

# Features of Using Models Pspice Software

<https://doi.org/10.31713/MCIT.2023.029>

Vasyl Zaiats

Department of Telecommunication and Computer Science  
Polytechnika Bydgoska (PBS)  
Bydgoszcz, Poland  
[zvmvz01@gmail.com](mailto:zvmvz01@gmail.com)

**Abstract**—The paper notes the features of the input language of this software package, methods of describing sentences of the input language, building own models of the studied object and using the models specified in the open library of this complex. Possible operating modes of the complex and methods of their effective use are described. The results obtained make it possible to carry out effective circuit and technical modeling and analysis of both known objects and newly created ones, while ensuring the economy of material and human resources.

**Keywords**—model; computer simulation; control directives; nonlinear signals; programming language; Pspice software packages.

## I. INTRODUCTION

In the analysis of dynamic processes of complex nature, it is advisable to use the discrete models, approaches to the construction of which are described in [1-4, 8-14].

When creating new real devices, studying unexplored physical or natural phenomena, building recognition and identification systems that have the desired characteristics of the information signal or unknown characteristics to be studied, it is advisable to conduct analytical or computer modeling and analysis, creating adequate mathematical models for project being developed or studied. This approach requires much less time and technical resources compared to physical or field experiment, especially at the preliminary stage of model development or analysis of a complex structured process in the absence of reliable a priori information about the environment and behavior of objects in it.

Recently, discrete models of systems [3, 4] are widely used in nonlinear dynamics, for which discreteness is inherent in the nature of the object of study, and is not a consequence of discretization of a continuous system.

The following features explain the expediency of using discrete models:

- simplicity of mathematical description in comparison with continuous models;
- the presence of a significantly wider range of dynamic modes compared to known models;
- infinite dimensionality, which allows to model each new harmonic of the process by introducing it into the

vector of state variables, while for continuous systems to solve this problem it is necessary to increase the dimension of the system;

- no need to determine the sampling step, assess the local and global errors of numerical methods, areas of stability and synchronization;
- convenience and better adaptability to the computer experiment compared to continuous models;
- the ability to predict the behavior of an object or phenomenon in the future based on incomplete input data.

There are a large number of specialized computer-aided design (CAD) systems designed for modeling simple and complex electronic circuits. Among them, the CAD software OrCAD should be especially noted, which allows the analysis of transient processes in digital and analog electronic circuits at the stage of their design, as well as their time, amplitude-frequency and phase-frequency characteristics. Schematic CAD can analyze the sensitivity of the circuit (reaction to small changes in the input signal and circuit elements), signal spectra in any nodes of the circuit and obtain information about other important features of the simulated electronic circuit. Another important distinguishing feature of circuit engineering CAD is that they have huge libraries of various electronic components, which greatly simplifies their use during the design of real electronic circuits.

However, it is worth noting that the used means of schematic technical design also have a number of significant shortcomings, among which the following should be noted first of all:

- complexity of the mathematical description of models of electronic components that are being formed;
- complexity of library development (existing component libraries do not always suit designers);
- lack of a systematic approach to the description of models of electronic components.

In general, usually all schematic technical CAD projects are descriptive in nature and the possibilities of using a hierarchical approach and modern modular programming tools in them are extremely limited. All component models, due to the lack of

mathematical blocks in schematic CAD, are described declaratively using linguistic programming tools, and modern visual programming tools are used only to form electronic circuit models from component models. These shortcomings make it somewhat difficult to describe complex electronic circuits and systems using modern schematic CAD tools.

Among the systems designed for schematic modeling of electronic devices, Pspice (OrCAD), Micro-Cap, Multisim (Electronics Workbench) (National Instruments Corporation), DIgSILENT PowerFactory [5-7] and SIMULINK (MathWorks) [12] program packages should be highlighted.

Discrete models are applicable both to the construction of devices with the desired modes and to the recognition and identification of situations in systems with complex dynamics and behavior.

