JOINUS

JOsephson INterface Utility Software

Manual in a nutshell

Simulation tool for superconducting circuits based on Josephson junctions

Version 2.0 - August 2019

SASAN RAZMKHAH & PASCAL FEBVRE
UNIVERSITE SAVOIE MONT BLANC
FRANCE

TABLE OF CONTENTS

LICENSE 2

FOREWORD 3

I. INSTALLATION 3

I.1. Compiling JOINUS 3

I.2. Required resources 4

I.3. JOINUS directory 4

I.4. Running JOINUS 5

II. JOINUS ENVIRONMENT 5

II.1. JOINUS interface 5

II.2. JOINUS menus 6

II.3. JOINUS built-in plotter 7

II.4. JOINUS *Optimizer and margin analyzer* tool 8

III. JOINUS TYPES OF SIMULATIONS 10

III.1. Time domain simulations 10

III.2. Parametric analysis 11

III.3. Temperature analysis 12

III.4. Bit Error Rate (BER) 13

III.5. Frequency response 14

III.6. I-V characteristics 15

REFERENCES 17

APPENDIX I – INFLUENCE OF THE TEMPERATURE ON SIMULATIONS 18

APPENDIX II – SQUIDMAP (SQUID MAGNETIC PATTERN SIMULATOR) 20

The software is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the U.S. Army Research Office grant W911NF-17-1-0120. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation herein.

Contact Pascal.Febvre@univ-smb.fr for comments, inquiries or needs of new functionalities of JOINUS

FOREWORD

Josephson interface utility software (JOINUS) is a program developed at the IMEP-LAHC laboratory, CNRS UMR5130 at Université Savoie Mont Blanc, France as part of COLDFLUX/SuperTools project [1] . The first objective is to simplify the design and simulation process of analog, digital and mixed signal circuits based on Josephson junctions. The second objective is to improve the accuracy of models for simulations and the features of software tools. JOINUS uses JSIM [2] or JoSIM [3] simulation engines for digital circuits. It will include later more simulators such as WRSPICE, an improvement of SPICE3 [4], or PSCAN [5]. JOINUS embeds a powerful built-in plotter to automatically plot output results. Other open-source plotters like GNUPlot or XMGrace are also available from the JOINUS interface.

I. INSTALLATION

I.1. Compiling JOINUS

JOINUS is a program written in the Qt C++ integrated development environment (IDE) [6] in order to be multi-platform. The source code can be run in Qt IDE on any platform. Follow the following steps to compile the code.

1) Open “*SIMGUI.pro”* file in Qt IDE and configure the *Release* and *Debug* properties from the project manager. For Unix-based systems, gcc14 can be used. For Windows systems it is recommended to choose “MinGW 32 or 64 bit” as debugger (see Figure 1). Once the code is built, the executable files are located inside the release or debug folders. Unix-based systems generate a stand-alone code.

Figure 1 Debugger initialization for Qt.

2) To make a standalone program on Windows, either you should follow the steps described in “https://wiki.qt.io/Build_Standalone_Qt_Application_for_Windows”, or you can directly use the executable “JOINUS.exe” from the release or debug folder and run the command :
3) “windeployqt.exe --quick” in the command console or in PowerShell for Windows 10. This command will add the required dynamic library (*.DLL) files to the path of JOINUS.
4) To open the help files, Adobe Acrobat Reader ActiveX is needed on Windows, it can be obtained from the Adobe website for free.
5) For more experienced users, use “*qmake.exe*” to compile the program from the terminal on Windows with the command:

qmake.exe SIMGUI.pro -spec win32-g++ "CONFIG+=qtquickcompiler"

6) On other operating systems run the "*qmake*" command in the terminal on the "*SIMGUI.pro*" file. For more information see:

   https://doc.qt.io/QtQuickCompiler/

The available plotters of JOINUS are a built-in custom plotter that uses Qcustomplot library [7], and the open-source XMGrace and GNUplot programs which can be called from JOINUS if they are installed on the system.

I.2. Required resources

JOINUS does not have an integrated simulator engine. It uses the JSIM and JoSIM open-source engines for simulations. Therefore, at least one of them is needed for correct operation of JOINUS.

JSIM and JoSIM are available here:

https://github.com/JoeyDelp?tab=repositories

JSIM is also available here:

http://www0.sun.ac.za/ix/?q=tools_jsim

Please check the license and terms of use on their respective repositories.

The Windows compiler uses Adobe Acrobat ActiveX to view PDF files.

A stand-alone program, SQUIDMAP, can be run from the Tools menu of JOINUS. This program, still under development, is also developed at Université Savoie Mont Blanc, in the same framework as for JOINUS.

I.3. JOINUS directory

For version v2.0 of JOINUS, the organization and the filenames should stay as shown in Figure 2 for Windows. Joinus.exe is the main file to run JOINUS.

There should be two other folders (Data & Help) and an optional SQUIDMAP folder for the SQUIDMAP tool.

JSIM and JoSIM, named respectively jsim_n.exe and JoSIM_n.exe for Windows OS, must be located inside the Data folder. The other files in the Data folder are provided examples for tests. The Help folder contains two PDF manuals for the program. One is the manual you are reading and the other one explains in detail how to do simulations with JSIM. They are both accessible through the help menu of JOINUS.

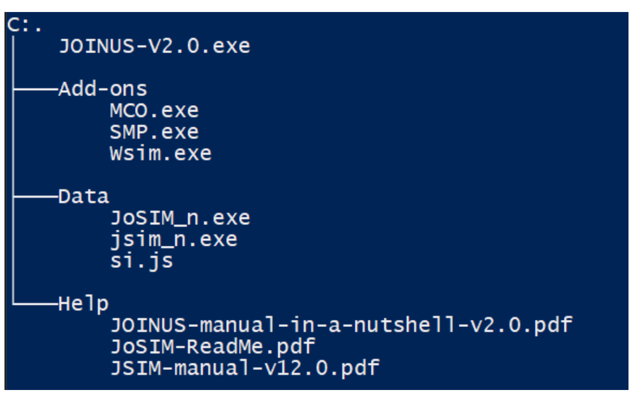

Figure 2 Files and folders of the JOINUS directory after installation.

I.4. Running JOINUS

In order to run the program just click on the executable file. The interface should look like the one of Figure 3.

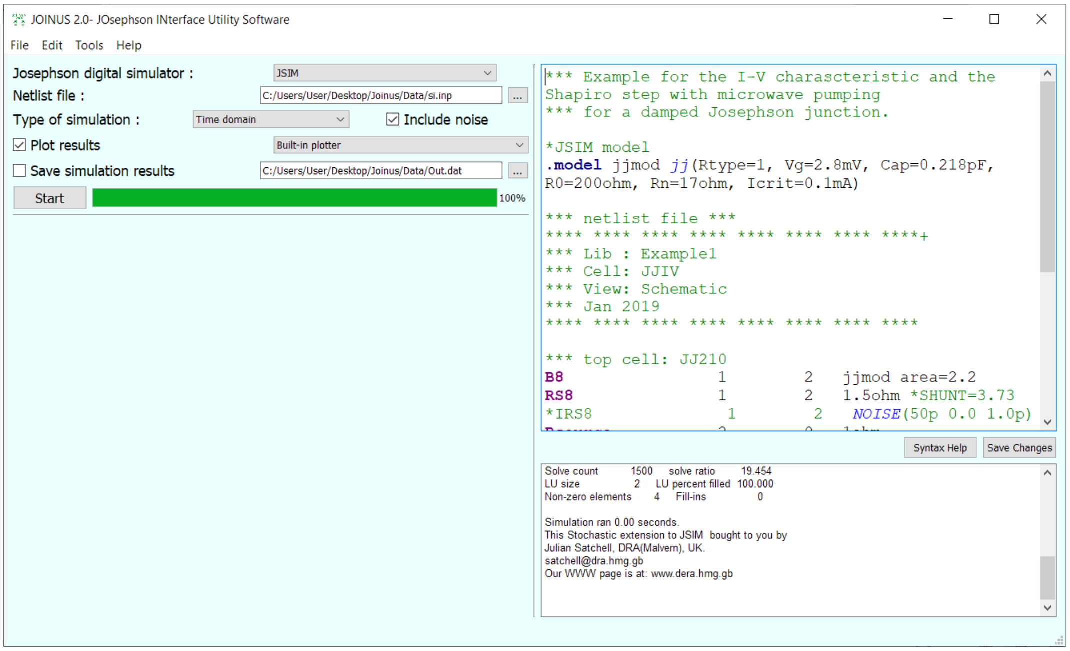

Figure 3 Interface of JOINUS software, version 2.0.

