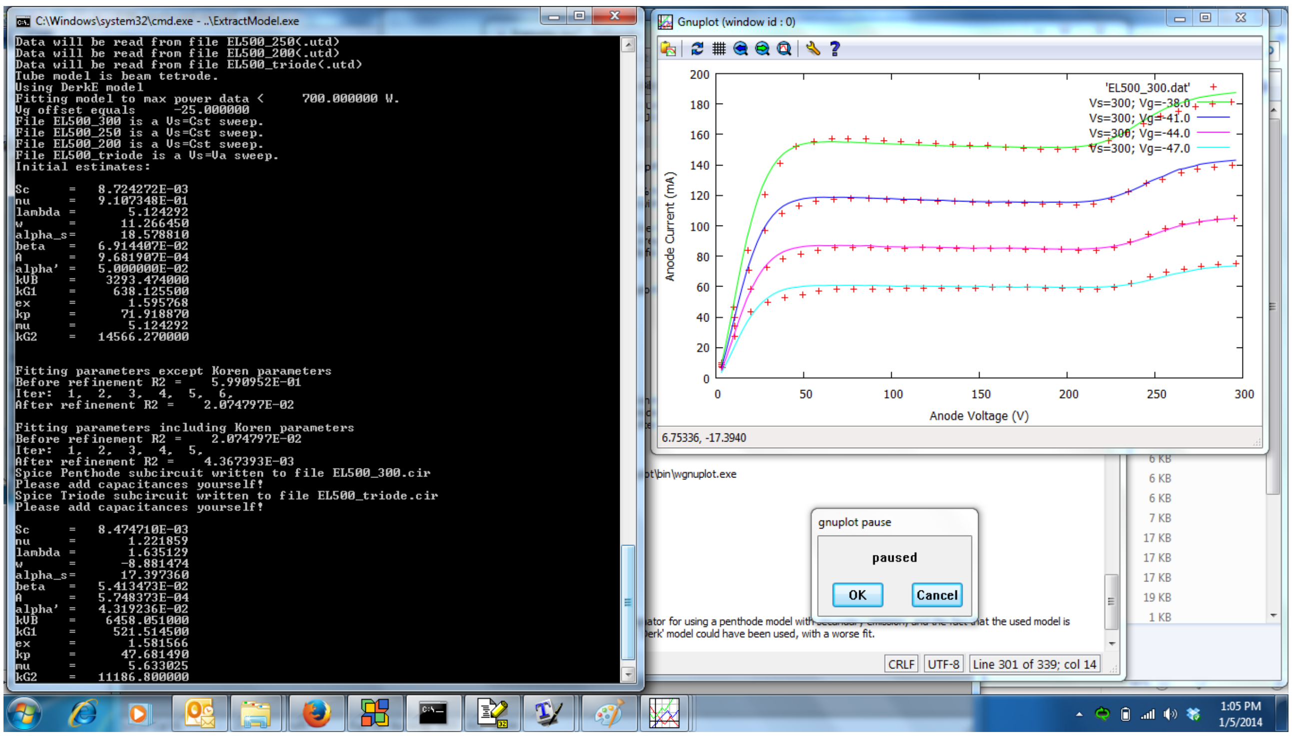

Figure 4: Output of `ExtractModel` for the EL500 example with the Penthode.ini initialization file.

The output is very similar to the output for the penthode model, the main difference being that there are more parameters that are displayed. `Gnuplot` will also show the same Pause pop-up; pressing 'OK' will have `Gnuplot` show the screen current or the next dataset. The output file types are identical as for a penthode.

7 Fitting a variable mu pentode

Fitting a variable mu pentode requires a rather different approach compared to what is outlined for a standar pentode and or a pentode with secondary emission. The sequence of fitting a triode strapped tube, followed by the pentode fit, is similar, however. We first describe the triode-strapped fit. Care should be taken in the dataset that it covers both the high and low-mu regions - *i.e.*, one should have a large range of V_{g1} voltages. A rather standard `.ini` file is shown below

```
C:\Program Files (x86)\gnuplot\bin\wgnuplot.exe
1
EF85_triode
X
Derk
10. !Pmax in fit
0 !Vg Offset
```

which leads to the result depicted below:

