Qucs-S/QucsStudio/Octave Schematic Synthesis Tools for Device and Circuit Parameter Extraction from Measured Characteristics

Mike Brinson
Centre for Communications Technology
London Metropolitan University
UK
Email: mbrin72043@yahoo.co.uk

***Abstract*—A universal technique for the extraction of device parameters or circuit component values from measured performance data is presented. The proposed method can be used by any circuit simulator that implements parameter sweep features and allows user defined tabulated data with independent voltage, or current, sources. A key feature of the reported extraction process is the use of schematic capture simulation icons, with their sweep parameters tabulated as a list of data points, synthesized from CSV measured data. By overlaying simulation output data on top of measured values, then varying user selected parameter/component values and re-simulating repeatable until the two data sets converge, it becomes possible to extract parameter/component values to within a specified error limit. In this paper FOSS circuit simulators Qucs-S/QucsStudio and the numerical analysis package Octave are used to demonstrate the application of the proposed schematic capture synthesis procedure in the investigate of diode inductance at high forward d.c. bias currents and a.c. signal band width.**

***Index Terms*—Qucs-S, QucsStudio, Octave. compact device modeling, parameter extraction, tuning, optimization**

I. Introduction

The extraction of device parameters and circuit component values from d.c., a.c., and transient measurements are significant steps in establishing the validity of device and circuit simulation models. One of the most important practical techniques used for this purpose relies on the comparison of measured and simulated output data where each dependent data set has a common independent axis selected from signal frequency (in the a.c., S, Z and Y domains) or time (in the transient domain) or swept parameter values (in all domains). The extraction process proceeds by overlaying simulation output data on top of measured data, varying user selected device parameters, or component values, then re-simulating the device/circuit under test until the two data sets line-up within a specified error limit. The Qucs-S/QucsStudio Free Open Source Software (FOSS) circuit simulators [1][2] allow individual or groups of parameters to be varied by ”manual slider tuning” (QucsStudio) or by computer controlled optimization employing objective target functions (Qucs-S and QucsStudio). This paper introduces a groundbreaking parameter extraction technique which links measured and simulated output data via Qucs-S/QucsStudio test bench schematics. To ensure that the independent X axis of the measured and simulated data have the same range and number of data points simulation is controlled by icons synthesized from the measured X scale data. The primary task of these icons is to set up and instantiate simulation while simultaneously ensuring that the independent axis of the measured and simulation output data are aligned automatically during parameter extraction. An overview of this process is shown diagrammatically in Fig.1. To illustrate the procedure data from a study of the admittance of a forward biased semiconductor pn junction diode is introduced, it’s model parameters extracted, analyzed and commented on. In this example the pn junction is represented by a non-linear Verilog-A module that models diode inductance generated by conductivity modulation and frequency dependent minority carrier lifetime at high forward d.c. bias currents [3][4][5].

II. Octave synthesis functions for the extraction of device and circuit parameters from measurements

A key feature of the reported parameter extraction method is the use of Qucs-S/QucsStudio simulation icons with their independent sweep parameter tabulated as a list of data points, for example in the case of a.c. simulation variable ”Sweep parameter” is set to acfrequency, variable ”type” is set to list and vector ”values” is set to a semicolon separated list of signal frequencies. The listed data points must be in ascending or descending order of magnitude. In those instances where only a few measured data points are needed, or indeed are available, they can be simply entered manually on a schematic diagram. In most cases however, this is impracticable due to the large number of data points, and it is better to convert CSV formatted data to a character separated number list using Octave [6]. Finally, after building the numerical lists, an Octave m function is used to synthesize one or more Qucs-S/QucsStudio simulation control icons. Although the icons shown in Fig.1 have different data formats, which largely depending on the simulation domain, the use of Octave to convert CSV tabulated measurements to Qucs-S/QucsStudio