II. JOINUS ENVIRONMENT

II.1. JOINUS interface

The different fields of the interface are explained below.

1) *Josephson digital simulator* : choose between JSIM or JoSIM simulator engine.
2) *Netlist file* : choose and load the netlist file. The netlist file syntax is discussed in the JSIM manual.
3) *Type of simulation* : choose the type of simulation to perform. All choices are explained in the section *JOINUS types of simulation* below.
4) *Include Noise* : tick checkbox to include thermal noise in simulations. Thermal noise current or voltage is calculated automatically from the values of resistors present in the netlist and the chip temperature (whose value can be fixed from the *Edit* menu). Noise generators are added automatically to the circuit netlist.
5) *Plot results* : tick checkbox to generate plot from output results. The plotter can be chosen from the combo box. The powerful built-in plotter is chosen by default. It is also possible to use GNUPlot or XMGrace plotting tools.
6) *Save simulation results :* tick checkbox to save the output file in the chosen directory. The destination folder and the name of the output file can be entered in the field located to the right of the checkbox. The file can be saved in ASCII or CSV format.
7) *Start button* : Starts the simulation process or the chosen routine. A progress bar shows the status of the simulations.

8) The netlist file is shown in the top right text window and can be edited directly. To apply changes made to the netlist file click on the *Save changes* button. If the modified netlist is not saved the initial netlist is used for simulations.
9) *Syntax help* : Shows the JSIM help file for syntax help. This file is also available from the *Help* menu.
10) The bottom right window is a built-in terminal. It displays in particular outputs and errors from simulations.

II.2. JOINUS menus

The JOINUS menus are shown in the next four figures.

1) The *File* menu allows to create a new project, to load and save the workspace and to quit JOINUS.
2) The *Edit* menu gives the user the possibility to choose font properties and change the background color. Other features in grey are not yet available.

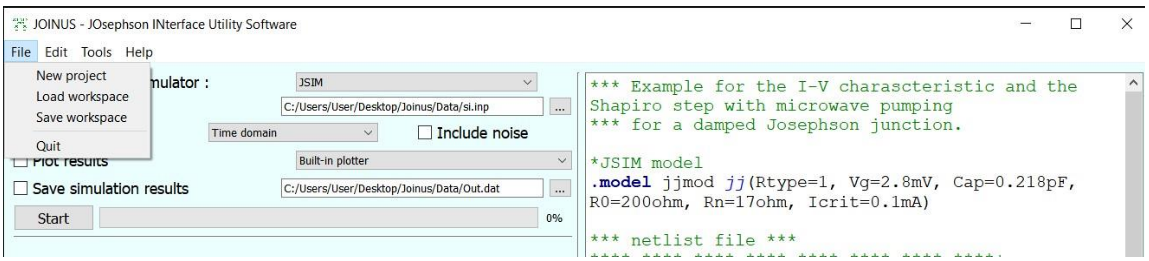

Figure 4 File menu of JOINUS software, version 2.0.

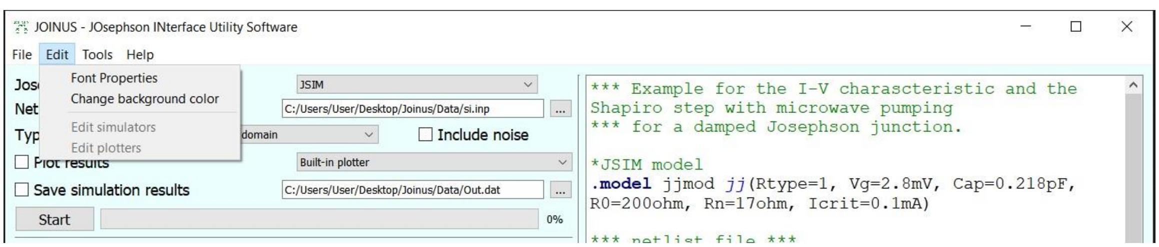

Figure 5 Edit menu of JOINUS software, version 2.0.

3) The *Tools* menu gives the user access to several choices :

a) the user can modify the default chip (global) ambient temperature. Temperature is used in several models used by JOINUS. Critical currents of Josephson junctions, gap voltage or London penetration depth are parameters that depend on temperature for instance.

b) The user can also (un-)select the adaptive step mode for simulations. The adaptive step mode is an algorithm that allows to extract, for instance, more accurate *I-V* curves by averaging the voltage over a duration chosen in a wise way. The drawback is that this mode is slower.

c) The built-in custom plotter can also be accessed from this menu.

d) The *optimizer and margin analyzer* is run from this menu. It allows to run iteratively the simulator engine (like JSIM or JoSIM) on a circuit netlist, either to find the margins of its parameters, or to optimize the nominal parameters with respect to design criteria. The different available options can be chosen easily once the *Optimizer and margin analyzer* window is opened, thanks to a help explanation to set parameters.

e) The user can modify advanced parameters such as the critical temperature of the superconductors and other constants.

f) The SQUIDMAP standalone application can be run from the *Tools* menu (see Appendix II).

4) The *Help* menu gives users access to all manuals.

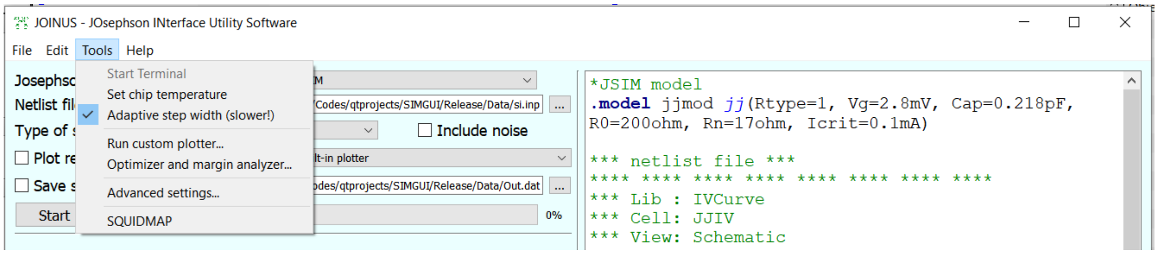

Figure 6 Tools menu of JOINUS software, version 2.0.

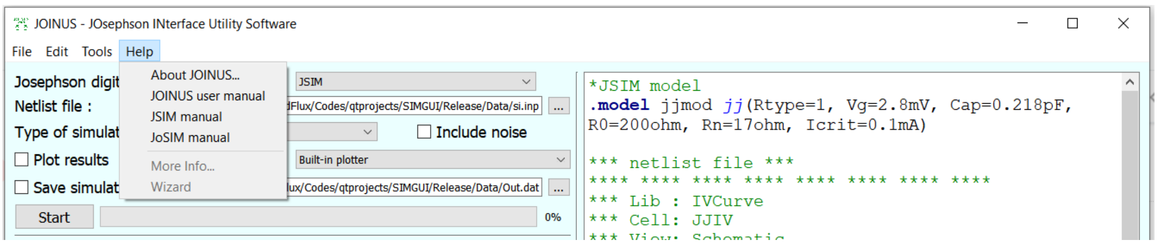

Figure 7 Help menu of JOINUS software, version 2.0.

II.3. JOINUS built-in plotter

An example of curves drawn on the custom built-in plotter of JOINUS is shown in Figure 8.