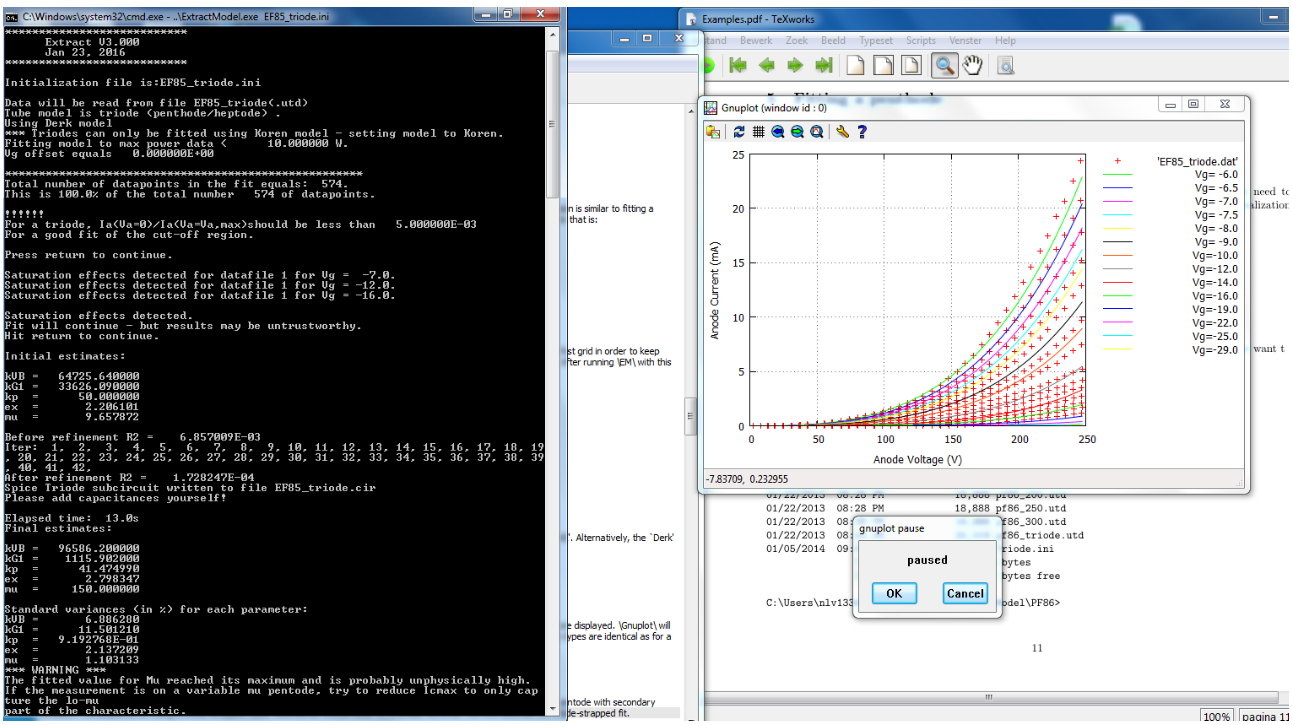

Figure 5: Output of `ExtractModel` for an EF85 triode fit producing wrong results - even though the fit superficially looks OK.

While superficially this seems a rather good fit, there are a few serious problems. The fitted μ value ran into the limit built-in in `ExtractModel`, for which reason `ExtractModel` issues a warning message:

```
 *** WARNING ***
The fitted value for Mu reached its maximum and is probably unphysically high.
If the measurement is on a variable mu pentode, try to reduce Icmax to only capture the lo-mu
part of the characteristic.
```

Also - on a closer look- there are structural misfits. The reason for these observations is that the Koren triode model cannot fit a variable-mu triode, and, hence, results in unphysical results. The remedy to this is to limit the maximum cathode current for the fit. In this case `Icmax=2` is included, leading to the following result:

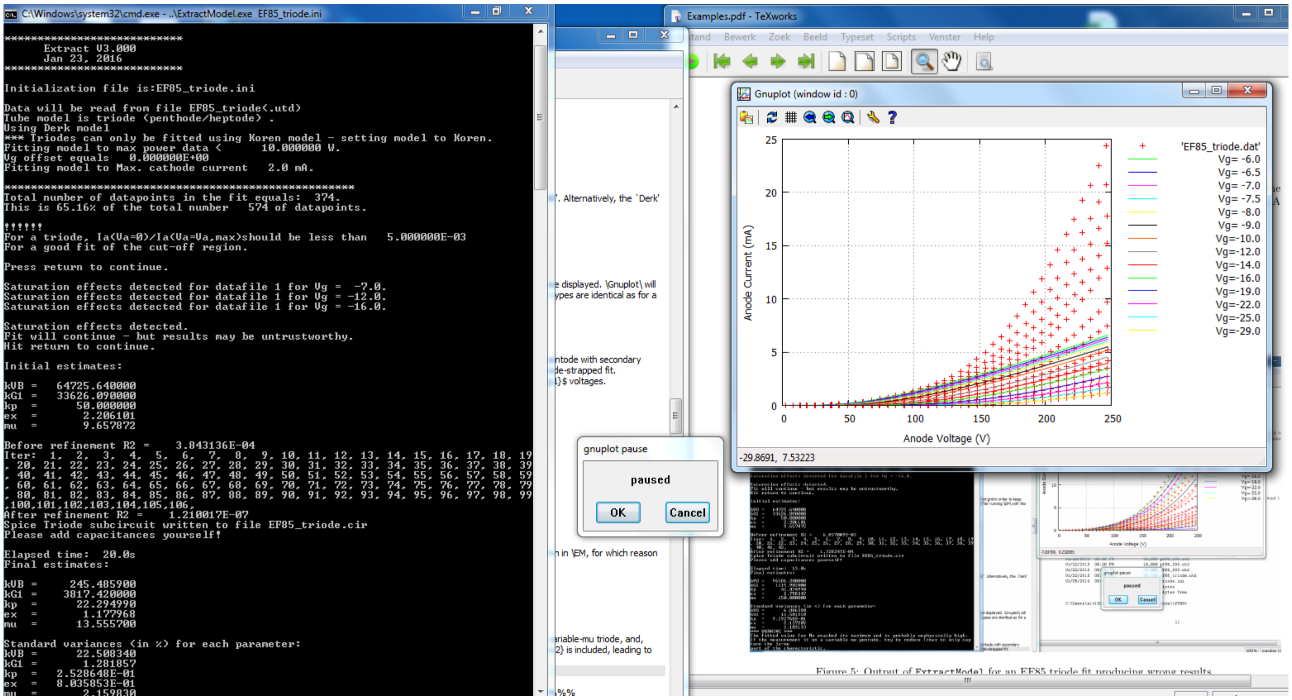

Figure 6: Output of `ExtractModel` for an EF85 triode fit producing correct results - even though the fit looks rather bad.

While on first sight this fit looks rather bad, the fit for the low anode currents is actually very good and corresponds to the low-mu part of the tube. Note, that this time the fitted mu value is realistic and `ExtractModel` no longer issues a warning.

With this triode file the process will continue similar to the fitting process for a regular pentode, with that difference that instead of 3-4 different V_{g1} values per screen voltage, now significantly more are needed to fully cover both the high and low mu part of the pentode. As a rule of thumb, at least 8 different V_{g1} values are required. A typical `.ini` file is provided below:

```
C:\Program Files (x86)\gnuplot\bin\wgnuplot.exe
5
EF85_75
EF85_150
EF85_200
EF85_250
EF85_triode
V
Derk
```

```
5. !Pmax in fit
0 !Vg Offset
```

The results of this fit are depicted below.

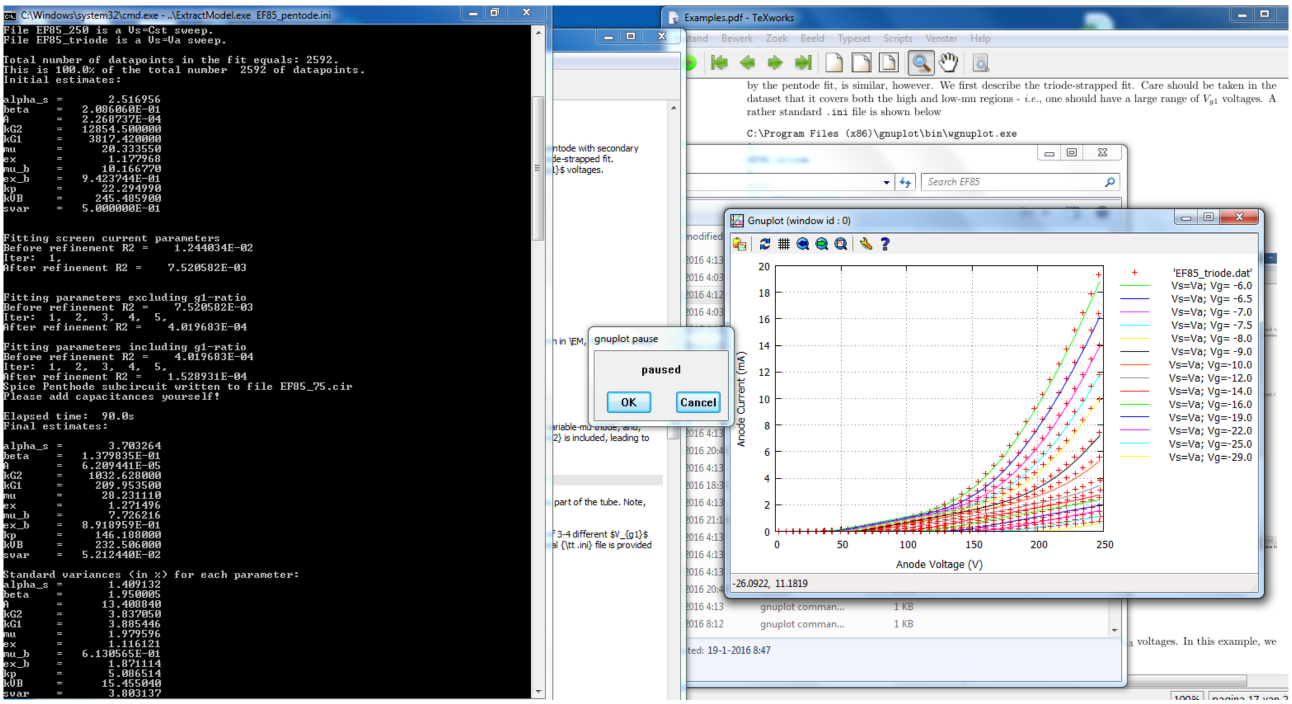

Figure 7: Output of `ExtractModel` for an EF85 pentode fit - here the triode data fit is shown.