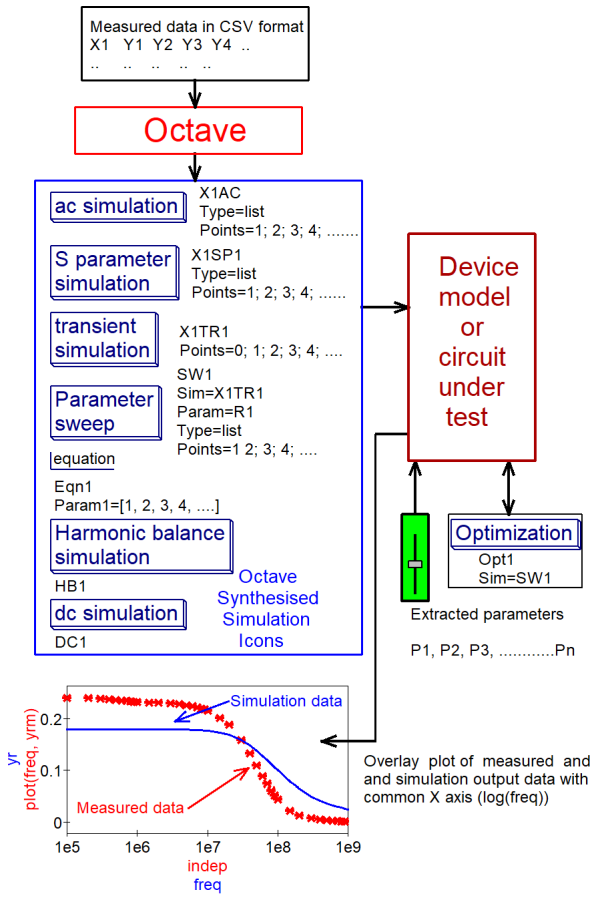

Fig. 1. A block diagram illustrating the fundamental stages for reading measured data, building extended independent and dependent variable lists and synthesizing sets of simulation control icons.

icon lists is in most cases similar. The example described in the text outlines the detail steps in the synthesis process. These can also be easily modified to change, for example, simulation type or add a higher number of dependent variables. The Octave m function given in Fig.2 lists the steps for synthesizing a set of d.c. simulation control icons applicable to extracting device parameters from measured diode *Id/Vd* characteristics. Sample sections of a typical diodeDCIV.csv data file and the initial dctemplate.sch template are also given in Fig.3.

III. The Extraction of Diode D.C. Circuit Parameters from Measured Data

Illustrated in Fig. 4 is a QucsStudio test bench schematic for extracting diode best fit d.c. parameters Is and Rs, with N = 1.0, from the Id/Vd overlay graphs plotted in Fig.4 (b). These parameters are defined in Table I and equation 5. The parameter tuning sliders drawn in Fig.4 (c) can be adjusted manually to obtain the best fit by visual comparison of the measured and simulated output data plots. Synthesized simulation icons formed from a tabulated list of measured data are represented in Fig.3 (c) by the normal Qucs-S/QucsStudio schematic symbols with attached horizontal lists of data points separated by a semicolon or comer deliminator character. The synthesized icons are generated, and stored in file DCsimTemplate.sch, by simulating Octave m function dcextractxy. Note that the simulation icon lists shown in Fig.3(c) have, for convenience, been truncated at the right hand side of the schematic diagram. Note also that the schematic illustrated in Fig.4 is different from the classical SPICE 3f5 diode model [7] in that it has an additional terminal F that senses signal frequency during simulation. In d.c. simulation the voltage on terminal F is set to 0 V d.c. to represent 0 Hz. Similarly, during other types of simulation it is set to a real voltage that represents the signal frequency in Hertz. Fig.5 in contrast to Fig.4 gives the details of an optimization simulation icon, its control settings and typical best fit output data, where in most instances, the slider extracted parameters act as initial values for a computer optimization refined fit.

```
function dcextractxy();
% An Octave function to extract measured x values (independent) and
% y (dependent) values from an n row, 2 column csv table. Extracted x
% and y lists are synthesized into a set of d.c. simulation icons.
% (C) 2020 Mike Brinson: Published under GNU General Public License
% Function dcextractxy() was developed from:
% fillmeas.m 1.0 (C) Z. Huszka 09-July-2019, and
% fillmeasurement1dep.m (C) Mike Brinson 11 October 2019.
% ==== Initialize variables =
measureddcdata="diodeDCIV.csv"; dctemplate="DCtemplate.sch";
dcsimtemplate="DCsimTemplate.sch";  csvdelim=";";
listdcx=[]; listdcy=[];
fidread=fopen(dctemplate,"r"); fidwrite=fopen(dcsimtemplate,"w");
% ===  Extract measured d.c. data ===
[dcx, dcy ]=textread(measureddcdata,"%f%f","delimiter",csvdelim);
% === Build modified x and y lists ===
for k=1:length(dcx)
  listdcx=[listdcx,num2str(dcx(k)),";"];
end
listdcx(end)=[];
for k=1:length(dcy)
  listdcy=[listdcy,num2str(dcy(k)),","];
end
listdcy(end)=[];
% === Synthesize modified simulation control icons  ===
while 1
  if findstr(line,"Eqn")
    found1=findstr(line,"["); found2=findstr(line,"]");
    line=[line(1:found1),listdcy,line(found2:end)];
    fprintf(fidwrite,"%s\n",line);
  elseif findstr(line,".SW")
    found=findstr(line,"");
    line=[line(1:found(end-1)),listdcx,line(found(end):end)];
    fprintf(fidwrite,"%s\n",line);
  else
    fprintf(fidwrite,"%s\n",line);
  end
  line=fgetl(fidread);
  if feof(fidread)
    if ischar(line)
      fprintf(fidwrite,"%s\n",line);
    end
    break
  end
end
fclose(fidread); fclose(fidwrite);
display("dcextractxy finished without error.\n");
return
```

