Trademarks and Registered Trademarks

- MATLAB and Simulink are registered trademarks of The Mathworks, Inc.
- Cadence, OrCAD, the OrCAD logo, OrCAD Capture, PSpice and PSpice A/D are trademarks of Cadence Design Systems, Inc.
- Microsoft, MS-DOS, Windows, the Windows logo, Windows NT and other Microsoft products are registered trademarks or trademarks of Microsoft Corporation.
- Adobe, the Adobe logo, Acrobat, the Acrobat logo, Exchange and Postscript are trademarks of Adobe Systems Incorporated or its subsidiaries, which may be registered in some countries or regions.
- All other product names are trademarks or registered trademarks of their respective owners.
- Unauthorized reprinting, reproduction or copying of this manual, in whole or in part, is forbidden.
- This manual is subject to change without prior notice.
Table of Contents

Introduction 1

Welcome to SLPS ................................................................. 1
SLPS overview ..................................................................... 1
How to use this guide ......................................................... 2

Chapter 1  Installation ......................................................... 1-1

Section 1  System requirements ........................................ 1-1
  Subsection 1 Required software ........................................ 1-1
  Subsection 2 License ......................................................... 1-1

Section 2  Installation ......................................................... 1-2
  Subsection 1 Installing SLPS ............................................ 1-2
  Subsection 2 Setting up MATLAB path .......................... 1-2
  Subsection 3 Uninstalling SLPS ....................................... 1-2

Chapter 2  Tutorial .............................................................. 2-1

Section 1  Creating a schematic ......................................... 2-1
  Subsection 1 Starting OrCAD Capture ............................ 2-1
  Subsection 2 Creating a project ....................................... 2-1
  Subsection 3 Editing a schematic ..................................... 2-2

Section 2  Simulation using PSpice ................................. 2-3
  Subsection 1 Setting up analyses .................................... 2-3
  Subsection 2 Running a simulation .................................. 2-4
  Subsection 3 Verifying results .......................................... 2-4

Section 3  Creating and setting up a block diagram .......... 2-5
  Subsection 1 Starting MATLAB ..................................... 2-5
  Subsection 2 Creating a block diagram ......................... 2-5
  Subsection 3 Setting up SLPS block parameters .............. 2-6
## Table of Contents

**Chapter 3 Creating Simulation models** 3-1

### Section 1 Preparing PSpice circuits 3-1
- Subsection 1 Creating a CIR file 3-1

### Section 2 SLPS block 3-2
- Subsection 1 SLPS settings window 3-3

### Section 3 Simulink simulation parameters 3-6

**Chapter 4 Simulation** 4-1

### Section 1 Starting a simulation 4-1

### Section 2 How to set up window in the event of a convergence error 4-1

### Section 3 Verifying results 4-2
- Subsection 1 Output files in PSpice 4-2

**Chapter 5 Examples** 5-1

### Section 1 RC circuit (RCCIR) 5-1

### Section 2 Nonlinear load (NLLOAD) 5-3
- Subsection 1 Schematic 5-3
- Subsection 2 Simulink model 5-4
- Subsection 3 Results 5-4

### Section 3 PLL model (PLL) 5-5
- Subsection 1 PLL composition 5-5
- Subsection 2 PLL model incorporating VCO electric circuit 5-7

### Section 4 Switched reluctance motor control 5-8
Subsection 1  Analyzing magnetic field by analysis tool using finite element method 5-8
Subsection 2  Modeling of electric circuit using PSpice .......................... 5-9
Subsection 3  Simulink model .............................................................. 5-10

Section 5  Switching power supply .............................................. 5-12
Subsection 1  Electric circuit design using PSpice .............................. 5-12
Subsection 2  Control system design using Simulink .......................... 5-14
Subsection 3  Results ..................................................................... 5-15

Section 6  DC Motor Control System ............................................. 5-17
Subsection 1  Outline of DC motor control system .............................. 5-17
Subsection 2  Simulink Model ............................................................... 5-19
Subsection 3  Simulink-PSpice Model (Ideal OPamp) .......................... 5-20
Subsection 4  Simulink-PSpice Model (Device OPamp) ........................ 5-23

Chapter 6  FAQ ........................................................................ 6-1

Chapter 7  Detailed information .................................................. 7-1

Section 1  Data exchange between Simulink and PSpice ....................... 7-1
Section 2  How to determine PSpice transient analysis parameters .......... 7-2
Introduction

Welcome to SLPS

The PSpice SLPS Interface (hereafter abbreviated "SLPS") is an interface tool between The MathWorks' MATLAB/Simulink system simulator and the PSpice A/D electric circuit simulator (hereafter abbreviated "PSpice"), which was developed jointly by Cybernet Systems Co., Ltd., and Cadence Design Systems, Inc. Using SLPS, you can insert PSpice electrical circuits into Simulink models, and you can join systems models and electric circuit models which were previously handled separately. This system provides a modeling environment that combines the advantages of each of the simulators.