The simplest and most convenient for the analysis of discrete models is a set of programs Pspice (Simulation Program with Integrated Circuit Emphasis) [11-14], which allows you to analyze them by relatively simple means, preserving the qualitative features of the process and in the first approximation to obtain analytical estimates to predict its behavior. This work is devoted to the description of the input language and features of using the Pspice software package.

## II. PSPICE INPUT LANGUAGE

Pspice software is designed to model analog, digital and analog-digital devices and identify the parameters of the models of the studied objects. Pspice input language is quite simple and easy to use [5, 12-14]. Compiling a task for execution begins with assigning names to the nodes of the schematic diagram, which can be integers from 0 to 9999 or alphanumeric characters. References to alphanumeric nodes are made in square brackets.

When you create a task, you specify an arbitrary file name that has a .CIR extension by default. The first taskbar is the title, which is then displayed in the source file (contains any text without Cyrillic). The comment is placed in "« ". Any text in the ribbon after ";" perceived as a comment. The task ends with the .END directive. The order in which the task bars are entered does not matter. The + character in the first position is used to continue a line in another line.

Pspice does not distinguish between uppercase and lowercase letters. Commas, sign equals, spaces are equivalent. You can describe several simulation tasks in one task, thus saving time on running each of the tasks. Compilation of the problem begins with the numbering of nodes and an arbitrary description of all elements of the schematic.

All sentences in the input language are divided into component descriptions and task management directives. The structure of the description of the components is as follows:

*<component name>, <number of two or more nodes>,  
<model name>, <numerical data>*

The component name consists of a sequence of Latin characters and numbers up to 133 characters long (8 characters recommended). The first character of the component name - the letters of the Latin alphabet from A to Z. The first character of the name determines the type of component:

B - helium arsenide field-effect transistor with n-type channel;

- C – capacitor;
- D – diode;
- E – voltage source, voltage controlled;
- F – current source controlled by current;
- G – voltage-controlled current source;
- H – voltage source controlled by current;
- I – independent current source;
- J – field-effect transistor with controlled p–n junction;
- K is the mutual inductance or magnetic core transformer;
- L – inductance;
- M – MOSFET – transistor;
- N – analog-to-digital converter;
- O – digital-to-analog converter;
- Q – bipolar transistor;
- R – resistor;
- S – voltage-controlled switch;
- T – delay line;
- V – independent voltage source;
- W – voltage-controlled switch;
- U – digital device;
- X – macro model.

Node numbers must follow in a specific order for each component. The component model name is optional. It is specified when creating your own model or using an existing one. Then following the numerical values of the component parameters. Scaling numbers is done with suffixes:

$$\begin{aligned}
 F &= 10^{-15}; \\
 P &= 10^{-12}; \\
 N &= 10^{-9}; \\
 U &= 10^{-6}; \\
 M &= 10^{-3}; \\
 K &= 10^3; \\
 MEG &= 10^6; \\
 G &= 10^9; \\
 T &= 10^{12}
 \end{aligned}$$

Component parameters are specified in two ways: directly in the sentence of the input file or using the .MODEL directive, which has the following structure:

*.MODEL <model name>, <type name>, <parameter name = value>, <permissible parameter scatter>*

The component type is defined as follows:

*RES- resistor  
CAP-capacitor  
IND-inductance  
D-diode  
NPN-bipolar n-p-n transistor  
PNP-bipolar p-n-p transistor  
UDLY - digital delay line*

The .MODEL directive specifies in parentheses a list of model parameter values:

*.MODEL RM RES (R=1.5 , TC1=0.2,TC2=0.05 )  
.MODEL CK CAP(C=1 , DEV= 0.1)*

The description of the model is realized in two sentences:

*RM 2 3 RM2*  
*MODEL RM2 (R=0.5 TR1=0.05 TR2=0.02)*

where the nonlinear resistance is calculated according to the formula:

$$RM2 = R \cdot (1 + TR1 \cdot (T - T0) + TR2 \cdot (T - T0)) \cdot (T - T0)$$

Quite similarly set by the value of nonlinear inductance and capacity.