Fig. 2. Octave function dcextractxy.m code: Works with QucsStudio netlist syntax. Modify as necessary for Qucs-S netlists.

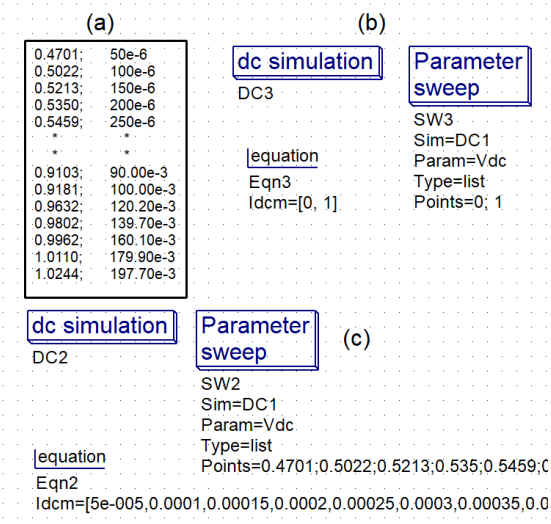

Fig. 3. Input and output data for Octave function dcextractxy(): (a) Measured input data, (b) dctemplate file "diodeDCIV.csv"; (c) dcsimtemplate file ;"DCsimTemplate;"

IV. Modeling Semiconductor PN Junction Diodes of High Forward D.C. Bias Over Wide Frequency Bandwidth

Conventional semiconductor diodes are constructed with metal contact terminals connected to short lengths of bulk or doped semiconductor material on either side of a pn junction. These add a finite amount of electrical resistance in series with the pn junction. The basic SPICE diode model represents this series resistance as a fixed parasitic resistance *Rs*. In reality however, at high d.c. currents, minority carrier charges accumulate in the series semiconductor material causing changes in the minority carrier density. Hence, as the diode d.c. current increases or decreases the resistance of the bulk/doped semiconductor varies. This process is often referred to as conductivity modulation [8] where R_0 is the intrinsic bulk semiconductor material resistance with no extrinsic charge, K_r is a real number that changes for different types semiconductor material, doping level and device geometry, I_d is the diode d.c. bias current, and the ratio K_r/R_0 is called the coefficient of conductivity modulation. One of the consequences of conductivity modulation in semiconductor diodes at high d.c. bias currents and high a.c. signal frequencies is the generation of an inductive component in the diode admittance $Y_d = Y_r + j \cdot Y_i(\omega)$ [9], where Y_r and Y_i are the real and imaginary components of the diode admittance respectively. The high frequency $Ldiode$ compact model in Fig.6 illustrates the structure and hierarchy of an experimental model that includes the basic conventional SPICE diode elements R_s, I_d and capacitance represented by nonlinear depletion and diffusion charges $Qdep$ and $Qdiff$ respectively. In the frequency domain resistor R_s models conductivity modulation and is taken to be a function of I_d. Similar to the conventional SPICE diode model it is considered to be a fixed value in the nonlinear d.c. domain. Experimental compact model $Ldiode$ models the device current phase shift at high frequencies by a simple algebraic function of the diode minority carrier lifetime (Tt) and a.c. signal frequency F. The physical properties of the experimental $Ldiode$ model are represented by the following equations

$$Id = Is \cdot \left(exp\left(\frac{V_d}{n \cdot VTH} \right) - 1.0 \right) \tag{1}$$

$$IRx = V_d / RGMIN \tag{2}$$

$$Idep = C0 \cdot \frac{Vj}{1-M} \cdot \frac{d}{dt} \left[\left(1 - \frac{Vd}{Vj} \right)^{-M} \right] \tag{3}$$