SLPS overview

Placing an SLPS block in a Simulink model enables Simulink to use the analysis engine of PSpice. In SLPS blocks, you can assign schematics for PSpice created with the OrCAD Capture schematic entry tool.
How to use this guide

This guide assumes that you are familiar with the operation of Microsoft Windows. It also assumes that you have a basic understanding of how Windows manages applications and files to start applications, and open and save your files. For more information on Windows operation, please refer to the Microsoft Windows User's Guide and/or other references.

In addition, SLPS is an interface tool between MATLAB/Simulink and PSpice, so this guide assumes that you are familiar with MATLAB/Simulink and PSpice, as well as their operation. For details on their respective simulator modeling techniques, analysis setting and result verification methods, please refer to the Simulink User's Guide or PSpice A/D User's Guide.
Chapter 1 Installation

Section 1 System requirements

Subsection 1 Required software

In order to install SLPS, the following products must first be installed on your computer.

- The MathWorks products R13 or higher
  - MATLAB 6.5 or higher
  - Simulink 5.0 or higher

- Cadence OrCAD products R10.0 SP2 or higher
  - Capture or Capture CIS
  - PSpice A/D
  Or install one of the following suite products
  - Unison EE
  - Unison Ultra
  - PDB Designer with PSpice
  Note: In order to use with OrCAD R10.0, SP2 (Service Pack 2) must be installed.

* Select the operating system and hardware which are required to run each product, according to their respective specifications.

Subsection 2 License

In order to perform simulations using SLPS, you must have licenses for the aforementioned The MathWorks products and the Cadence OrCAD products. Additionally, your Cadence license file must include the "PSpiceSLPSOpt" feature, which is an SLPS license.

When SLPS is activated, this SLPS license will be acquired along with a PSpiceA/D or Unison license.
Section 2  Installation

Subsection 1  Installing SLPS

Before installing SLPS, login with a user name having administrator privileges for the machine, and close all open applications.
SLPS installation is performed by running the SLPS installation file.

Subsection 2  Setting up MATLAB path

Set the following two paths of the installed SLPS directory in MATLAB:

- [SLPS]/SLPS
- [SLPS]/SLPSdemos
  
  [SLPS] is the directory where SLPS is installed.

See MATLAB Help or other references for the path setting method.
SLPS is now ready to use.

Subsection 3  Uninstalling SLPS

To uninstall SLPS, open "Add/Remove Programs" on the Windows control panel, select SLPS from the list, and press the "Add/Remove" button.
When deletion of SLPS is compete, delete the two SLPS paths set in MATLAB.

Subsection 4  Upgrading OrCAD products

When OrCAD products have been upgraded, SLPS need to be reinstalled. To reinstall SLPS, uninstallation of SLPS is required in advance. Setting up MATLAB path is not required unless MATLAB has not been upgraded at this time.
Chapter 2  Tutorial

Section 1  Creating a schematic

Subsection 1  Starting OrCAD Capture

Start OrCAD Capture from the Start menu
"Start" -> "All Programs" -> "OrCAD10.0" -> "Capture (CIS)"

Subsection 2  Creating a project

Create a project with Capture. From the Capture menu, select the following:
"File" -> "New" -> "Project..."

(1) Name: Specify a project name:
MOSCKT

(2) Create a New Project Using:
Select the purpose of the new project.
"Analog or Mixed A/D"

(3) Location: Specify the work directory to save your files.
C:/Work  (Location is arbitrary)

Press "OK".
In the displayed "Create PSpice Project" dialog box, select "Create a blank project" and press "OK".

The project window and schematic page will be displayed, as indicated below.

**Subsection 3 Editing a schematic**

Edit the schematic below. See the OrCAD Capture User’s Guide or other references for help on entering the schematic.

The circuit is comprised of a MOSFET, resistors and voltage sources. The input signal for verifying operation is a 0.5µ wide 1-shot pulse. To use in specifying output voltage from the SLPS block later, attach the node name "OUT" to the drain node.