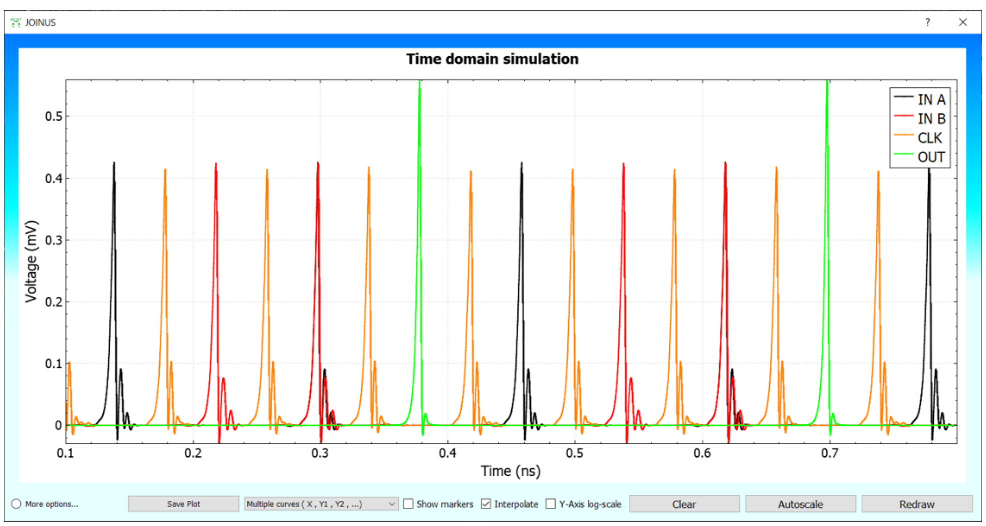

Figure 8 Window of the built-in plotter of JOINUS.

- The scale of each axis can be modified by scrolling the mouse wheel with the pointer of the axis. The full graph is scaled in both dimensions when the pointer is on the graph.
- The *Autoscale* button brings back the full scale that fits all curves in the window.
- The *Clear* button clears all curves. Individual curves can be cleared by right-clicking on them.

- The *Redraw* button is self-explanatory.
- Title, legends and axis labels can be changed by double-clicking on them. The location on the legend can be set by right-clicking on it.
- The scale can be set to logarithmic or linear mode: click checkbox for logarithmic scale.
- Data can be interpolated in a linear way : click checkbox to interpolate data.
- Markers can be shown on each data point : click checkbox.
- Different graphing modes can be selected:
  - Single curve (X , Y): plots the first two columns from the data file, the first column is on the X axis.
  - Single curve (Y , X): plots the first two columns from the data file with the second column as X axis.
  - Multiple curves (X , Y1 , Y2 , ...) : plots all columns from the data file, the first column is used for the X axis while all other columns are on the Y axis.
  - Multiple curves (X1 , Y1 , X2 , Y2 , ...) : plots all columns from the data file, columns of odd numbers are X and the rest are Y.
  - Bar Plot : is used for show margins from the *Optimizer and margin analyser* (see below).
- *More options...* menu :
  - export graphs in different image formats,
  - choose the colors of plots,
  - select data files already saved on computer to plot to compare with new simulation data.

II.4. JOINUS *Optimizer and margin analyzer* tool

The interface for margin calculations is shown in Figure 9. It can be reached from the *Tools* menu. Most of the options are self-explanatory. You can see more help by clicking “?” on the top right part of the window and move to the field. The calculations of margins are done by choosing the *Margin analyzer* radio button on the bottom right of the window. The display of margins is shown in Figure 10.

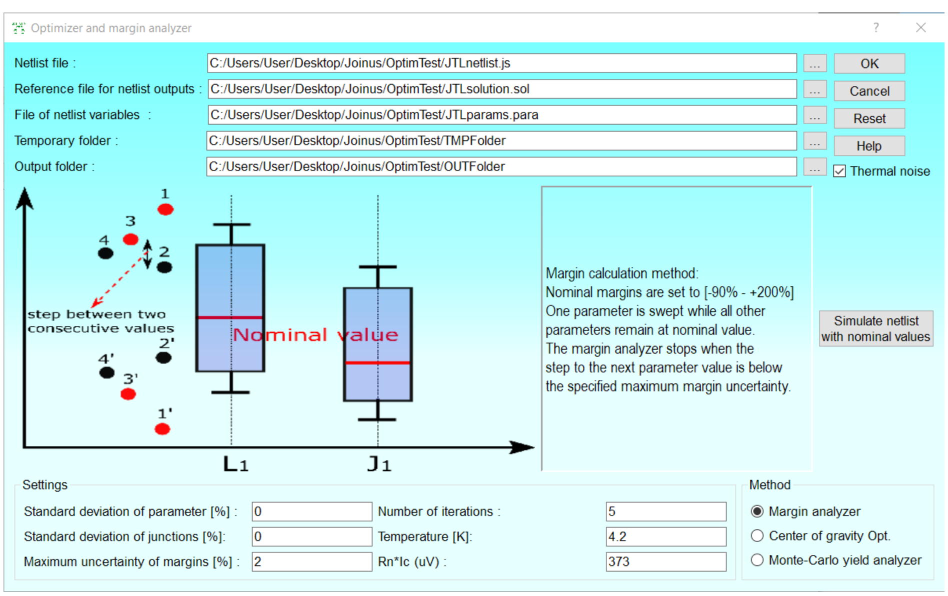

Figure 9 Window for the margin analyzer of the "Optimizer and margin analyzer" tool.

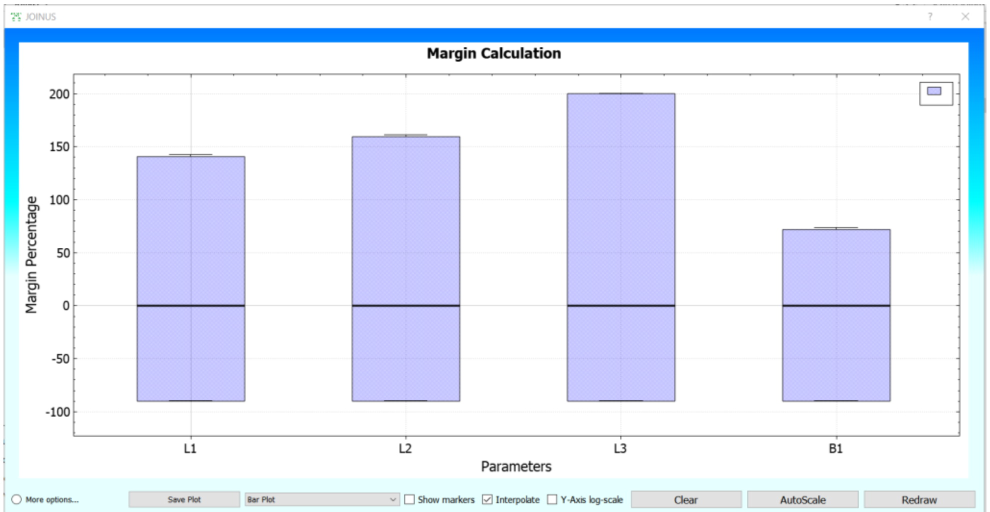

Figure 10 Margin calculations displayed in the JOINUS built-in plotter (Bar plot display).

The JOINUS *Optimizer and margin analyzer* tool provides a built-in optimizer based on the center of gravity method that can be run by choosing the *Center of analyzer* radio button on the bottom right of the window (see Figure 11).