$$\forall \quad Vd <= 0V$$

$$Idep = C0 \cdot \frac{d}{dt} \left(Vd + \frac{M \cdot Vd^2}{2.0 \cdot VJ} \right), \tag{4}$$

$$\forall \quad Vd > 0V$$

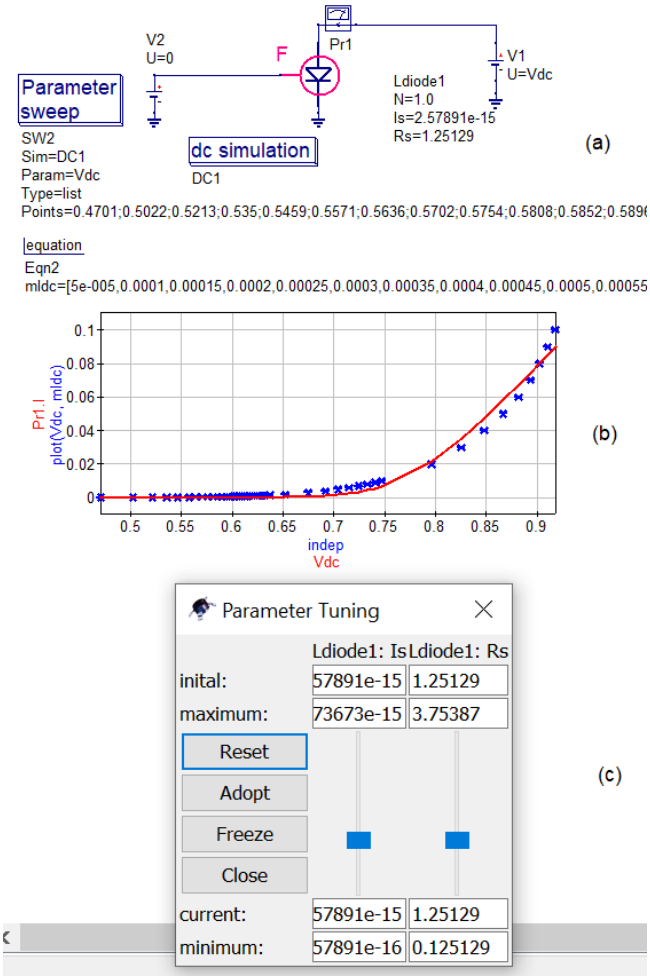

Fig. 4. Model Ldiode d.c. parameters $N = 1.0$, Is and Rs extracted using Octave m function dcextractxy() and manual "tuning sliders": (a) QucsStudio test bench schematic, (b) measured Id/Vd data (* plot) and simulation output data (solid line) overlay plots, and (c) slider settings displayed along side schematic.

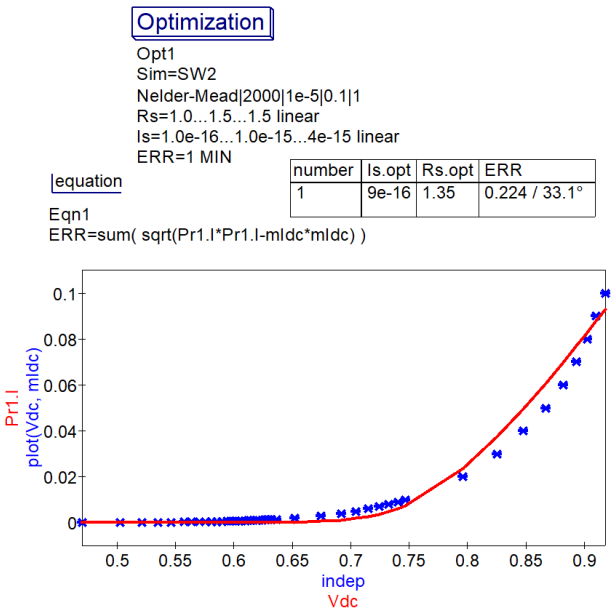

Fig. 5. Model Ldiode d.c. parameters N, Is and Rs extracted using Octave function dcextractxy(): QucsStudio optimization analysis using Nelder-Mead algorithm and objective minimized goal using target function $sum(sqrt((PrI1.I * PrI1.I) - (mIdc * mIdc)))$; measured Id/Vd and simulation output data overlay plots; best fit parameters Is = 9e-16A and Rs = 1.25$|omega$.