Note that the mu-values are realistic, and that the fit of the triode data (based on the pentode fit) is now substantially better as before. It is also clear that as the variable-mu pentode model is computationally rather complex, fitting is slow - in this case 1.5 minute was needed for the fit to complete. `ExtractModel` is not designed for speed...

8 Fitting a Heptode

For a heptode, a number of datafiles need to be available for various V_{g1} and V_{g3} voltages. In this example, we will fit a heptode. Below is an example of a good set of data files required.

```
03/16/2014  01:56 PM          15,912 ECH81T.utd
03/08/2014  10:36 AM          16,666 ECH81_100_g3_10.0.utd
03/08/2014  10:38 AM          16,666 ECH81_100_g3_15.0.utd
03/08/2014  10:26 AM          16,666 ECH81_150_g3_10.0.utd
03/08/2014  10:21 AM          16,666 ECH81_150_g3_15.0.utd
```

```
03/08/2014  10:04 AM            16,666 ECH81_triode_g1_2.0.utd
03/08/2014  09:39 AM            20,807 ECH81_triode_g3_10.0.utd
03/08/2014  09:44 AM            20,807 ECH81_triode_g3_15.0.utd
              12 File(s)        211,661 bytes
               0 Dir(s)  90,205,777,920 bytes free
```

As a first step, the parameters required to estimate the space current need to be generated. This is done by fitting a triode model to the total space current (as also done in penthode model fitting). A sample input file (triodeX.ini in the example set) is listed below:

```
C:\Program Files (x86)\gnuplot\bin\wgnuplot.exe
1
ECH_triode_g3_10.0
X
Koren
2.2 !Pmax in fit
0 !Vg Offset
```

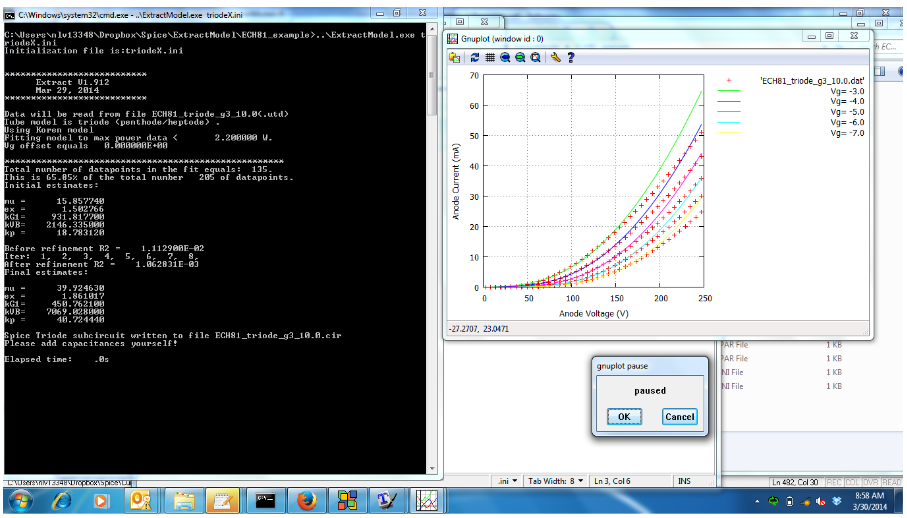

Figure 8: Output of `ExtractModel` for the ECH81 example with the triodeX.ini initialization file.

In Fig. 8, the output is shown. The fit for high currents is not perfect: the tube shows saturation for high cathode currents, and therefore the max anode dissipation has been set at 2.2W (see the `.ini` file). The output of this run is now a `Triode.par` file, that will be used lateron to generate starting values.