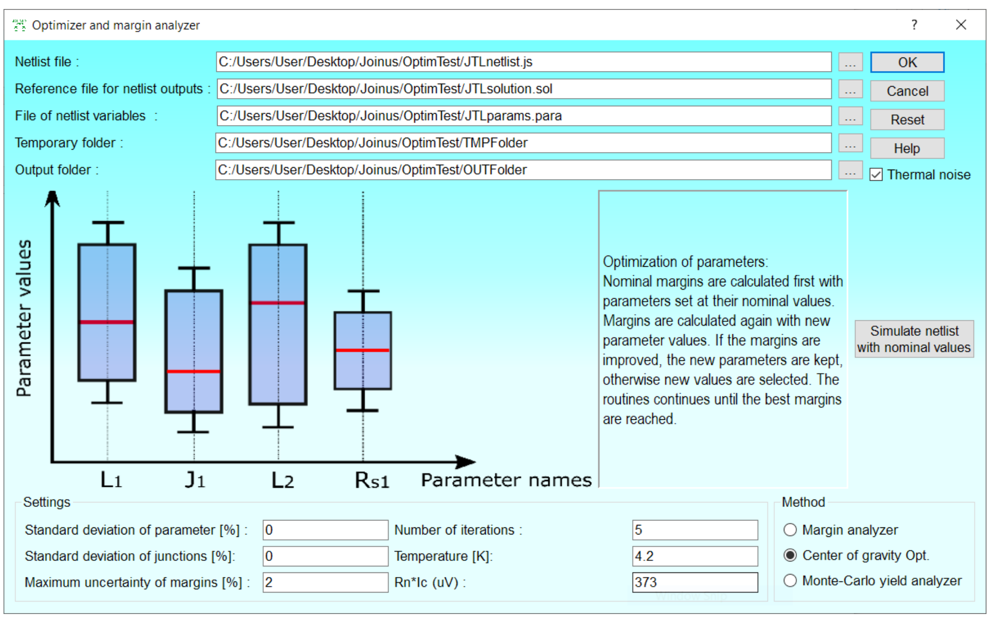

Figure 11 Window for the margin optimizer based on the center of gravity method. This window is part of the "Optimizer and margin analyzer" tool.

A last the *Optimizer and margin analyzer tool* provides a built-in Monte-Carlo yield analyzer that can be run by choosing the corresponding radio button on the bottom right of the window (see Figure 12). The window explains the method used to simulate the yields curves.

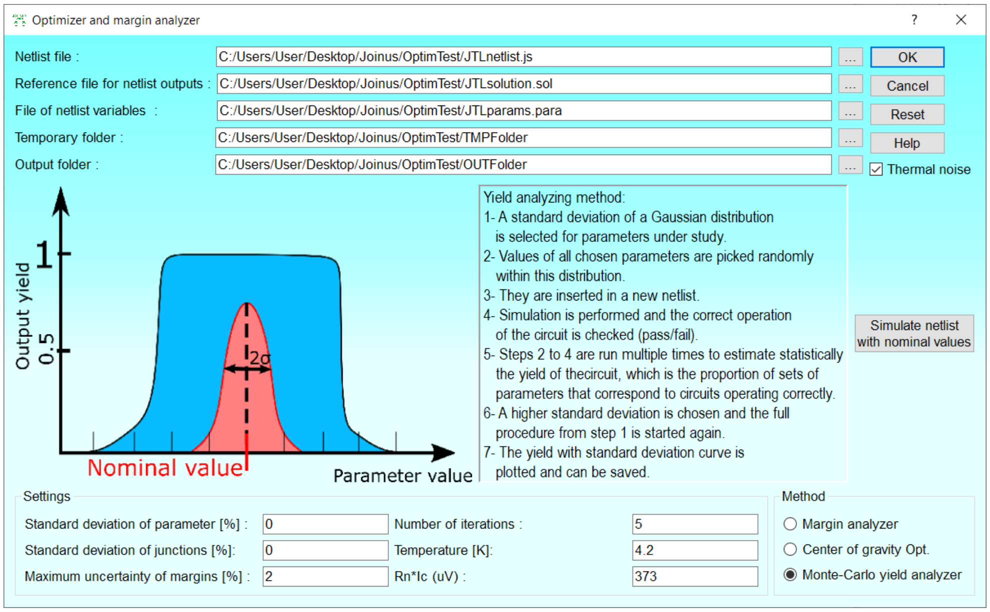

Figure 12 Window for the Monte-Carlo yield analyzer. This window is part of the "Optimizer and margin analyzer" tool.

III. JOINUS TYPES OF SIMULATION

JOINUS can perform different types of simulation through integrated routines. They are available from *Type of simulation* drop down menu of the interface window (see Figure 13).

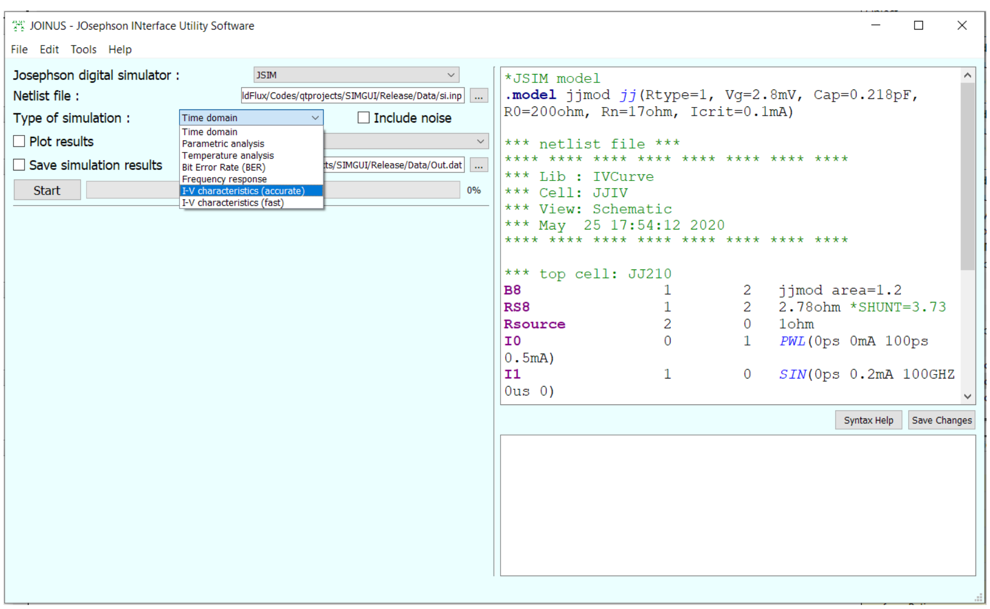

Figure 13 Interface of JOINUS software, version 2.0, showing the "Type of simulation" drop down menu.

III.1. Time domain simulations

This is the standard default mode of simulation. JOINUS calls the chosen Josephson digital simulator to simulate the netlist chosen in the *Netlist file* field and displays results in the console and on a plotter window if the *Plot results* checkbox is ticked. Noise is added to resistances if they *Include noise* checkbox is ticked. Simulations are done for the chip temperature specified in the *Tools* menu. See Figure 14.