When analyzing transients, you can specify a set of nonlinear time functions, the behavior of which is shown in Fig. 1-3. By varying the parameters of these functions, you can simulate signals of different shapes and durations, using the following names:

- EXP*( $y_1, y_2, t_1, t_2, t_3$ ) – exponential function;
- PULSE*( $y_1, y_2, t_1, t_2, t_3, t, T$ ) – impulse function;
- SIN*( $y_0, y_u, f, t_1, d, \varphi$ ) – harmonic function.

Note that setting a value of the parameter d other than zero in the harmonic function can simulate a vanishing sine wave or cosine.

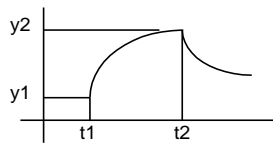


Figure 1. Example of exponential function

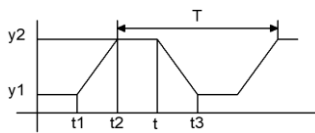


Figure 2. Example of impulse function

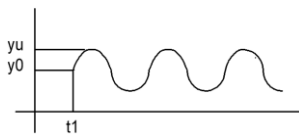


Figure 3. Example of harmonic function

To model the dependent current and voltage sources, the polynomial function POLY is used, which is approximated by the expression:

$$Y = P_0 + P_1 X_1 + P_2 X_2 + \dots + P_N X_N + P_{N+1} X_1^2 + P_{N+2} X_1 X_2 + \dots + P_{N+N} X_1 X_N + P_{2N+1} X_2^2 + \dots + P_{2N+N-1} X_2 X_N + \dots$$

At the N = 1

$$Y = f(X_1) = P_0 + P_1 X_1 + P_2 X_1^2 + \dots$$

At the N = 2

$$Y = f(X_1, X_2) = P_0 + P_1 X_1 + P_2 X_2 + P_3 X_1^2 + P_4 X_1 X_2 + P_5 X_2^2$$

At the N = 3

$$Y = f(X_1, X_2, X_3) = P_0 + P_1 X_1 + P_2 X_2 + P_3 X_3 + P_4 X_1^2 + P_5 X_1 X_2 + P_6 X_1 X_3 + P_7 X_2^2 + P_8 X_2 X_3 + P_9 X_3^2 + P_{10} X_1^3 + P_{11} X_1^2 X_2 + P_{12} X_1^2 X_3 + P_{13} X_1 X_2^2 + P_{14} X_1 X_2 X_3 + P_{15} X_1 X_3^2 + P_{16} X_2^3 + P_{17} X_2^2 X_3 + P_{18} X_2 X_3^2 + P_{19} X_3^3$$

This function can be applied to the construction of systems for modulation, in particular, to the analysis of amplitude and phase-modulated oscillations.

### III. BASIC TASK MANAGEMENT DIRECTIVES IN THE PSPICE ENVIRONMENT

The main task management directives in the Pspice environment allow you to perform 9 types of analysis:

- 1) analysis of the DC circuit;
- 2) multivariate analysis when changing the parameters of the scheme;
- 3) sensitivity of circuit characteristics to changes in parameters;
- 4) calculation of transient functions;
- 5) frequency characteristics of the linearized circuit;
- 6) spectral densities of internal noise;
- 7) calculation of transient characteristics;
- 8) spectral analysis;
- 9) static analysis by the Monte Carlo method.

The first mode is always implemented without any directives and a map of nodal potentials is issued. The .DP directive is used for complete information.

Multivariate analysis at the variation of a variable or several variables has the form:

.DC <name of first change >, <start value> <finish value>, <value of increment >, ..., <name of finish change >, <start value>, <finish value> < value of increment >

The sensitivity of the characteristics is calculated according to the directive:

.SENS < output variables >.

Let's say if the circuits consist of R1, R2, G3 and we have a record:

.SENS V(g), V(r).

then the values of dV (g) / dR1, dV (r) / dR1, dV (g) / dR2, dV (r) / dR2, dV (g) / dG3, dV (r) / dG3 will be calculated.