As a next step, an estimate needs to be generated to the parameters of the virtual penthode. This is done by fitting a triode model to the anode current as function of V_{g3}. A sample `.ini` (in the sample input this is `triodeY.ini`) file is listed below.

```
C:\Program Files (x86)\gnuplot\bin\wgnuplot.exe
1
```

```
ECH81_triode_g1_2.0
Y
Koren
2.2 !Pmax in fit
0 !Vg Offset
```

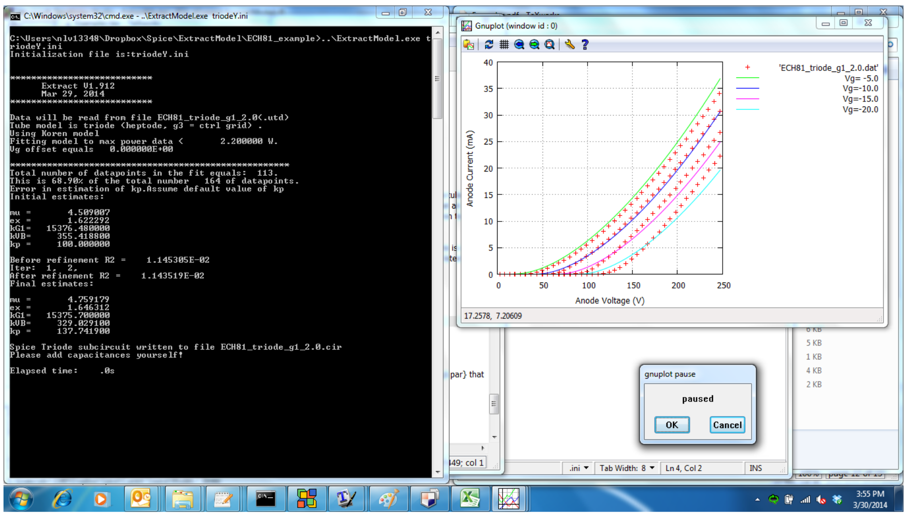

Figure 9: Output of `ExtractModel` for the ECH81 example with the triodeY.ini initialization file.

Note that as tube type, 'Y' is given. This signals `ExtractModel` to generate a parameter file `Triode_g3.par` that will be used later in the full heptode fitting. In Fig. 9, the output generated by running `ExtractModel` with the `triodeY.ini` file, is shown. The fit for high currents is not perfect; this is due to the fact that the current source for this triode (g_3 and g_4 strapped together) is modulated by g_2 - which by nature of the tube is connected to g_4. By now two `.par` files should have been generated:

```
03/22/2014  03:23 PM               110 Triode.par
03/22/2014  03:24 PM               110 Triode_g3.par
```

Once this is completed, the full heptode model can be generated. A sample `.ini` file is

```
C:\Program Files (x86)\gnuplot\bin\wgnuplot.exe
7
ECH81_100_g3_10.0
ECH81_150_g3_10.0
ECH81_100_g3_15.0
ECH81_150_g3_15.0
ECH81_triode_g1_2.0
ECH81_triode_g3_10.0
ECH81_triode_g3_15.0
H
```

```
Derk
1 !Pmax in fit
0 !Vg Offset
Icmax = 20
```

Note, that here the optional instruction `Icmax = 20` is given. This instructs `ExtractModel` to ignore data with a cathode current larger than 20 mA in the fit. A screen shot is provided in Fig. 10. As with pentodes, pressing

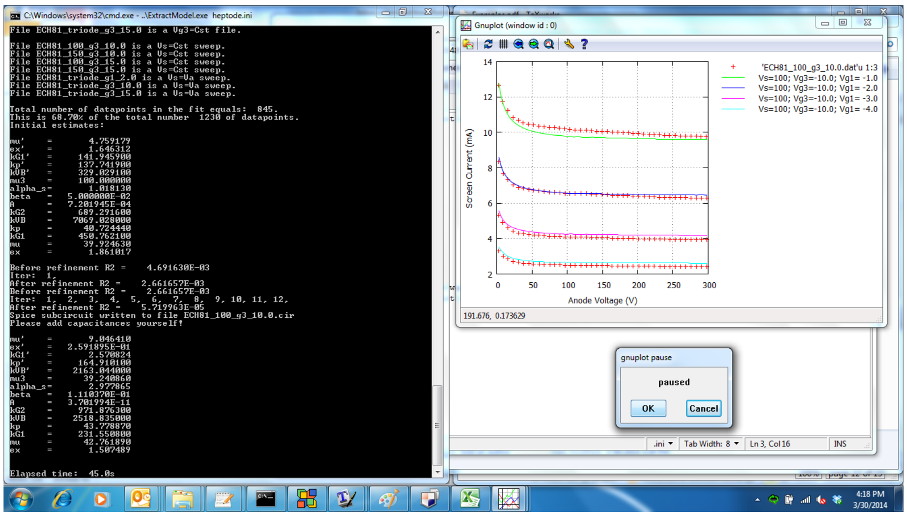

Figure 10: Output of `ExtractModel` for the ECH81 example with the heptode.ini initialization file.