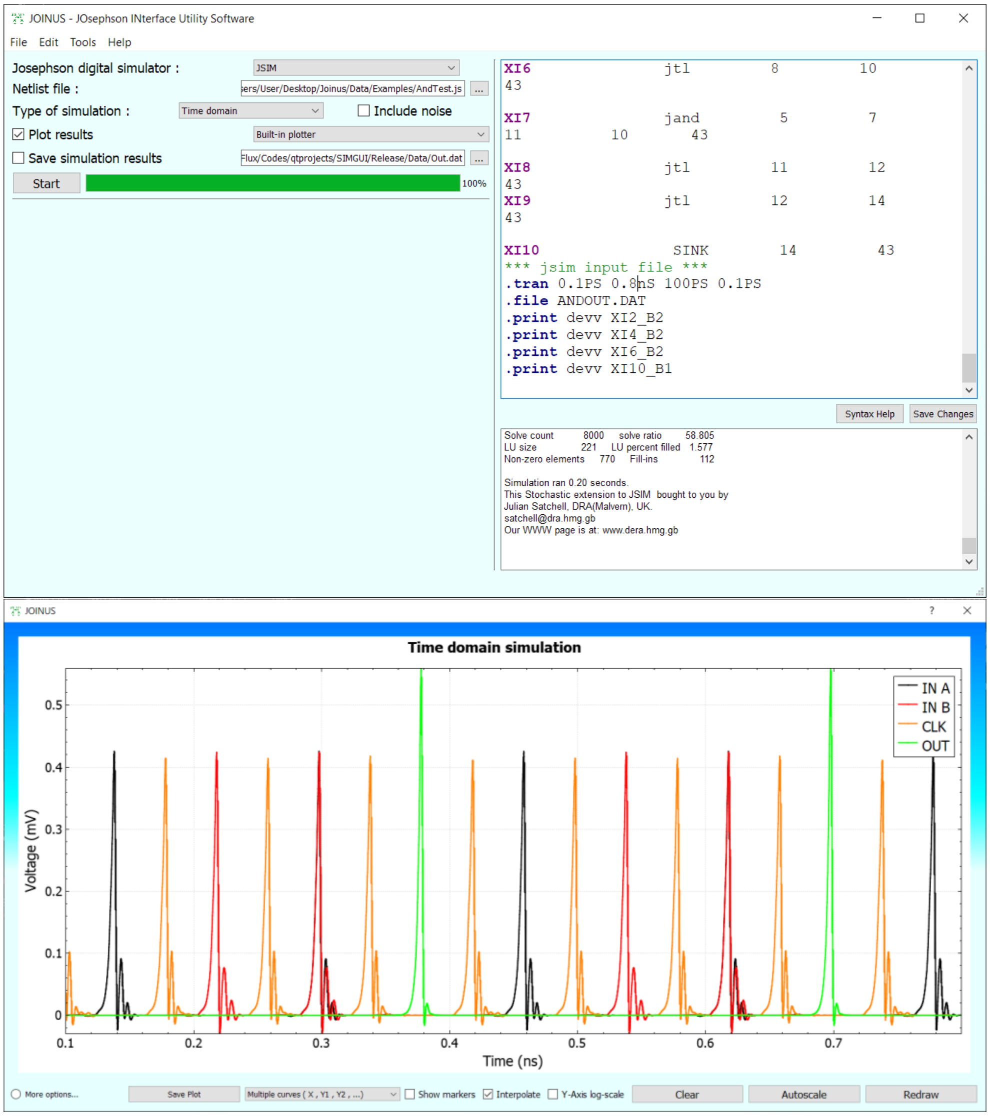

Figure 14 Time domain simulation front panel and results displayed on the JOINUS built-in plotter

III.2. Parametric analysis

The parametric analysis allows to choose a netlist parameter to sweep it across a specified range. The parameter name, nominal parameter value in the netlist, boundary values and number of steps within the range must be entered on the front panel (see Figure 15). By clicking on *Start* JOINUS simulates the netlist for each of the parameters within the specified boundaries and with the given number of steps. It generates the output results in the DATA folder at the program root directory. If the *Save simulation results* checkbox is ticked, the outputs will be stored in the destination folder defined by user. If the JOINUS built-in plotter is used it is possible to visualize the evolution of results when the parameter under study is modified by sliding the cursor at the bottom of the graph (the left position corresponds to the minimum value while the right position corresponds to the maximum chosen boundary value). The value of the parameter is modified in the title of the graph when the cursor is moved.

Figure 15 Parametric analysis front panel with results displayed on the JOINUS built-in plotter.

III.3. Temperature analysis

The *temperature analysis* mode is a specific type of parametric analysis that involves changes in all elements of the netlist that depend on temperature (see Appendix). The front panel is shown on Figure 16 along with some results. As for the parametric analysis it is possible to visualize the evolution of results when the temperature is modified by sliding the cursor at the bottom of the graph (the left position corresponds to the minimum temperature while the right position corresponds to the maximum temperature). The value of the parameter is modified in the title of the graph when the cursor is moved.

Note that simulations are run by default with niobium as the superconductor. To simulate circuit netlists for another superconducting material, the critical temperature Tc must be changed in the *Advanced settings...* option of the *Tools* menu.

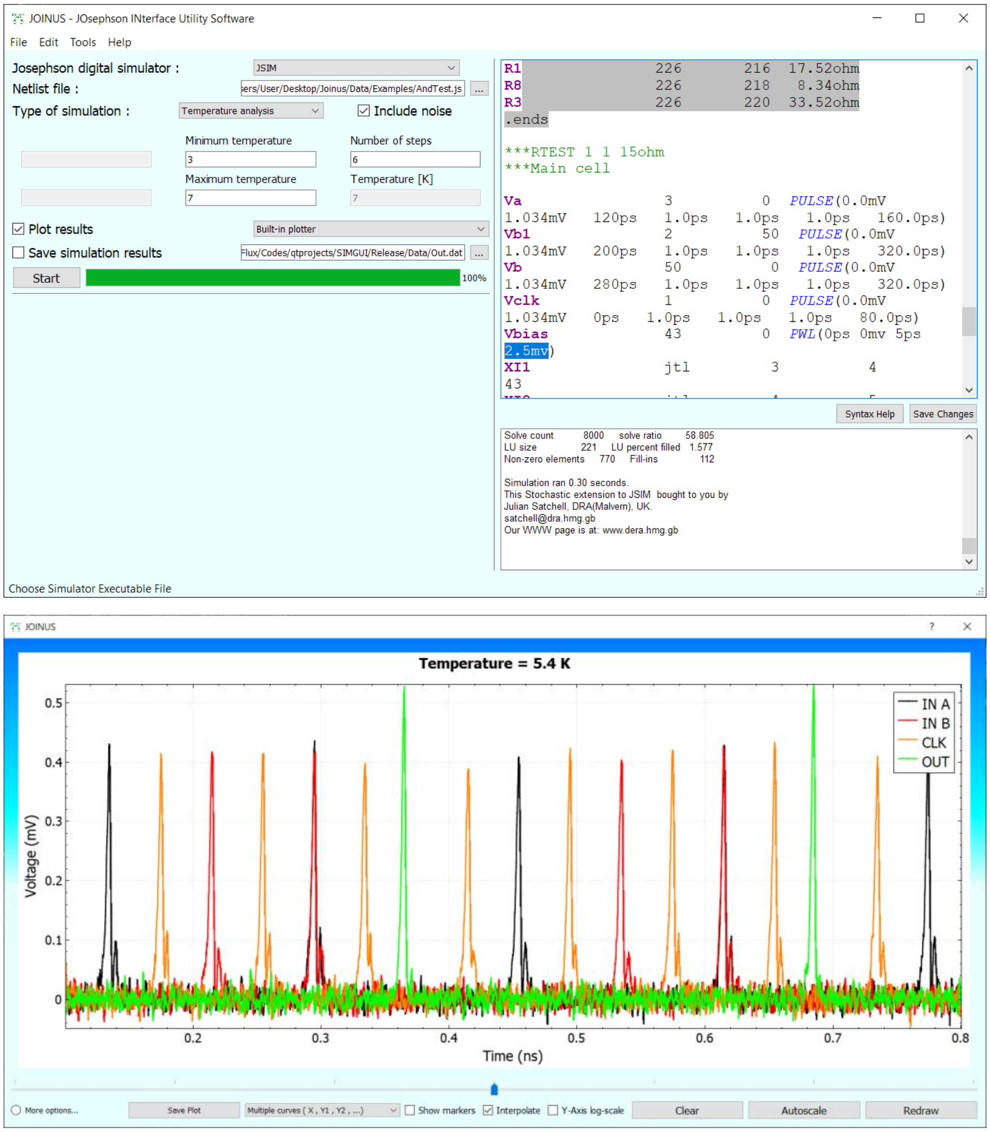

Figure 16 Temperature analysis front panel with results displayed on the JOINUS built-in plotter.

III.4. Bit Error Rate (BER)

For BER analysis, JOINUS needs the name of the element that is varied, its nominal value, along with boundary values and the number of steps within the chosen boundaries. The *multiplier* field is the ratio of the number of input pulses divided by the expected number of output pulses. For instance, if one expects only one output pulse for two input pulses the multiplier value is 2. By clicking on the *Start* button JOINUS simulates the netlist for each of the steps and generate temporary outputs for each step. JOINUS will store the data points for BER after processing the outputs if the *Save simulation results* checkbox is ticked. Figure 17 shows the front panel for the BER analysis along with results shown in the built-in plotter window.