Libraries used:
- M2N6800 (in POWERMOS.olb)
- R (in ANALOGold)
- VSRC (in SOURCE.olb)

After you finish editing the schematic, save the project as follows:

"File" -> "Save"
Section 2  Simulation using PSpice

Subsection 1  Setting up analyses

To set up analyses, choose the following in Capture.

"PSpice" -> "New Simulation Profile"

In the displayed dialog box, type "Tran" in the Name text box, choose "none" from the "Inherit From" list and press "Create". In the Simulation Settings dialog box, specify as follows:

1. Analysis type: Select a type of analysis. Now, select the following in order to compare with the results from Simulink:
   "Time Domain (Transient)"

2. Run to time: Specify the stop time for analysis. Type the following:
   "3u" (3µsec)

Press "OK".
Subsection 2  Running a simulation

Select the following command from the Capture menu:

"PSpice" -> "Run"

PSpice A/D will open and start a simulation. When the simulation is completed without any problem, an empty plot window appears.

Subsection 3  Verifying results

Place a marker for displaying a waveform on the schematic. From the Capture menu, select the following:

"PSpice" -> "Markers" -> "Voltage Level"

Place a marker on the MOSFET drain node.

The drain voltage waveform is displayed on the PSpice window.
Section 3  Creating and setting up a block diagram

Subsection 1  Starting MATLAB

Start MATLAB from the Start menu.
"Start" -> "All Programs" -> "MATLAB..." -> "MATLAB..."

Set the current directory of MATLAB to the work directory where PSpice files are saved.
```
cd c:/work
```

Subsection 2  Creating a block diagram

Create a new model from the MATLAB command window.
"File" -> "New" -> "Model"

Call the SLPS library from the MATLAB command window.
```
slpslib
```

Create a block diagram like the following using an SLPS block.

Blocks used:
- Repeating Sequence /Sources
- Scope /Sinks
Repeating sequence parameters

When the model is finished, save it.

"File" -> "Save"

Type "mosckt.mdl" as the name of the file, and save it under the MATLAB current directory.

* Note: If the model is not saved in the current directory, you cannot set up the following SLPS block parameters.

Subsection 3 Setting up SLPS block parameters

Open the setting window by double-clicking on the SLPS block.

Designating Capture project file

At "Capture Project file:", designate the created project file "MOSCKT.opj".

When a project is designated, cir files included in the project (created for each PSpice analysis setup) are listed as "PSpice Circuit File", so designate the file to be used. Here, only tran.cir is listed, so it is automatically selected.
Designating input/output

Designate input and output to the SLPS block. 
Press the "Select" button to the right of "Input Sources:".

![Image of input source selection]

Because all power sources in a circuit are displayed, click “V1” as the input power source and “Close”.
Now, you can see that V1 is in the "Input Sources:" list.

![Image of input source selection]

To designate output, press the "Select" button to the right of "Output:" and select "V (OUT) " from the list.

![Image of output source selection]

This completes setting.
Close the SLPS Settings window by pressing the "OK" button at the bottom of the window.
Subsection 4 Setting up Simulink analyses

To set up Simulink analyses, select the following from the model window menu:
"Simulation" -> "Simulation parameters"

(1) Simulation time: Stop time: Set the analysis stop time to the same stop time as PSpice.
   \[3e-6 \text{ (3 µsec)}\]

(2) Solver options: Use fixed step.
   Type: Fixed step, Discrete

(3) Fixed step size: For the step size, designate a sufficiently small value, about 1/1000th of the
   analysis stop time.
   Fixed step size: \[1e-9\]

After you finish setting up, save it.
"File" -> "Save"
Section 4  Simulation and verification

Subsection 1  Simulation using Simulink

Start a simulation for analysis.

"Simulation" -> "Start"

Subsection 2  Verifying results

When analysis is finished, double-click on Scope, display the waveform, and perform Autoscale.

You see that the same waveform appears as in PSpice simulation results.
Chapter 3  Creating Simulation models

Section 1  Preparing PSpice circuits

Subsection 1  Creating a CIR file

A CIR file (extension: cir) contains information of the PSpice analysis settings and the net list to be used. The file is used to assign PSpice circuits to a SLPS block.

A CIR file is created when performing a PSpice simulation in PSpice A/D. To assign a circuit to SLPS, you need to analyze with PSpice A/D. In this case, analysis is done only to create a CIR file, so set a short analysis time to finish a calculation quickly. No need to set the analysis time for simulation which is set in Simulink. When a change is added to a circuit, always perform analysis again using PSpice so that the changes are reflected in the CIR file.