The .TF <output variable> <voltage or current source name> directive is used to calculate the transfer functions. For example, when calling the .TF V (5) VIN directive, the result will be:

dV(5)/dVIN.

Frequency characteristics are calculated according to the directive:

.AC <h> <start frequency> <end frequency>.

The results are output according to the .PRINT or .PROBE directive.

A directive is used to calculate transients:

*.TRAN* <output step> <end time> <start time>  
<maximum calculation step> *.UIC*.

If the maximum output step is not specified, it is automatically selected as <end time> / 50.

Before the calculation of transients, the calculation of direct current. The UIC parameter is used to cancel this mode.

But then it is necessary to set the initial conditions on the reactive elements. Example:

*C1 3 4 100Φ IC=0.5.*

The result of the analysis is displayed in the form of a table when using the *.PRINT*, *.PROBE* or *PLOTTER* directives. There are a number of additional directives [5, 6] for ease of operation in the Pspice environment

#### IV. CONCLUSION

The paper describes features the application and functionality of the basic version of the Pspice software package, its input language, ways to specify sentences in working files, use management directives, build your own models and use models of existing software in your own library. Note that the set of Pspice programs is open and can be replenished with new models when creating and writing to the library.

These specific examples of the description and analysis of nonlinear signals, generator circuits and non-linear power supplies can be effectively used both in the analysis of similar processes and in the development of own models of objects.

The results presented in the work can be used both in the course of research and in the development of the software complex Pspice [5, 12 14] and will be useful when using other software products [5-7] for modeling and developing automated design systems for radio electronic devices and their modes of operation.

Thus, attention is paid to the basic version of Pspice because the description of the models themselves in modern software complexes is often linguistic in nature, which complicates their effective application and modification.

#### REFERENCES

- [1] V.M. Zaiats, "Constraction and analises of model descrete oscillation systems," Cybernetics and Systems Analysis, vol. 2, 2000, pp. 161–165.
- [2] V.M. Zaiats, "The Models Descrete Oscillation Systems," Computer Tegnology of Print, 2011, pp. 37–39.
- [3] V.M. Zaiats, "Methods, Algorithms and Software for Simulation and Analysis of the Dynamics of Complex Objects and Systems Based on Discrete Models," Lviv: New World, 2009.
- [4] S.F. Sharkovsky, A.G. Kolyada, V.V. Sivak, and A.N. Fedorenko, "Dynamics of one-dimensional mappings," Kyiv: Ukraine, 1989. 216 p.
- [5] "DIgSILENT PowerFactory. Version 14.0," Gomaringen: DIgSILENT GmbH, 2011.
- [6] "Power System Solutions – DIgSILENT." [Online]. Available: <http://www.digsilent.de>. [Accessed: October 22, 2023].
- [7] "DMCC Engineering." [Online]. Available: <http://dmcc.com.ua/>. [Accessed: October 22, 2023].
- [8] J. Vlach and K. Singhall, "Computer Methods for Circuit Analysis and Design," New York: VNRC, 1980.
- [9] L. Feldman, A. Petrenko and O. Dmitrieva, "Numerical Methods in informatic," Kiev: BHV, 2006.
- [10] G. Holl and J.M. Watt, "Modern Numerical Methods for Ordinary Differential Equations," Oxford: Clarendon press, 1979.
- [11] "MicroSim PSpice A/D & Basics+," Irvine: MicroSim Corporation, 2012.
- [12] V. Zaiats, "Methods and means of computer information technologies in applied applications," Lviv: Ukrainian technologies, 2017.
- [13] V.M. Zaiats, "Elements of the Theory Information Technology in Applied Applications," Rivne: National University of Water and Enviromental Engineering, 2023.
- [14] V.M. Zaiats, J. Majeviskyi, and M.M. Zaiats, "Features and functionality of the software complex Pspice," Computer-integrated technologies: education, science, production, vol. 42, 2021. pp. 50–57. DOI: 10.336910/67751-2524-0560-2021-42-48.