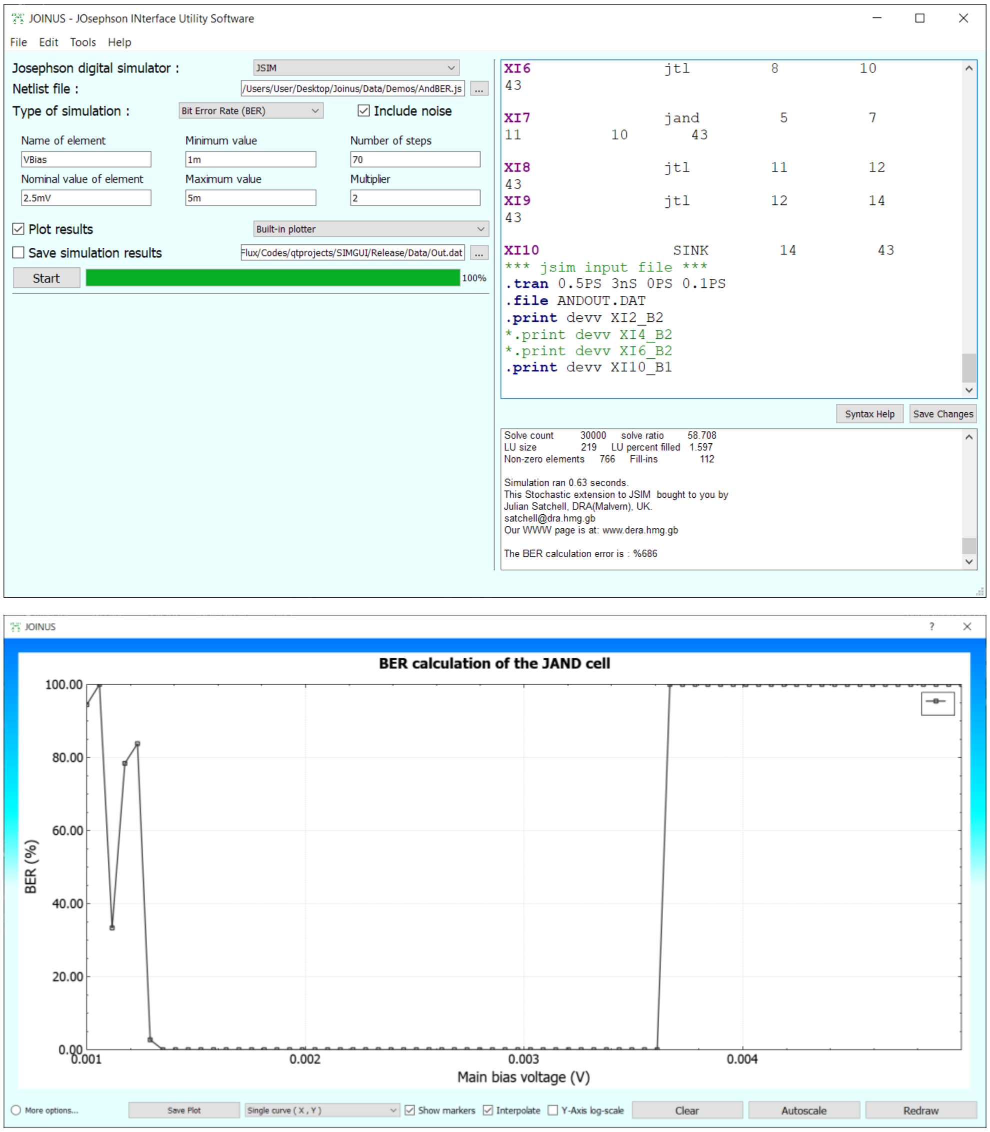

Figure 17 Bit Error Rate front panel with results displayed on the JOINUS built-in plotter for an AND cell.

III.5. Frequency response

This mode of operation is useful to test the frequency range of operation of a circuit. It applies a pulse train at the input node of the circuit netlist and counts the number of output pulses.

The user must enter the following inputs in the parameter fields of the front panel :

- the node number of the pulse train generator and its name,

- the minimum and maximum frequencies of the pulse train,

- the number of steps of the parameter value to be swept,

- the multipliers separated by commas. The multipliers are the number of input pulses divided by the number of expected output pulses. (Ex : 1,4,2 for one input and 2 outputs with an expected number of pulses of the first output of a quarter of the number of input pulses, and an expected number of pulses of the second output of half the number of input pulses.)

The phase or voltage of the input and output junctions to be monitored must be added in the netlist.

Example: .PRINT DEVV B1_X1 (the first line is always the input)

.PRINT PHASE B2_X2 (the other lines are always the outputs)

The front panel for frequency response along with an example of simulation result for a 3-bit counter are shown in Figure 18.

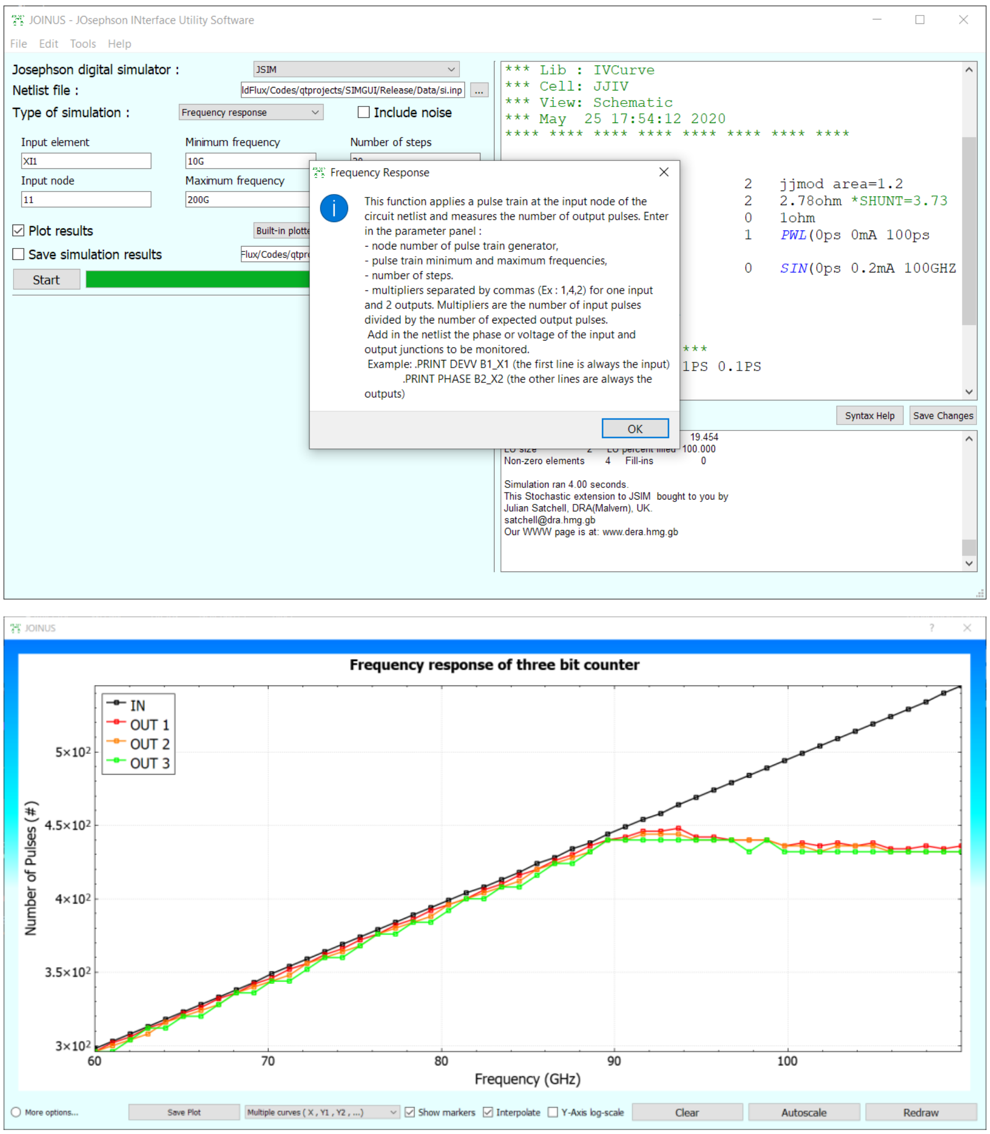

Figure 18 Frequency response front panel along with results displayed on the JOINUS built-in plotter

III.6. I-V characteristics

There are two modes to simulate current-voltage characteristics (I-V characteristics) based on a fast algorithm (but less accurate) or an accurate algorithm (but that takes longer to simulate). JOINUS uses the first output of the netlist as Current (Y-axis) and the second output as Voltage (X-axis) and plot them versus each other. The user must enter the current source name, the boundary values for the current, the number of points per cycle for the accurate algorithm (or the averaging window size for the fast algorithm) and the number of cycles (number of times the current is swept between its boundary values), as shown in Figure 19. After that the program will run by start button and draw the I-V curve. For the accurate, but slower, algorithm based on adaptive simulations, choose the *Adaptive step width* from the *Tools* menu, described above.