When the schematic is finished, perform setting for transient analysis in Capture.

* If transient analysis is already finished, there is no need to do this work.

"PSpice" -> "New Simulation Profile"
Designate analysis settings as follows:

Analysis type: **Time Domain (Transient)**

Run to time: An arbitrary short time (This doesn't have to be the analysis stop time designated with Simulink.)

Click "OK" and run a simulation once.

"PSpice" -> "Run"

When PSpice starts up and an analysis is finished, a CIR file is created. Then close the PSpice window.

---

**Section 2 ** **SLPS block**

The SLPS block library is called from the MATLAB command window using the following command:

```
slpslib
```

SLPS can be used by placing an SLPS block in the Simulink model.
Subsection 1  SLPS settings window

By double-clicking on the SLPS block, the Settings window appears, as shown in the following.
*To open the SLPS Settings window, a Simulink model in which an SLPS block is inserted must be saved, and the MATLAB current directory must be set to the directory where the Simulink model is saved, or these folders must be set in the MATLAB path.

(1) Capture Project file
Designate a Capture project file (*.opj) containing a PSpice schematic to be assigned.

(2) Open Capture button
Open the designated project in Capture.
If a project has not been designated, Capture will start without project.
When pressing this button, you cannot open a designated project if Capture has already started.
Press this button after closing Capture that is open, or manually open the designated project from Capture.

(3) Reload/Clear All button
If a schematic has been changed, you can update information by pressing the Reload button.
To clear the items on the SLPS Settings window, press the "Clear All" button.

(4) PSpice Circuit file
All CIR files contained in the project selected in (1) will be listed, so select the CIR file you wish to use.
(5) **Message area**  
Errors, status and other messages are displayed here during setting.

(6) **Input Sources:**  
Designate the voltage source (V*) and current source (I*) for supplying input data (from Simulink to the SLPS block) into the circuit. If a voltage source is selected, the input data will be supplied to the circuit as a voltage value, and if a current source is selected, it will be supplied as a current value. The sequence listed here is the sequence of input signals to the SLPS block. At least one input source must be specified.

(7) **Input Select button**  
When this button is pressed, all power sources contained in the circuit referenced by the CIR file selected in (3) are listed. Click on the power source you wish to use, and add to the list in (4).

(8) **Input add/move/delete button**  
Use this to add, change sequence or delete items from the list in (4).

(9) **Outputs:**  
Designate data in the circuit to be output from the SLPS block to Simulink. The sequence listed here is the sequence of output signals to Simulink.

(10) **Outputs Select button**  
When this button is pressed, all output variables in the CIR file(s) selected in (3) are listed. Click on the variables to be used and output to the list in (7).

(11) **Outputs add/move/delete button**  
Use this to add and change sequence or delete items from the list in (7).

(12) **Data Saving Option:**  
This option designates how data in PSpice is saved during simulation. Data is saved in the PSpice data file (*.dat).  
- **ALL**  
  All data is saved. In this case, it will take long to analyze to save the data, and the size of the data file will be big.
- **Selection Only**  
  Only the data designated in the output list in (7) is saved.
- **None**  
  Data is not saved at each analysis step, speeding up analysis.

(13) **ITL4 Max:**  
This is the upper limit of the ITL4 option parameter which is automatically increased when a convergence error occurs in PSpice. ITL4 is a parameter which sets an upper limit of the number of repeated calculations per step when PSpice is performing transient analysis, and by increasing this value, you can increase the maximum number of calculations until you obtain a result. Increasing this value has no effect on the precision of the result.

If a convergence error occurs with SLPS, analysis is automatically executed again after increasing the ITL4 value by 10 times. If the value increased by 10 times exceeds the value specified with...
ITL4 Max, analysis uses the ITL4 Max value, and if a convergence error still occurs, analysis will be aborted.

If a convergence error often occurs, set this value to a large value such as 1000 in order to find a result. However, if you designate a very large value, the application may not respond for a long time. Therefore, set it to the minimum value required.

### Specifying SLPS output

Values which can be set as SLPS output are: node voltage, current passing through a device and power dissipation of a device. The specification format conforms to that of PSpice.

- **Node voltage**
  
  Syntax: $V([\text{NODENAME}])$
  
  NODENAME: Node name in circuit

- **Current passing through a device**
  
  Syntax: $I([\text{DEVICENAME}])$
  
  DEVICENAME: Reference name of device

  The sign of current passing through a device conforms to PSpice. The direction from Pin 1 to Pin 2 of the device is defined as positive.