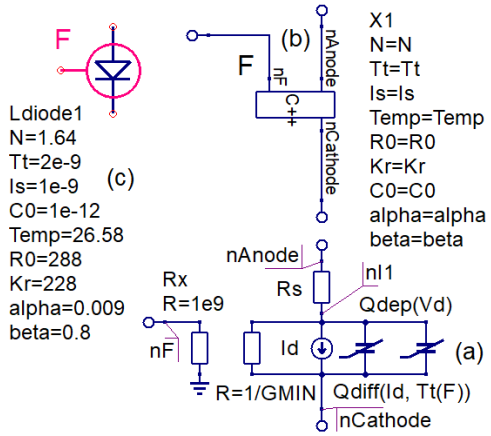

Fig. 6. Ldiode compact model: (a) electrical equivalent circuit, (b) Verilog-A synthesized C++ code block and (c) schematic symbol.

$$R_s = \frac{R_0}{(1 + K_r.I_d} \tag{5}$$

$$Idiff = \frac{Id}{(N \cdot VTH)} \cdot Tt(F) \cdot \frac{dVd}{dt} \tag{6}$$

Where $Tt(F) = Tt \cdot \left(1.0 + \alpha \cdot F^{\beta}\right)$ and F is the a.c. signal frequency in Hz and $GMIN$ is the minimum conductance supported by Qucs-S/QucsStudio (normally in the range 1e-12S to 1e-9S). Other symbols are defined in Fig.6 or Table I. Fig. 7 lists the Verilog-A module code [10][11] for the experimental $ldiode$ module introduced in Fig. 6. This module is structured around a subset of the basic SPICE diode model with extensions in the forward .d.c. bias region to account for conductivity modulation at high currents. To ensure correct a.c. performance of the device in the reverse bias region a simplified depletion capacitance model is included in the $ldiode$ module Verilog-A code.

TABLE I
LDIODE COMPACT MODEL PARAMETER VALUES

Name	Description	Unit	Default
N	Emission coefficient		1.64
Is	Saturation current at Temp.	A	1e-9
Tt	Minority carrier lifetime at Temp.	s	2e-9
$R0$	Bulk semiconductor resistance at Temp.	Ω	288
Kr	Value depends on doping and geometry	$1/A$	226
α	Conductivity modulation coeff.		0.009
β	Conductivity modulation power coeff.		0.8
M	Grading coefficient		0.5
Vj	Junction potential	V	0.7
$Temp$	Diode temperature	Celsius	27

```
`include "disciplines.vams"
`include "constants.vams"
module DiodeModel9(nAnode, nCathode, nfreq);
inout nAnode, nCathode, nfreq;
// (C) 2020 Mike Brinson: Published under GNU
//  General Public License V2 or later.
electrical nAnode, nCathode, nfreq, ni0, ni1, ni2;
parameter real N = 1.44; parameter real Tt = 2e-9;
parameter real Is = 5.6e-10; parameter real Temp=26.58;
parameter real R0 = 22.5; parameter real Kr = 180;
parameter real C0 = 1e-12; parameter real M  = 0.5;
parameter real Vj = 0.7; parameter real alpha = 0.01;
parameter real beta  = 2.0; parameter real Rs = 0.5;
branch (nAnode, ni1) B1; branch (ni1, nCathode) B2;
branch (nfreq)  B3;
real TempK, VTH, Id, GMIN;
analog begin
GMIN = 1e-12; TempK = Temp+273.15; VTH = (`P_K/`P_Q)*TempK;
Id =  Is*(limexp(V(B2)/(VTH*N))-1.0) + GMIN*V(B2);
if ( V(B3) <= 0.0)
  begin
  I(B1)  <+ V(B1)/(Rs+1e-9); I(B2)  <+ Id; I(B2)  <+ ddt(Tt*Id);
  I(B2)  <+ ddt(C0*(Vj/(1.0-M))*pow( (1.0-V(B2)/Vj), -M));
  end
else
begin
 I(B1) <+ V(B1)*(1.0/(R0/(1+Kr*Id))); I(B2) <+ Id;
 I(B2) <+ ddt(V(B2)*Id*Tt*(1.0+alpha*pow(V(B3), beta))/(N*VTH));
 I(B2) <+ ddt(C0*(V(B2)+M*V(B2)*V(B2)/(2.0*Vj)));
end
I(B3)  <+ V(B3)*1e-9;
end
endmodule
```

Fig. 7. Verilog-A module DiodeModel9 for the experimental diode model $Ldiode$: with F=0.0 Hz the model reverts to a basic SPICE level 1 diode model. Noise and reverse bias breakdown effects are not modeled