'OK' will show the fit of subsequent currents I_a and I_{g2g4}. The full screen output is provided below. Experience has shown that the exact values listed, are very dependent on the machine used. It is also instructive to try to run `ExtractModel` with different values of the maximum anode power and/or cathode current. If these values are set at a too high level, `ExtractModel` gets trapped in an optimization minimum that is incorrect.

```
C:\Users\nlv13348\Dropbox\Spice\ExtractModel\ECH81_example>..\ExtractModel.exe heptode.ini
Initialization file is:heptode.ini

***************************
      Extract V1.912
      Mar 29, 2014
***************************

Data will be read from file ECH81_100_g3_10.0(.utd)
Data will be read from file ECH81_150_g3_10.0(.utd)
Data will be read from file ECH81_100_g3_15.0(.utd)
Data will be read from file ECH81_150_g3_15.0(.utd)
Data will be read from file ECH81_triode_g1_2.0(.utd)
Data will be read from file ECH81_triode_g3_10.0(.utd)
Data will be read from file ECH81_triode_g3_15.0(.utd)
```

```
Tube model is hexode/heptode.
Using Derk model
Fitting model to max power data <       1.000000 W.
Vg offset equals   0.000000E+00
Fitting model to Max. cathode current  20.0 mA.

*****************************************************
File ECH81_100_g3_10.0 is a Vg3=Cst file.
File ECH81_150_g3_10.0 is a Vg3=Cst file.
File ECH81_100_g3_15.0 is a Vg3=Cst file.
File ECH81_150_g3_15.0 is a Vg3=Cst file.
File ECH81_triode_g1_2.0 is a Vg1=Cst file.
File ECH81_triode_g3_10.0 is a Vg3=Cst file.
File ECH81_triode_g3_15.0 is a Vg3=Cst file.

File ECH81_100_g3_10.0 is a Vs=Cst sweep.
File ECH81_150_g3_10.0 is a Vs=Cst sweep.
File ECH81_100_g3_15.0 is a Vs=Cst sweep.
File ECH81_150_g3_15.0 is a Vs=Cst sweep.
File ECH81_triode_g1_2.0 is a Vs=Va sweep.
File ECH81_triode_g3_10.0 is a Vs=Va sweep.
File ECH81_triode_g3_15.0 is a Vs=Va sweep.

Total number of datapoints in the fit equals:  845.
This is 68.70% of the total number  1230 of datapoints.
Initial estimates:

mu'    =       4.759179
ex'    =       1.646312
kG1'   =     141.945900
kp'    =     137.741900
kVB'   =     329.029100
mu3    =     100.000000
alpha_s=       1.018130
beta   =   5.000000E-02
A      =   7.201945E-04
kG2    =     689.291600
kVB    =    7069.028000
kp     =      40.724440
kG1    =     450.762100
mu     =      39.924630
ex     =       1.861017

Before refinement R2 =    4.691630E-03
Iter:  1,
After refinement R2 =    2.661657E-03
```

```
Before refinement R2 =    2.661657E-03
Iter:  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12,
After refinement R2 =    5.719963E-05
Spice subcircuit written to file ECH81_100_g3_10.0.cir
Please add capacitances yourself!

mu'    =       9.046410
ex'    =   2.591895E-01
kG1'   =       2.570824
kp'    =     164.910100
kVB'   =    2163.044000
mu3    =      39.240860
alpha_s=       2.977865
beta   =   1.110370E-01
A      =   3.701994E-11
kG2    =     971.876300
kVB    =    2518.835000
kp     =      43.778870
kG1    =     231.550800
mu     =      42.761890
ex     =       1.507489

Elapsed time:  45.0s
```

Running this will result in a file `ECH81_100_g3_10.0.cir`, which is the LTSpice file for the heptode model. As in the previous examples, this name is derived from the first datafile entry in the `.ini` file.

Fitting a heptode is not always trivial, and especially when all high current (*i.e.*, high cathode current) datapoints are included, `ExtractModel` sometimes ends up in a false minimum. This is easily recognized as typically the high currents will have a decent fit (but not great), and the low current data are very badly represented. A trick to guide `ExtractModel` to the right minimum is to first execute the fit with a relatively low number of datafiles, and a rather low value of the maximum current. An example `.ini` file is below:

```
C:\Program Files (x86)\gnuplot\bin\wgnuplot.exe
7
ECH21_100_g3_0.0
ECH21_150_g3_0.0
ECH21_200_g3_0.0
ECH21_triode_g1_2.0
ECH21_triode_g1_1.0
ECH21_triode_g3_5.0
ECH21_triode_g3_10.0
H
Derk
1 !Pmax in fit
0 !Vg Offset
```

```
Icmax = 10
```

