

# A MATLAB TOOL FOR (NOT ONLY) SWITCHED CIRCUITS ANALYSIS

Ing. Tomáš LUKL, Doctoral Degree Programme (3)  
Dept. of Telecommunications, FEEC, BUT  
E-mail: thomlu@eabrnno.cz

Supervised by Dr. Vít Novotný

## ABSTRACT

The article deals with a program SCISIP, which is a program made in mathematical and programming tool Matlab. SCISIP is created for simulation of general not only switched circuits. Reader can find a description of open and modular structure of the program as well as used algorithms in the most important module – simulation core. In the end one can find simulation results of SI lossless integrator.

## 1 INTRODUCTION

General switched circuits are circuits with quantum signal processing. Switch is the most used electrical element in these circuits. In real world, these switches are usually realized by transistors in CMOS technology (like IC in most cases). This is for case of externally controlled switches (filters). While in internally controlled switches (Switch Mode Power Supplies) most used elements for switching diodes, thyristors are used. There is also 3<sup>rd</sup> group of switched circuits – mixed controlled switches (internally as well as externally controlled switches) which are used in  $\Sigma$ - $\Delta$  converters.

During testing stage of development of a new switched circuit we want to analyze it using a computer. We can use Spice-like simulators and transistor-level circuit models. But in this case, simulator spends a lot of computing time to calculate a circuit response in the switching instants (which we are not so interested in) because of nonlinear character of switch models. Using linear or even ideal models of switches is not so time consuming and not so inaccurate. But in this case, Spice-like simulators have problems using ideal switch models. This leads for a need to develop a new tool for switched circuits analysis. There were developed a lot of these tools in the past, e.g. MALINSC, MATSC, COCOSC, SWANN. But these tools are either old (so no longer supported) or very specific (for a closed group of switched circuits, e.g. switched capacitors – SC or just linear SC).

In the previous paragraph the main reason why to start development of a new switched circuits analysis tool was presented. Most important properties were set: ability to analyze not only general switched circuits, open and modular structure, easy to use environment and availability to the universities. Its name is SCISIP – Switched Circuits Simulation Program.

## 2 PROGRAM STRUCTURE

### 2.1 MATLAB ENVIRONMENT

SCISIP is still being developed in Matlab environment (in this moment, Matlab r14sp1). This has some advantages:

- C++ or Java programming language similarity.
- There are many of predefined mathematical functions in Matlab.
- Matlab core itself works with vector and matrices, which are very useful for algorithms formation (modified nodal analysis – MNA, Newton-Raphson iteration method – NR or the other methods for numerical integration).
- Effective creation of graphical user interface – GUI.
- Classical OOP technique is implemented (and used).

Each advantage has its disadvantage. So in this case:

- Matlab has to be installed and executed. There is a tool called Matlab Compiler which allows forming a stand-alone application but this is useless for such big project like SCISIP.
- There is a limited group of graphical elements of GUI which we can use. There are also problems while using OpenGL graphical rendering mode (however this mode is very effective).
- Common problem in these days: a newer version of software, a bigger demand on hardware. This true holds double for Matlab.
- Mutual incompatibility was a big problem in previous majority releases of Matlab (especially between versions 4 and 5). This can be quite a big problem in large projects.

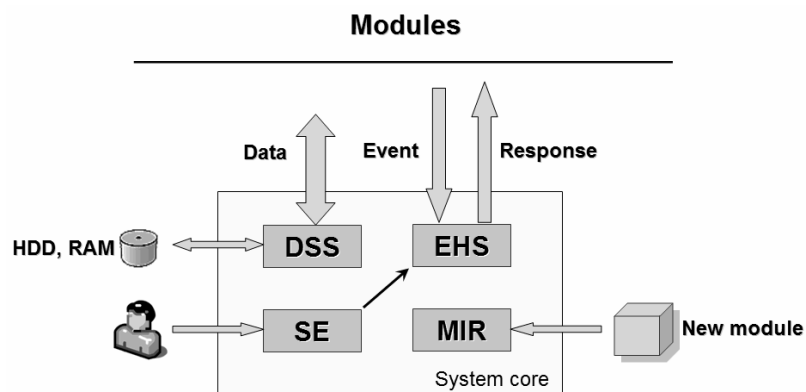
### 2.2 MODULARITY

The most important requirement was modular and open structure. Each module represents logically specified part of whole system. Open structure represents the fact, that a new module can be easily “plugged” into existing system which allows a user to extend its functionality. Additionally, the source code of the program will be (after completion) available (except the system core source code).