- **Pin inflow current**
  
  Syntax: $I[\text{PINNAME}](\text{[DEVICENAME]})$
  
  PINNAME: Pin name of element

  For pin inflow current, the direction flowing into the pin is regarded as positive.

- **Power dissipation of device**
  
  Syntax: $W([\text{DEVICENAME}])$
  
  Power dissipation is calculated as $I^*V$, but its absolute value is taken, so current directions do not matter.
Section 3  Simulink simulation parameters

Set up Simulink simulation parameters using the Simulink model window.

"Simulation" -> "Simulation parameters"

SLPS can use the all solver options: fixed step, variable step and all.
Adjusting maximum step size

Because SLPS data exchange only operates with each Simulink step, in order to avoid overlooking phenomena from the PSpice circuit, Simulink’s maximum step size (Simulink’s fixed step size when fixed step is selected) must be a sufficiently small value. The value, however, should not be smaller than is needed, or Simulink’s overall analysis may become slow.

As you can see from the diagram, you cannot obtain the correct waveform if the maximum step size is set to a large value.

If you want to see a waveform within PSpice in order to determine the step size, you can check it by starting up PSpice A/D (described later) and opening the PSpice data file, though it will not be displayed in Simulink.
Chapter 4  Simulation

Section 1  Starting a simulation
To analyze a Simulink model containing an SLPS block, select the following command from the Simulink model menu, just as is done when analyzing an ordinary Simulink model.

"Simulation" -> "Start"

Section 2  How to set up window in the event of a convergence error
If a convergence error occurs within PSpice during analysis with Simulink, increase the option parameter ITL4 of PSpice up to the value specified in ITL4 Max in SLPS settings, and retry analysis. However, if PSpice cannot get a result, the following window will be displayed:

Abort
Aborts analysis.

Interact
Recalculates by changing PSpice analysis options.

If you choose Interact, the following PSpice Option Parameter Setting window will be displayed:
After you change these parameters, press the [OK] button. PSpice will continue with the calculation using the changed analysis options. The meanings of each parameter are indicated below:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITL1</td>
<td>Maximum number of iterations of convergence calculation during bias point analysis</td>
</tr>
<tr>
<td>ITL2</td>
<td>Maximum number of iterations of convergence calculation during DC analysis</td>
</tr>
<tr>
<td>ITL4</td>
<td>Maximum number of iterations of convergence calculation at each time step during transient analysis</td>
</tr>
<tr>
<td>VNTOL</td>
<td>Absolute precision of current</td>
</tr>
<tr>
<td>ABSTOL</td>
<td>Absolute precision of voltage</td>
</tr>
<tr>
<td>RELTOL</td>
<td>Relative precision of voltage and current</td>
</tr>
</tbody>
</table>

Normally, if a convergence error occurs during analysis, set the value of ITL4 to a large value such as 1000. Then you can improve convergence without changing analysis accuracy.

For details on each parameter, see the PSpice manual.

---

Section 3 Verifying results

Check the output of SLPS using the scope in Simulink.

Subsection 1 Output files in PSpice

When analysis is performed by the PSpice engine which has been called from SLPS, a PSpice data file (*.dat) and output file (*.out) will be created, just as in simulation with PSpice A/D. However, if None is selected as the Data Saving Option in the SLPS Settings window, an empty file will be created for a PSpice data file.

A PSpice data file stores calculation data within SLPS, either for the entire circuit or for just the part selected as SLPS output.

Information like the analysis log within PSpice is output in text format to the PSpice output file, so this is useful for verifying PSpice internal errors that may occur during analysis.

Each file with the following name is created in the same directory as the designated CIR file.

- **PSpice data file**: `SLPS_[CIR file name].dat`
- **PSpice output file**: `SLPS_[CIR file name].out`
Displaying SLPS internal data using PSpice A/D

You can verify the results obtained in SLPS by opening the PSpice data file in PSpice A/D. Start PSpice A/D and select the following:

"Start" -> "Program" -> "OrCAD 10.0" -> "PSpiceAD"

And open the PSpice data file as follows:

"File" -> "Open"

Select a data file created with SLPS. For information on how to display waveforms with PSpice, please see the PSpice User's Guide.