where data is fitted that display cathode currents well over 25mA. Fitting the data with this set of limitations gave a good fit, whereas fitting the data with `Icmax=30` and `Pmax=3` gave a bad fit. As always, `ExtractModel`generates a file `Model.par` which contains the values of all parameters. We will now trick `ExtractModel` by copying this file to a file `H.par` (the name is irrelevant) and forcing `ExtractModel` to use these values as start parameters by adding a command `RdStart=H.par` to the `.ini` file. We also change the maximum current and anode dissipation values now:

```
C:\Program Files (x86)\gnuplot\bin\wgnuplot.exe
7
ECH21_100_g3_0.0
ECH21_150_g3_0.0
ECH21_200_g3_0.0
ECH21_triode_g1_2.0
ECH21_triode_g1_1.0
ECH21_triode_g3_5.0
ECH21_triode_g3_10.0
H
Derk
3 !Pmax in fit
0 !Vg Offset
Icmax = 30
RdStart=H.par
```

With these new startparameters, convergence is good and led to an excellent fit. We can repeat this trick by copying the new `Model.par` file to, *e.g.*, `H2.par` and adding a new datafile to the set of files to be fitted. The new start parameters are read from file `H2.par` by changing the command for reading the start parameters to `RdStart=H2.par`.
Happy simulating in Spice!

9 Fitting a Pentode with g_3 modulation

Like with a heptode, a number of datafiles needs to be available for various V_{g1} and V_{g3} voltages. In this example, we will fit a pentode with g_3 modulation. Below is an example of a good set of data files required.

```
04/20/2014  08:56 AM            25,857 EF80_200_g1_2.0.utd
04/20/2014  08:45 AM            25,857 EF80_200_g1_3.0.utd
05/04/2014  06:54 AM            12,648 EF80_200_g3_0.0.utd
04/20/2014  08:47 AM            25,857 EF80_250_g1_3.0.utd
05/04/2014  06:54 AM            12,648 EF80_250_g3_0.0.utd
05/04/2014  06:54 AM            12,648 EF80_300_g3_0.0.utd
05/04/2014  06:54 AM            12,750 EF80_triode_g3_0.0.utd
              12 File(s)        211,661 bytes
               0 Dir(s)  90,205,777,920 bytes free
```

`ExtractModel` takes a `Penthode.par` file as input. This file can be generated by fitting the penthode with a regular Derk or DerkE penthode model. This results in a file `Model.par` (see section on penthode fitting). This file can be copied to a file with the name `Penthode.par`. Once this is completed, the full penthode model can be generated. A sample `.ini` (in this example called `pg3.ini`) file is

```
C:\Program Files (x86)\gnuplot\bin\wgnuplot.exe
7
EF80_200_g3_0.0
EF80_250_g3_0.0
EF80_300_g3_0.0
EF80_200_g1_2.0
EF80_200_g1_3.0
EF80_250_g1_3.0
EF80_triode_g3_0.0
F
DerkE
1.       !Pmax in fit
0        !Vg Offset
Icmax = 20
```

In the above, file naming is not free, as in heptode fitting. A file ending on `_g1_3.0` indicates that this is a file where g_1 has been constant at $-3V$, and g_3 has been connected to the uTracer. Note, that here the optional instruction `Icmax = 20` is given. This instructs `ExtractModel` to ignore data with a cathode current larger than 20 mA in the fit. A screen shot is provided in Fig. 11. As with regular pentodes, pressing 'OK' will show the

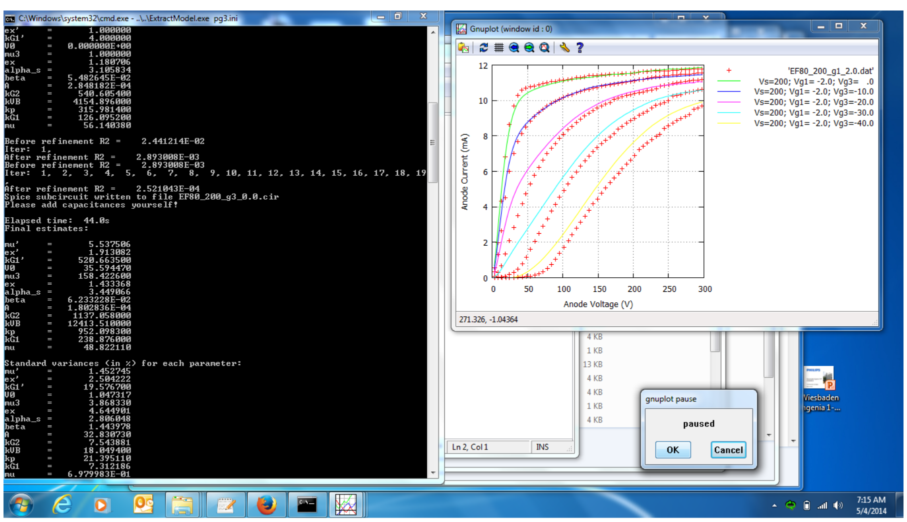

Figure 11: Output of `ExtractModel` for the EF80 with g_3 modulation example with the `pg3.ini` initialization file.