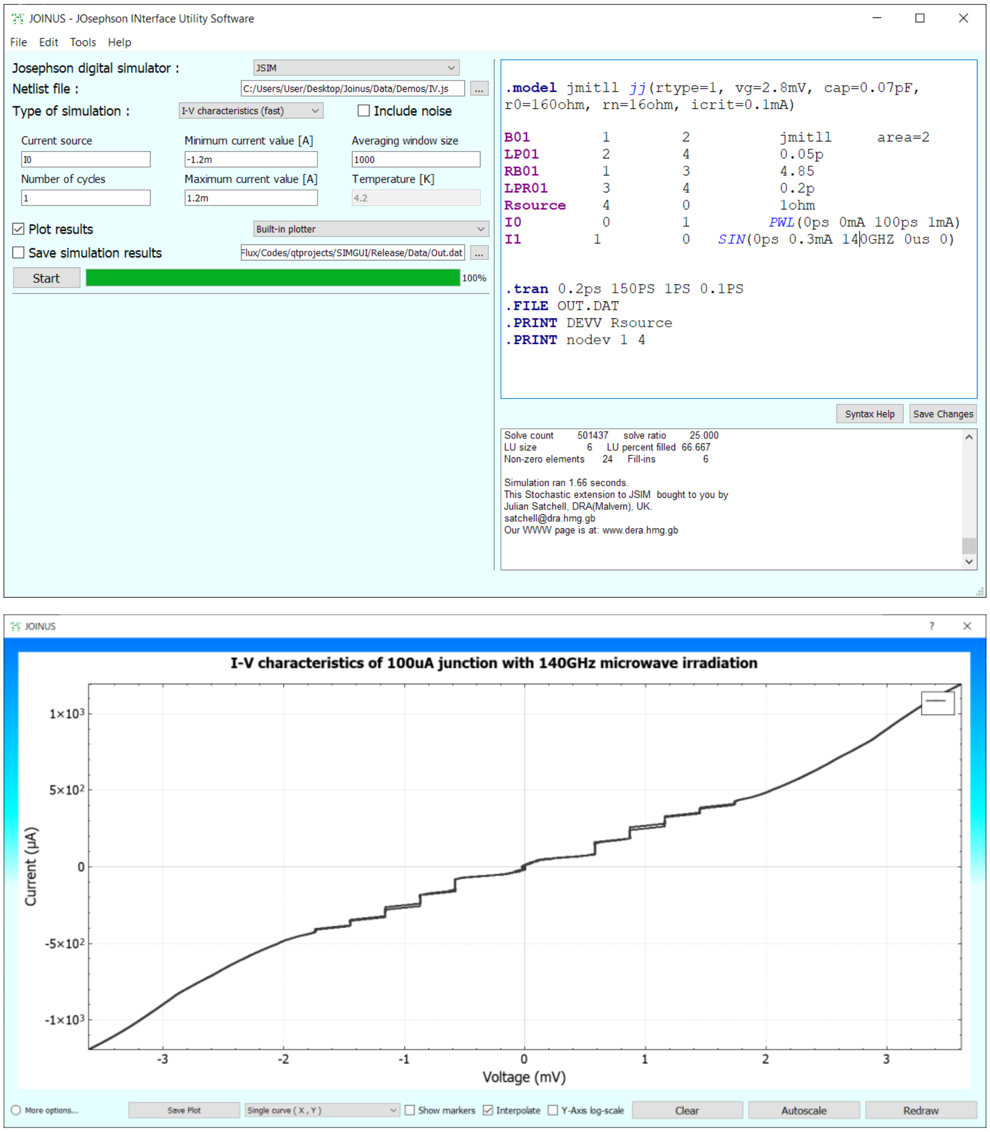

Figure 19 I-V curve calculation front panel along with an I-V curve in presence of a 140 GHz microwave signal displayed on the JOINUS built-in plotter.

REFERENCES

[1] C. J. Fourie *et al.*, "ColdFlux Superconducting EDA and TCAD Tools Project: Overview and Progress," *IEEE Transactions on Applied Superconductivity*, vol. 29, no. 5, pp. 1–7, Aug. 2019.

[2] F. E. S and V. D. T, "A Josephson integrated circuit simulator (JSIM) for superconductive electronics application.," *Ext Abstr 1989 Int Supercond Electron Conf ISEC 89*, pp. 407–410, 1989.

[3] J. A. Delport, K. Jackman, P. l Roux, and C. J. Fourie, "JoSIM—Superconductor SPICE Simulator," *IEEE Transactions on Applied Superconductivity*, vol. 29, no. 5, pp. 1–5, Aug. 2019.

[4] S. R. Whiteley, "Josephson junctions in SPICE3," *IEEE Transactions on Magnetics*, vol. 27, no. 2, pp. 2902–2905, Mar. 1991.

[5] S. Polonsky, P. Shevchenko, A. Kirichenko, D. Zinoviev, and A. Rylyakov, "PSCAN'96: new software for simulation and optimization of complex RSFQ circuits," *IEEE Transactions on Appiled Superconductivity*, vol. 7, no. 2, pp. 2685–2689, Jun. 1997.

[6] T. Q. Company, "Qt | Cross-platform software development for embedded & desktop." [Online]. Available: https://www.qt.io. [Accessed: 18-Feb-2019].

[7] "Qt Plotting Widget QCustomPlot - Introduction." [Online]. Available: https://www.qcustomplot.com/. [Accessed: 19-Jul-2019].

[8] A. Barone and G. Paterno, *Physics and application of the Josephson effect*. Canada: Wiley-InterScience Publications, 1982.

[9] S. Fujita and S. Godoy, *Quantum Statistical Theory of Superconductivity*. Springer Science & Business Media, 1996.

[10] J. Bardeen, L. N. Cooper, and J. R. Schrieffer, "Theory of Superconductivity," *Phys. Rev.*, vol. 108, no. 5, pp. 1175–1204, Dec. 1957.

[11] M. Tinkham, *Introduction to Superconductivity*. Courier Corporation, 1996.

[12] A. A. Golubov, M. Yu. Kupriyanov, and E. Il'ichev, "The current-phase relation in Josephson junctions," *Rev. Mod. Phys.*, vol. 76, no. 2, pp. 411–469, Apr. 2004.

[13] C. K. Poole, H. A. Farach, and R. J. Creswick, *Handbook of Superconductivity*. Academic Press, 1999.

[14] E. W. Lim and R. Ismail, "Conduction Mechanism of Valence Change Resistive Switching Memory: A Survey," *Electronics*, vol. 4, no. 3, pp. 586–613, Sep. 2015.

[15] H. Nyquist, "Thermal Agitation of Electric Charge in Conductors," *Phys. Rev.*, vol. 32, no. 1, pp. 110–113, Jul. 1928.

[16] J. B. Johnson, "Thermal Agitation of Electricity in Conductors," *Phys. Rev.*, vol. 32, no. 1, pp. 97–109, Jul. 1928.

APPENDIX I – INFLUENCE OF THE TEMPERATURE ON SIMULATIONS

The temperature of the superconducting circuits has an influence on several parameters and elements used in the simulations performed by JOINUS. It modifies the Cooper pair density and therefore most of the characteristics of superconductors and Josephson junctions, such as the critical current, the critical magnetic field, the London penetration depth and the energy gap. A change of temperature has an impact on active and passive elements in any circuit. Another consequence of a temperature change is a modification of thermal fluctuations and thermal noise due to the charges flowing into the resistances of the circuit (like the shunt resistors of Josephson junctions or the bias resistors).