Waveforms are displayed by opening SLPS data file in PSpice A/D.
Chapter 5  Examples

Section 1  RC circuit (RCCIR)

Use a simple RC circuit indicated below as an electrical circuit. (Capture project file: RCCIR.opj)

The Simulink block diagram is shown below (rcmdl.m).
SLPS block parameters are set as shown in the following diagram:

- Capture Project file: **RCCIR.opj**
- PSpice Circuit file: **tran.cir**
- Input Sources: **Vin**
  Input data from Simulink as a voltage value, using the voltage source Vin in the circuit.
- Outputs: **V (1) / V (2) / I (R)**
  For Simulink, the voltage values at nodes 1 and 2 in the circuit, and the current value passing through the resistor R are output.
- Data Saving Option: **None**
  In order to speed up analysis, select None not to save analyzed data in SLPS.
- ITL4 Max: **10**
  Convergence errors rarely occur because this is a linear circuit, so set it to the default value of 10 times.

In the Simulink analysis settings, analysis is set up with a step size of 0.1 seconds, from 0 to 10 seconds. The following results will be obtained by manually executing rcp lt.m from the MATLAB command window, or executing analysis by selecting the START block in the block diagram.
Section 2  Nonlinear load (NLLOAD)

Simulate the state where the values of devices in the circuit are controlled by Simulink and the load connected to the circuit is nonlinear.

Subsection 1  Schematic

The electric circuit is roughly divided into two parts: an amplifier and a load connected to the output of the amplifier. The load resistance is determined by Simulink.

The voltage control impedance $Z_X/\text{anl\_misc.olb}$ is used as the load resistance of the circuit. The output impedance of this device is determined by the following formula:

$$Z_{\text{out}} = Z_{\text{ref}} \times V$$

- $Z_{\text{out}}$: Impedance between output pins
- $Z_{\text{ref}}$: Reference impedance (connected to Reference pin)
- $V$: Control voltage

In the circuit, a VSLPS voltage source (which adds a 1nV offset) is connected between the $Z_X$ control voltage pins, and a 1Ω reference resistor is inserted between the Reference pin and ground. A 1pΩ load impedance offset resistor is connected between NODE2 and ground, and NODE1 is connected to the output of the amplifier circuit. Therefore, the load impedance in respect to the amplifier circuit is given by the following formula.

$$Z_{\text{load}} = 1\Omega \times (\text{VSLPS}+1\text{nV}) + 1\text{p}\Omega$$
Subsection 2  Simulink model

The Simulink model is shown below. (nlload.m)

The voltage values of NODE1 and NODE2 are taken as output from SLPS. The load resistance characteristic is determined by the hierarchical block "Load Equation", and the resistance value is returned to SLPS. The resistance value is determined as follows:

\[ L_{\text{Out}} = 2000 \times e^{\text{abs}(V(\text{NODE1}) - V(\text{NODE2}))} \]

The resistance value \( L_{\text{Out}} \) is an exponential function of the voltage differential between NODE1 and NODE2.

Please note that in this model, when the SLPS block is inserted in the feedback loop, a delay of one cycle step of Simulink will occur between the feedback loop's input (SLPS block's input) and its output (SLPS block's output). Therefore, to obtain high-precision results, you must force Simulink to take small time steps.

Subsection 3  Results

Left: Output voltage (\( V(\text{NODE1}) \)) waveform
Lower left: Time axis waveform of resistance value
Right: Output voltage vs. Resistance characteristic

You can confirm that the load value changes with time, and the output voltage waveform is distorted due to that reason.

This example shows a nonlinear load, but it can be applied to load fluctuation of a motor or counter EMF.
Section 3  PLL model (PLL)

PLL (Phase Locked Loop) circuit can halve transmitter output that is synchronized with the input signal and therefore expands usability to digital circuit blocks, mobile communications, etc.

Subsection 1  PLL composition

The PLL is comprised of the following three modules:

1. **Phase comparator**

This detects the phase difference between two signals, and outputs that difference. Here it detects the phase difference between the input signal to the PLL, and the feedback PLL output signal.

The PLL input/output waveform to be designed here is assumed to be a pulse waveform, so it is given by a simple adder, as shown in the diagram. In reality, the output value offset is adjusted by the characteristics of the loop filter and VCO being used.

2. **Loop filter**

This eliminates AC components from the phase comparator output and creates a VCO control signal for the next stage.

A loop filter is an LPF (Low Pass Filter), and a transmission function block is used in Simulink. Here, the frequency characteristic (when fc=10Hz) can be checked using the freqs function of MATLAB.

Examples
3. **VCO**

A VCO (Voltage Controlled Oscillator) is an oscillating circuit where oscillation frequency varies with control voltage.

In the end, the output of this VCO becomes the output of the PLL, so the VCO control voltage is controlled so that the oscillation frequency is synchronized with the PLL input signal.