The main part of system structure is the **system core**. Its main purpose is to handle **events** from all modules and perform a **requested action**. A schema of system core is shown in the Fig. 1. There are four function blocks in system core:

- EHS – Event Handler System. A module calls this block and passes an event ID. EHS looks for this ID in the event table and performs requested operation (e.g. – simulation module finishes analysis and sends an event to inform, that analysis is complete. EHS looks at the table and calls a module, which displays the results in a graph). Event ID is a 16 bit number of class “unsigned integer”.

- DSS – Data Storage System. The big issue is how to store a large amount of data within multiple modules (shared data) or how to store a module’s data which need to be persistent. In Matlab, there is a function for a persistent variable declaration, but this is not too smart procedure. DSS provides uniform access to modules data through a structure, which is saved in RAM or HDD. EHS event table or EMHS (error and message handler system) data are stored using DSS.
- SE – SCISIP Exec provides a Matlab command line interface for executing SCISIP functions. SE is the first point where users enter SCISIP system. They can either execute GUI or command line analysis procedure.
- MIR – Module InstalleR provides a user friendly tool for upgrading an existing module or installing a new one.



**Fig. 1:** *Designed system structure.*

Modules themselves are another part of program structure. There are naturally two types of modules. Internal – predefined modules, which can not be removed (they can be just updated). Simulation core, already mentioned EMHS module, LSM (language support module) and GHM (GUI Handler Module) are internal modules. EMHS module provides uniform way of displaying errors, warnings and other messages, LSM module provides language support for GUI and EMHS through language definition files (default language of SCISIP is English and another language definition file comes with Czech). External modules then are user defined modules.

There is no strictly defined or required structure of user modules. They just have to offer necessary functions for accessing public module data (e.g. in case of simulation core a data structure with analysis results).

### 2.3 INPUT FILES DEFINITION

There are two input files of SCISIP from the simulation core point of view: netlist file and model file.

**Netlist** is a structure which describes how the single electrical parts are interconnected. Additionally, it provides information which algorithm shall be used, the time span for transition analysis, etc. In SCISIP, netlist is a clear text file, which has simple and intuitive syntax (structure). It consists of commands. Each command **must be** on a single line. Each line can begin with a special character:

- % - introduces a comment line.
- \$ - introduces a key word. In this moment, there are defined about 15 keywords. These keywords are in fact a set-up of the analysis process.
- English letters – introduces a line of an element wiring.
- The others are ignored.

An example of a netlist file is shown in the Tab. 1 on the left. This netlist describes a circuit shown in the fig. 2.

**Model** is also a clear text file which describes a model of an element. Its syntax is very similar to the syntax of Netlist file just with model specific keywords.

<pre> \$ analysis bias algorithm AL maxiter 100 deltamax 1u analysis tran algorithm TR maxiter 20 tspan 0 3m 'Dig' dsource name='Dig' T=40u \$ U 1 0 U [U=5] R 1 3 R [R=1k] I1 1 3 I [I=400u] I2 1 2 I [I=200u] I3 1 6 isin [Ia = 130u;f = 1k] T1 5 2 0 !nmos [Cgs=20p;L=2u;w=20u] T2 4 2 0 !nmos [Cgs=20p;L=2u;w=20u] T3 4 3 0 !nmos [Cgs =20p;L=2u;w=40u] S1 6 2 sw [Ron=1u;Roff=0.1G;clk=~dig'] S2 2 5 sw [Ron=1u;Roff=0.1G;clk='dig'] S3 2 4 sw [Ron=1u;Roff=0.1G;clk=~dig'] </pre>	<pre> model diode modeltype math modfuntype nlin pindef A K inputs uAK outputs iA,iA_uAK moddef::   Is = 10e-15;   Ut = 25.5e-3;   ud = uAK;   if ud&gt;1     id = Is*exp(1/Ut)*(1+ud/Ut-1/Ut);     did = Is*exp(1/Ut)/Ut;   elseif ud&lt;0     id = Is*(1/(1-ud/Ut)-1);     did = Is/Ut * 1/(1-ud/Ut)^2;   else     id = Is*(exp(ud/Ut)-1);     did = Is/Ut*exp(ud/Ut);   end   iA = id;   iA_uAK = did; \$endef </pre>
---	--