V. MEASUREMENT OF DIODE ADMITTANCE, OPERATING AT HIGH D.C. FORWARD BIAS, OVER THE FREQUENCY RANGE 50KHZ TO 1GHZ.

A fundamental test bench for measuring or simulating the admittance of a forward biased diode is given in Fig. 9. Central to the test set up is the forward biased diode with separate d.c. and a.c. signal supplies. These signals are isolated by the d.c. blocking capacitor Cb and resistor Rm. Voltage source

Vm supplies d.c. power to the test circuit to set the diode d.c. bias current in the range 4mA to 100mA. The value of Rm is set at each of the specified measurement/simulation d.c. current level to be at least twenty times the diode d.c. forward resistance, yielding a high degree of isolation between the a.c. and d.c. signals. This simple form of bias tee network has been chosen to allow measurements of a satisfactory accuracy over the measurement/frequency range 50kHz to 1GHz. Values for the real and imaginary parts of diode admittance were obtained by measuring a wide band diode S11 parameters, at each of the different d.c. forward bias states with a vector network analyzer [14], followed by conversion of the measured data to CS tables composing n rows by 3 columns (F, Y_r, Y_i). The test bench in Fig. 9 gives details of the corresponding simulation test bench and the conversion of S11 values to admittance, see equation eqn2.

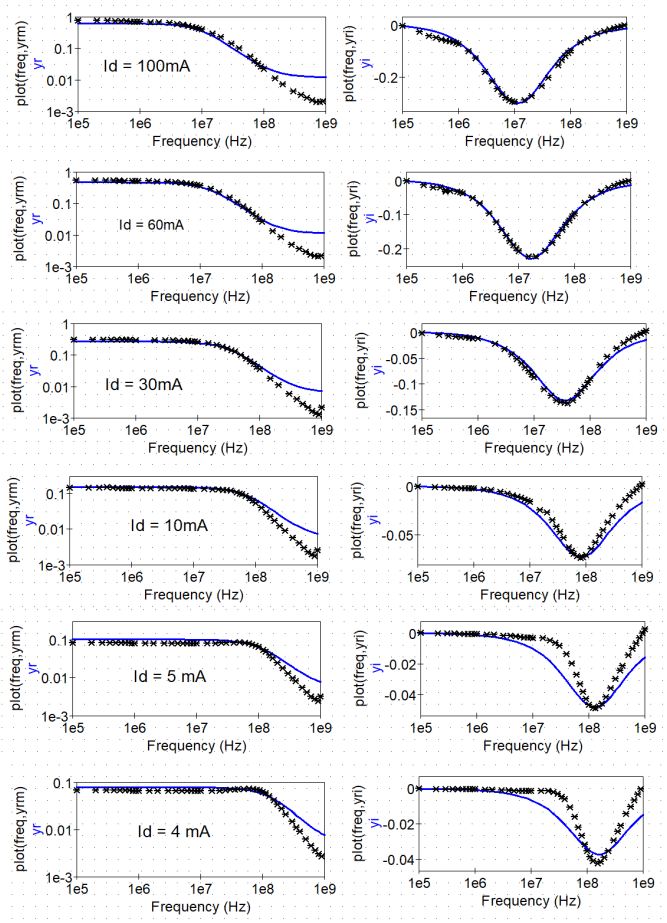

Fig. 8. Forward biased diode admittance plots for 4mA d.c. <= Id <= 100mA d.c.: plots in the left column are for Y_r (Ω) versus frequency (Hz) and plots in the right column are for Y_i (Ω) versus frequency (Hz): d.c. forward bias currents are shown on each row of plots. Solid lines represent simulation output and "*" measured data.

VI. Discussion

Extraction of the diode parameters from measured F, Y_r and Y_i data requires one test bench per d.c. forward bias current. Plots of a typical set of measured/simulation data are illustrated in Fig.8. The simulated values for Y_r and Y_i are shown fitted to the measured data. These were obtained with $R0 = 1.2e4$ and QucsStudio "tuning sliders" varying parameters Kr and α. The other parameters were either extracted from d.c. measurements (N=1.0, Is= 9.0e-16 A) or assumed to be typical physical values for a broad band diode (Tt=6.9e-12s, $C0$=1.0e-12F, M=0.5, Vj=0.7). In the case of parameter β a value of 0.87, approaching a linear function of F, was found to provide a reasonable fit to the measured data, particularly at frequencies in the mega Hertz region. At d.c. forward bias currents above 3 mA the imaginary part of the diode admittance Y_i shows inductive properties. Although, the sections of the $ldiode$ model that represent conductivity modulation and frequency dependent carrier lifetime are very simple the fit between measured and simulated data is good up to frequencies in the 100