The VCO electrical circuit described in Capture is shown at right. (Capture project file: VCO.opj)

The simulation results using PSpice are indicated below.

You can see that the oscillation frequency is about 1.8 KHz at an input voltage of 1V, and it increases almost linearly along with an increase in the input voltage.
Subsection 2  PLL model incorporating VCO electric circuit

The Simulink PLL model is shown below.

The simulation results are shown as follows.

From the VCO control signal, it can be confirmed that the oscillation frequency is locked after 70ms. In this section, an electrical circuit is incorporated by SLPS in VCO only, and the entire PLL operation is checked. If an electrical circuit is incorporated in a loop filter, phase comparator, etc, delays of elements used in the circuit and the effects of the circuit’s time constant can also be confirmed.
Section 4  Switched reluctance motor control

When designing a motor control system, you need to understand the characteristics of the entire system, such as torque determined by motor dimension, the electric circuit including motor inductance and switching device, and position control associated with rotor rotation.

Subsection 1  Analyzing magnetic field by analysis tool using finite element method

With ANSYS, a general-purpose finite element analysis program you analyze electromagnetic field of the motor form and obtain the inductance value to model a motor.

The switched reluctance motor (SRM) used here as an analysis example employs a 3-phase excitation motor, with 6 stator poles and 4 rotor poles, as shown in the diagram below.

![SRM model](image)

The SRM can obtain torque using only the magnetic force between rotors and stators. Therefore, if let $L$ be the excitation coil inductance, $i$ the current, and $W$ the magnetic energy, the torque $T$ can be expressed with the following formula:

$$T = \frac{dW}{d\theta} = \frac{1}{2} i^2 \frac{dL(\theta)}{d\theta}$$

Therefore, as you can see, torque is derived from changes in inductance. The inductance of the excitation coil to drive the motor varies with the rotation angle of the rotor. You can obtain the inductance value for each angle by performing magnetic field analysis of the motor.
Subsection 2  Modeling of electric circuit using PSpice

In PSpice, the inverter circuit is modeled as the load of variable inductance. The inductor model placed in the circuit indicates motor inductance components. The inductor value can be determined externally using modeling techniques that separate electrical characteristics and magnetic circuits. During analysis, analysis results from ANSYS are referenced by Simulink, and the inductance value determined by the rotor position is passed as a parameter. The gate signal of the switching transistor is also provided from Simulink, the current value passing through the inductance is monitored, and its value is returned to the outside.
Subsection 3  Simulink model

The entire motor control model created with Simulink is shown below.

The results are shown as follows.

Inductance values for each phase  
Torque
The number of rotations varies with load torque, motor dimensions, switching device and switching waveform, etc. These effects can also be simulated.

<Reference>
Please refer to the following document for the details of this model.
"A new Calculation Model of Switched Reluctance Motor for use on SPICE",
Section 5  Switching power supply

Power electronics circuits, such as switching power supplies and motor control, may have both electric circuits which switch a large current and high voltage in a short time using a power device, and control circuits with ICs. If you attempt to model an entire circuit of this type using an electric circuit simulator like SPICE, you must express all of the control parts with the electric device level. In this case, the entire circuit becomes complex, so you may have to spend a lot of time not for verification but for other tasks at the initial stage of design. As the number of devices increases, convergence errors are likely to occur, by which it may take longer to analyze data or analysis may stop before it is complete, in the worst-case scenario.

On the other hand, if the entire circuit is simulated with the system simulator MATLAB/Simulink, it is easier to express the control algorithm in numerical formulas, but it is difficult to accurately express the rise/fall characteristics of the switching device, and the electrical characteristics of inductors and transformers because it does not have a library of devices.

Subsection 1  Electric circuit design using PSpice

The following shows the circuit diagram with the forward converter inserted:

![Forward converter circuit diagram](image)

The conditions for this circuit are as follows:
- Rated input voltage 130V
- Output voltage 10V
- Output current 10A

Here, find the optimal voltage of the snubber circuit's capacitor connected to Q1 in order to minimize power loss due to switching.
The following shows the VCE waveform of Q1 when the value of C2 is varied from 0.1n to 1.0n using the PSpice function of parametric analysis.

From these results, you can see that the optimal value is approximately 0.5nF where power loss is limited and VCE takes the smallest value during switching.

When 0.5nF is applied to C2, the following output current, voltage and VCE waveforms appear:
To specify load from Simulink, use a variable impedance $ZX$ instead of the load resistor $R_1$, thereby making it possible to determine the resistance value using $V_{load}$.