fit of subsequent currents I_a and I_{g2}. The full screen output is provided below. Experience has shown that the exact values listed, are very dependent on the machine used. It is also instructive to try to run `ExtractModel` with different values of the maximum anode power and/or cathode current. If these values are set at a too high level, `ExtractModel` gets trapped in an optimization minimum that is incorrect.

```
C:\Spice\extractmodel\EF80\EF80_g3>..\..\ExtractModel.exe pg3.ini

***************************
      Extract V1.960
      Apr 16, 2014
***************************

Initialization file is:pg3.ini

Data will be read from file EF80_200_g3_0.0(.utd)
Data will be read from file EF80_250_g3_0.0(.utd)
Data will be read from file EF80_300_g3_0.0(.utd)
Data will be read from file EF80_200_g1_2.0(.utd)
Data will be read from file EF80_200_g1_3.0(.utd)
Data will be read from file EF80_250_g1_3.0(.utd)
Data will be read from file EF80_triode_g3_0.0(.utd)
Tube model is penthode with g3 modelling.
Using DerkE model
Fitting model to max power data <       1.000000 W.
Vg offset equals   0.000000E+00
Fitting model to Max. cathode current  20.0 mA.

*****************************************************
File EF80_200_g3_0.0 is a Vg3=Cst file.
File EF80_250_g3_0.0 is a Vg3=Cst file.
File EF80_300_g3_0.0 is a Vg3=Cst file.
File EF80_200_g1_2.0 is a Vg1=Cst file.
File EF80_200_g1_3.0 is a Vg1=Cst file.
File EF80_250_g1_3.0 is a Vg1=Cst file.
File EF80_triode_g3_0.0 is a Vg3=Cst file.

File EF80_200_g3_0.0 is a Vs=Cst sweep.
File EF80_250_g3_0.0 is a Vs=Cst sweep.
File EF80_300_g3_0.0 is a Vs=Cst sweep.
File EF80_200_g1_2.0 is a Vs=Cst sweep.
File EF80_200_g1_3.0 is a Vs=Cst sweep.
File EF80_250_g1_3.0 is a Vs=Cst sweep.
File EF80_triode_g3_0.0 is a Vs=Va sweep.

Total number of datapoints in the fit equals:  635.
This is 50.48% of the total number  1258 of datapoints.
Initial estimates:
```

```
mu'      =      10.000000
ex'      =       1.000000
kG1'     =       4.000000
V0       =   0.000000E+00
mu3      =       1.000000
ex       =       1.180706
alpha_s =        3.105834
beta     =   5.482645E-02
A        =   2.848182E-04
kG2      =     540.605400
kVB      =    4154.896000
kp       =     315.981400
kG1      =     126.095200
mu       =      56.140380

Before refinement R2 =    2.441214E-02
Iter:  1,
After refinement R2 =    2.893008E-03
Before refinement R2 =    2.893008E-03
Iter:  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 1
,
After refinement R2 =    2.521043E-04
Spice subcircuit written to file EF80_200_g3_0.0.cir
Please add capacitances yourself!

Elapsed time:  44.0s
Final estimates:

mu'      =       5.537506
ex'      =       1.913082
kG1'     =     520.663500
V0       =      35.594470
mu3      =     158.422600
ex       =       1.433368
alpha_s =        3.449066
beta     =   6.233228E-02
A        =   1.802836E-04
kG2      =    1137.058000
kVB      =   12413.510000
kp       =     952.098300
kG1      =     238.876000
mu       =      48.822110

Standard variances (in %) for each parameter:
mu'      =       1.452745
ex'      =       2.504222
```

```
kG1'     =      19.576700
V0       =       1.047317
mu3      =       3.868330
ex       =       4.644901
alpha_s  =       2.806048
beta     =       1.443978
A        =      32.830730
kG2      =       7.543881
kVB      =      18.049400
kp       =      21.395110
kG1      =       7.312186
mu       =   6.979983E-01
```

This example shows also a new feature introduced in V1.96: standard variances in the fitted parameters. These standard variances provide a lower estimate of the actual errors in the parameters, see the Theory document for more details. The run of `ExtractModel` depicted above has generated the LT Spice circuit file `EF80_200_g3_0.0.cir` which can now be used. Happy simulating!

References

[Gnu13] www.gnuplot.info, 2013. Release used in preparing this document is: 09.10.2013: Release gnuplot 4.6.4.

[Kor01] Norman Koren. http://www.normankoren.com/Audio/Tubemodspice_article.html, 2001.

[Phi13] Frank's electron tube pages. http://www.tubedata.org/, 2013.

[Spa48] Karl Spangenberg. *Vacuum tubes.* McGraw-Hill, 1948.