TABLE II
LDIODE EXTRACTED R, Kr AND α PARAMETER VALUES FOR DIFFERENT D.C. BIAS CURRENTS

name	100mA	60mA	30mA	10mA	5mA	4mA
kr(1/A)	927.2	1374.62	2130.6	5341.6	9828.5	10674
α(1/Hz)	6.8e-5	5.5e-5	3.9e-5	2.9e-5	2.79e-5	2.5e-5

TABLE III
LDIODE EXTRACTED F_r AND Y_i VALUES AT DIFFERENT D.C. BIAS CURRENTS

name	100mA	60mA	30mA	10mA	5mA	4mA
F_r(Hz)	1e7	1.5e7	3.5e7	8e7	12.9e7	17.4e7
$Y_i(\omega)$	-0.302	-0.229	-0.138	-0.076	-0.049	-0.041

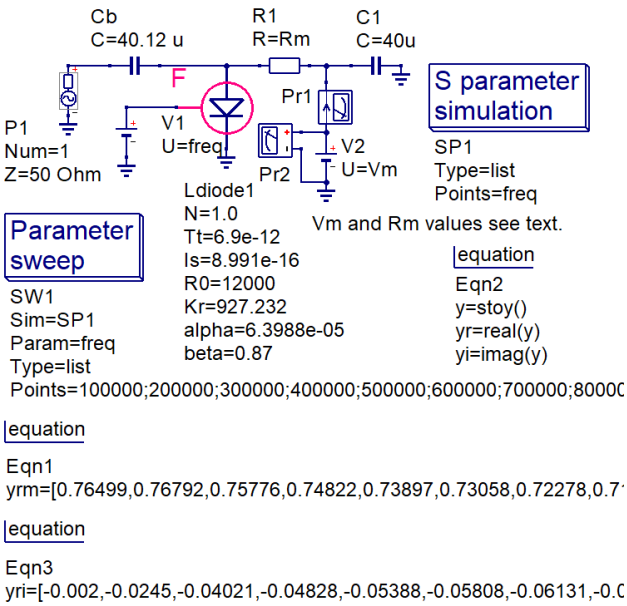

Fig. 9. A test bench for measuring or simulating the admittance of a forward biased diode in the range 4mA to 100mA over the frequency range 50kHz to 1GHz.

MHz region. Above 100 Mhz significant deviations occur, particularly beyond the clearly visible resonance in the Y_i plots. Values for the extracted parameters Kr and α are listed in Table II. Similarly, Table III gives values for the observed resonance frequencies (F_r) and the associated Y_i data. Further improvement in the data fit, if required, are potentially possible with the optimization extension to parameter extraction based on the initial parameter values extracted with "slider tuning", see Fig. 6.

VII. Conclusion

The ability to extract device or circuit parameter values from measured data is an important attribute of both circuit simulation tools and compact models. This paper introduces a new technique for parameter extraction from measured data using synthesized schematic capture icons that control circuit simulation and play a central role in the parameter extraction process. Provided a circuit simulator implements swept parameter/component value facilities and allows user defined data for voltage, or current, sources the proposed technique can be applied universally across simulators. When combined with Octave the development of more versatile compact modeling tools becomes a definite possibility, opening up a number of significant routes for future research. To demonstrate the proposed parameter extraction technique an example is introduced in the text that illustrates its use to investigate the inductive properties of d.c. forward biased semiconductor diodes.

References

[1] V. Kusnetsov and M. Brinson, "Qucs-S: Qucs with SPICE". Version 0.0.22, https://ra3xdh.github.io/, 2020. [Accessed January 2020].

[2] M. Margraf, "QucsStudio Version 2.5.7, http://dd6um.darc.de/ QucsStudio/qucsstudio.html, 2018, [Accessed January 2020].

[3] W. Shockley, "The Theory of p-n Junctions in Semiconductors and p-n Junction Transistors", 1949, Bell System Technical Journal, pp. 435-489.

[4] A.A Barna and D. Horelick, "A Simple Diode Model Including Conductivity Modulation", IEEE Transactions on Circuit Theory, Vol. CT-18, No. 2, 1971, pp. 233-240.

[5] L. Ladany, " An analysis of inertial inductance in a junction diode", IRE Trans. Electron Devices, Vol. ED-7, 1960, pp. 303-310.