Gap energy

From BCS theory the general expression of the gap voltage is given by [8]–[10],

$$\Delta_j = \sum_k \sum_i v_{ji} \Delta_i \frac{\tanh\left(\frac{\beta E_K^{(i)}}{2}\right)}{2E_K^{(i)}}$$

where :

- β is the temperature-dependent parameter defined as $1/k_B T$ where k_B is the Boltzmann constant,

- E is the total energy of the electrons assembled in Cooper pairs: $E = (\Delta^2 + \epsilon^2)^{\frac{1}{2}}$,

- v_{ji} accounts for interactions between electrons of the Cooper pairs,

- K is the state of the electron spin that can be up or down.

We can assume gap energy is symmetric for superconductors in element form (like Nb, Al, …). By solving this equation at the bulk limit and for element superconductors ($\Delta_i = \Delta_j = \Delta$),

$$1 = V_0 N(0) \int_0^{\hbar\omega_D} \frac{dE}{\sqrt{E^2 + \Delta^2}} \tanh\left(\frac{\sqrt{E^2 + \Delta^2}}{2k_B T}\right)$$

The gap energy value Δ can be estimated as [11],

$$\Delta = \Delta_0 \sqrt{\cos\left(\frac{\pi}{2}(T/T_C)^2\right)}$$

where Δ_0 is calculated from the weak coupling limit,

$$2\Delta_0 = 3.53 k_B T_C$$

Normal resistance of Josephson junctions

The value of the normal resistance of a Josephson junction is [12] :

$$R_N = \frac{\pi\Delta}{2eI_0} \times \tanh\left(\frac{\Delta}{2k_B T}\right)$$

Critical currents for bulk superconductors and Josephson junctions

The critical current relation of the superconductor can be estimated as [13] :

$$I_C = I_0(1 - (T/T_C)^2)\sqrt{1 - (T/T_C)^4}$$

Normal metal resistance

The dependence of normal metal resistance to temperature can be calculated from the more generic form of Ohm’s law [14]:

$$J = \sigma_0 Eexp\left(-\alpha T^{-1/n}\right)$$

$$R = R_0 \exp\left(\alpha T^{-\frac{1}{n}}\right) \xRightarrow{linear\ estimation} R = R_0(1+\propto (T - T_0))$$

The value of *n* is 1 for pure ohmic conduction and can vary if there are impurities. For α,

$$\alpha = \frac{E_C - E_F}{k_B}$$

E_C is the conduction band energy, E_F is the Fermi level energy and k_B is the Boltzmann constant.

Thermal noise

The Johnson thermal current noise per unit bandwidth flowing through a resistance R is [15] [16] :

$$I_N = \sqrt{\frac{4k_B T}{R}}$$

APPENDIX II – SQUIDMAP (SQUID MAGNETIC PATTERN SIMULATOR)

The SQUID MAgnetic Pattern (SQUIDMAP) software, as its name suggests, was developed to calculate the magnetic patterns for SQUIDs, e.g. the dependence of the critical current of a SQUID versus the external applied magnetic field. This tool is helpful to design magnetic field-robust cells for digital electronics, or other types of sensors. At this stage it can handle superconducting loops with either one (RF-SQUIDs), two (DC-SQUIDs) or three (bi-SQUIDs) junctions. As an example the contours of phases of Josephson junctions of a DC-SQUID are shown in Figure 20 while the magnetic pattern is shown in Figure 21. The blue curve of Figure 22 gives the relation between the phases of each junction of the DC-SQUID as a result of the flux quantization constraint in the SQUID loop. The black and red curves are the contours in the phase space corresponding to the maximum total current that can flow through the SQUID. Black curves are the physical stable solutions while red curves correspond to unstable (mathematical) solutions. The intersections of blue and black/red curves give the critical current of the SQUID for a given magnetic field. Note that, depending on the size of the loop (and value of the SQUID inductances) several solutions can exist, some of them being metastable. By sweeping the external field SQUIDMAP can construct the magnetic pattern of the DC-SQUID, shown in Figure 21. One can observe that in this particular case two solutions exist for some magnetic field values.

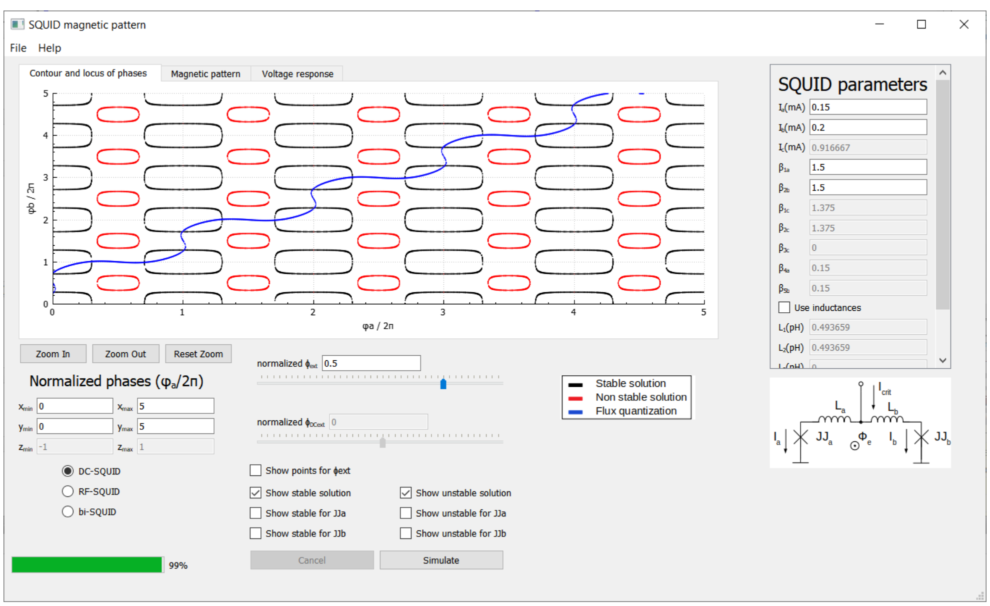

Figure 20 SQUIDMAP software front panel for a DC-SQUID

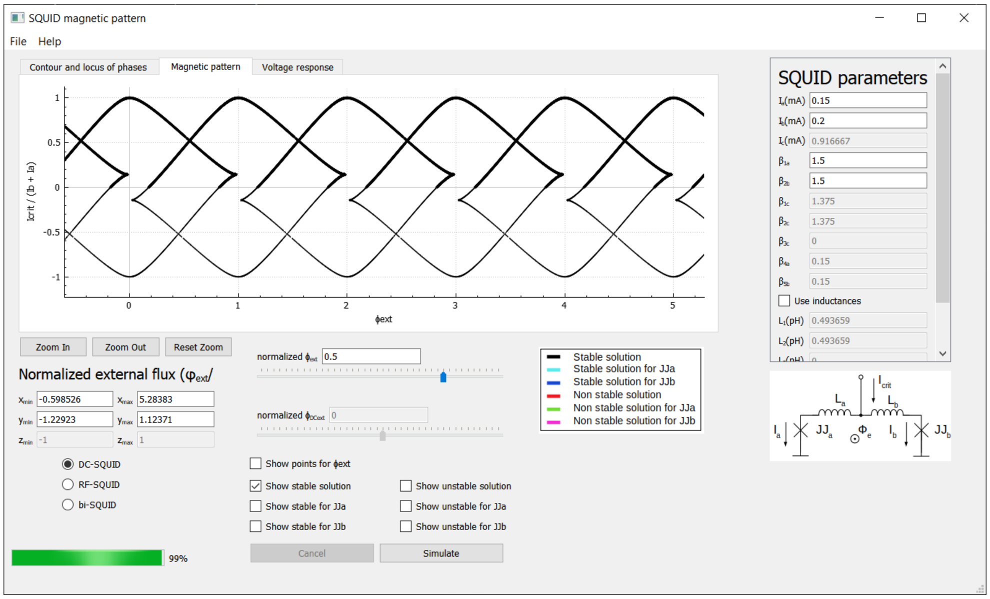

Figure 21 Magnetic pattern of a DC-SQUID determined by SQUIDMAP.