**Tab. 1:** Example of Netlist file (left) and Model file (right) with a nonlinear static model of a diode.

### 3 UTILIZATION OF ANALYSIS METHODS

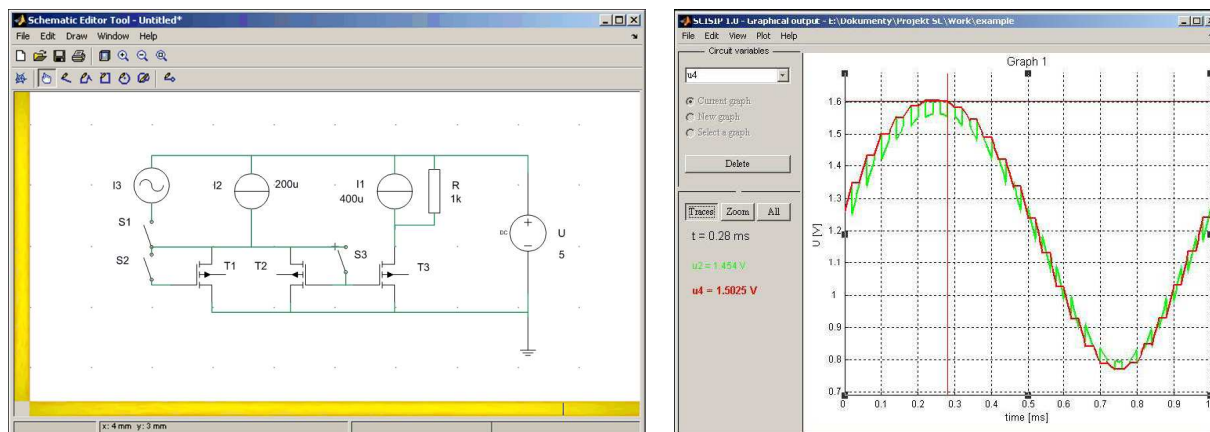
The simulation core contains mathematic apparatus which can solve system of nonlinear differential equations which mathematically describes analyzed circuit. Algorithms can also handle with inconsistent conditions, which appear when model of ideal switches is used (e.g. we interconnect two capacitors with different initial voltage using an ideal switch). Following methods were used:

- We solve a system of nonlinear differential equations within the program loops. Finally we must solve a big system of linear algebraic equations. LU decomposition method is used.
- System of nonlinear equations is solved by NR iteration method. When this method fails, homotopy method is alternatively used.
- System of nonlinear differential equations is solved by trapezoid or BDF methods.

- In the case of linear switched circuits, backward Laplace transform is used for analysis.

## 4 EXAMPLES

An example of analysis of a switched circuit using SCISIP is shown in the fig. 2. On the left is a schema of SI lossless integrator drawn in Schematic Editor. Transient response is shown on the right picture plotted in the window of Presentation Module.



**Fig. 2:** An example of analysis of SI lossless integrator using SCISIP. The schema in Schematic editor is shown on the left and the transient response on the right.

## 5 CONCLUSION

The new switched circuits simulation tool in Matlab was presented in the paper. The tool is still being developed, so presented results are the main functional fragments of whole program. After SCISIP will be completed, it can be used in some theoretical classes (like Circuit Theory, Analogue Circuits, etc.) and their computer practices.

## ACKNOWLEDGEMENTS

This paper was worked out under the support of FRVŠ 3248/2006/G1 project.

## REFERENCES

- [1] Ogrodzki, J.: Circuit Simulation Methods and Algorithms, CRC Press 1994, ISBN 0-8493-7894-X.
- [2] Lukl, T., Novotný, V., Mišurec, J.: Computer-Aided Circuit Analysis with Respect to Switched Circuits. WSEAS Transactions on Electronics, ISSN 1109-9445, 2005, vol. 2, p. 139 - 143.
- [3] Vlach, J., Bedrosian, D.: Analysis of Switched Networks. International Journal of Circuit Theory and Applications, vol. 20, p. 728.1-728.17, 1992.
- [4] Lukl, T.: Využití homotopií pro DC analýzu nejen spínaných obvodů. Elektrovue - Internetový časopis, (<http://www.elektrovue.cz>), ISSN 1213-1539, 2005.