[6] John W. Eaton, David Bateman, Søren Hauberg and Rik Wehbring. GNU Octave version 5.1.0 manual: a high-level interactive language for numerical computations. 2019, CreateSpace Independent Publishing Platform. ISBN 1441413006, URL http://www.gnu.org/software/octave/doc/interpreter/, [Accessed January 2020].

[7] B. Johnson, T. Quarles, A.R. Newton, D. O. Pederson and A Sangiovanni-Vincentelli, "SPICE3 Version 3f User's Manual", Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA, 1992.

[8] W.H Ko, "The forward transient behaviour of semiconductor junction diodes", Solid-State Electron., Vol. 3, 1961, pp. 59-69.

[9] Y. Kanai, "On the inductive part in the ac characteristics of the semiconductor diodes", J. ///phys. Soc. Japan, vol. 10, 1955, pp. 718-720.

[10] Accellera, "Verilog-AMS Language Reference Manual, version 2.2", available from: http://www.accellera.org, [Accessed February 2019].

[11] L. Lemaitre, C. McAndrew and S. Hamm, "ADMS- Automatic device model synthesizer", IEEE Custom Integrated Circuits Conference, 2002, pp. 27-30.

[12] DG8SAQ, "VNWA 3E Vector Network Analyzer", SDR-Kits Ltd, 2013, Trowbridge Wilts., UK. Available from https://www.sdr-kits.net.


```
function acyextract();
% (C) 2020 Mike Brinson: Published under GNU GPl V2.0 or later.
% Developed from: fillmeas.m 1.0 (C) Z. Huszka 09-July-2019, and
% fillmeasurement1dep.m (C) Mike Brinson 11 October 2019.
% == Initialize variables ==
measuredacdata="ywb4m.csv";
actemplateyr="actemplateyr.sch"; actemplateyi="actemplateyi.sch";
acsimtemplateyr="acsimtemplateyr.sch";
acsimtemplateyi="acsimtemplateyi.sch";
csvdelim=";"; listacx=[]; listacyr=[]; listacyi=[];
% == Extract measured d.c. data ==
[acx, acyr, acyi ]=textread(measuredacdata,"%f%f%f",
  "delimiter",csvdelim);
% == Build modified x and y lists ==
for k=1:length(acx) listacx=[listacx,num2str(acx(k)),";"]; end
listacx(end)=[];
for k=1:length(acyr) listacyr=[listacyr,num2str(acyr(k)),","]; end
listacyr(end)=[];
for k=1:length(acyi) listacyi=[listacyi,num2str(acyi(k)),","]; end
listacyi(end)=[];
% == Synthesize AC Yr Eqn and frequency simulation icons
fidreadyr=fopen(actemplateyr,"r");
fidwrite=fopen(acsimtemplateyr,"w");
line=fgetl(fidreadyr);
while 1
  if findstr(line,"Eqn") found1=findstr(line,"[");
   found2=findstr(line,"]");
   line=[line(1:found1),listacyr,line(found2:end)];
   fprintf(fidwrite,"%s\n",line);
  elseif findstr(line,".SW")
  found=findstr(line,"");
  line=[line(1:found(end-1)),listacx,line(found(end):end)];
  fprintf(fidwrite,"%s\n",line);
  else
  fprintf(fidwrite,"%s\n",line);
  end
  line=fgetl(fidreadyr);
  if feof(fidreadyr)
    if ischar(line) fprintf(fidwrite,"%s\n",line);
  end
  break
 end
end
fclose(fidreadyr); fclose(fidwrite);
display("acextractyr finished.\n");
% == Synthesize AC Yi Eqn simulation control icon  ==
fidreadyi=fopen(actemplateyi,"r");
fidwrite=fopen(acsimtemplateyi,"w");
line=fgetl(fidreadyi);
while 1
  if findstr(line,"Eqn") found1=findstr(line,"[");
   found2=findstr(line,"]");
   line=[line(1:found1),listacyi,line(found2:end)];
   fprintf(fidwrite,"%s\n",line);
  else
  fprintf(fidwrite,"%s\n",line);
  end
  line=fgetl(fidreadyi);
  if feof(fidreadyi)
    if ischar(line) fprintf(fidwrite,"%s\n",line);
    end
    break
  end
end
fclose(fidreadyi); fclose(fidwrite);
display("acextractyi finished.\n");
return
```

Fig. 10. An Octave function to extract measured frequency values (independent) and Yr and Yi (dependent) values from an n row, 3 column CSV table. Extracted x and y lists are synthesized into a set of a.c. simulation icons.