![Circuit diagram with variable load]

**Subsection 2  Control system design using Simulink**

Using MATLAB/Simulink, design a PWM controller by placing an SLPS block as shown below.

![Switching Power Supply]

Assign the previous circuit diagram to the SLPS block, designate the output voltage $V_{out}$ and output current $I_{out}$, and the power dissipation of the switched transistor $Tr$ Power Dissipation. In the PWL control model, the PI controlled output voltage is compared with the reference value $Reference$, and then with a triangle wave, to create a PWM signal.
In the PWL control model, the PI controlled output voltage is compared with the reference value, Reference, and then with a triangle wave to create a PWM signal.

**Subsection 3 Results**

Set up analyses in Simulink as follows:

- Stop time: $1 \times 10^{-3}$ secs
- Solver type: Fixed step
- Solver: ode5
- Step size: $1 \times 10^{-6}$ secs

When the load resistance is set to $1\Omega$, get the following simulation results:

Top: Output voltage Bottom: Output current
From the above results, you can see that output stabilizes at about approx. 0.4mS after starting the simulation, and it changes along with the fluctuations in load.

In this design example, PSpice has been used for electrical circuits, and the Simulink design environment for control. Electrical circuits have been used for the parts where electrical characteristics (like element delay) have an effect and those which are difficult to express in formulas. To design the basic circuit, PSpice has been used. To design the control system which drives the circuit, a Simulink model which is flexible in expressing numerical formulas has been used. Finally, both models have been combined using SLPS, with which the entire system has been simulated while taking electrical characteristics into consideration.
Section 6  DC Motor Control System

Due to the tradeoff between simulation speed and result accuracy, it is not an efficient method to use highly accurate models at the first stage of designing, or the simulation speed will be very slow. It is recommended to use proper models for each stage of designing process. You may find that proper modeling makes optimizing the system parameters easier.

Subsection 1  Outline of DC motor control system

Below is the diagram of a DC motor control system.

This diagram consists of a stator or field winding which produces a constant magnetic flux for excitation. This winding is modeled by inductance $L_s$ and resistance $R_s$. When voltage $V_s$ is applied to the stator winding, current $I_s$ flows which produces flux $\psi$. The rotor winding is also modeled by its inductance $L_R$ and resistance $R_R$. When voltage $V_R$ is applied to the rotor winding, current $I_R$ flows. $V_R$ is the output of a controller which is driven by a differential input voltage $V_D$. The signal flow is from $V_D$ to $V_R$ but not in the reverse direction. Thus, the armature reaction between rotor and stator is neglected.

Stator

First, we write the mesh equation for the stator winding

$$V_s = R_s I_s + \frac{d\psi}{dt}$$

In general, the magnetic flux $\psi$ is a nonlinear function of the stator current $I_s$. For simplicity, we assume $\psi$ to be a linear function of $I_s$

$$\psi = L_s I_s$$
And we get

\[ V_S = R_S I_S + L_S \frac{dI_S}{dt} \quad \text{or} \quad \frac{dI_S}{dt} = \frac{V_S - R_S I_S}{L_S} \]

**Rotor**

For the mesh equation of the rotor winding, we get

\[ V_R = R_R I_R + L_R \frac{dI_R}{dt} + V_I \quad \text{or} \quad \frac{dI_R}{dt} = \frac{V_R - R_R I_R - V_I}{L_R} \]

\( V_I \) is the opposing e.m.f. which is induced in the rotor winding when it rotates in the excitation field. \( V_I \) can be written as the product of a constant \( C_i \), flux \( \Phi \) and angular velocity \( \omega \).

\[ V_I = C_i \Phi \omega \]

**Mechanical dynamics**

For the moments on the rotor shaft, we can write the balance equation:

\[ J_M \frac{d\omega}{dt} = M_R - M_L - M_F \]

With

\[ M_R = C_2 \Phi I_R \quad M_F = C_3 \omega \]

\( M_R \) is the rotor driving torque which is a linear function of excitation flux \( \Phi \) and rotor current \( I_R \) multiplied by constant \( C_2 \). \( M_L \) is the given load moment and \( M_F \) is the internal friction moment of the motor which is assumed to be a linear function of angular velocity \( \omega \) multiplied by constant \( C_3 \). \( J_M \) is the moment of inertia of all rotating masses mechanically connected to the rotor.

**Controller**

The controller is driven by the voltage difference \( V_D \) between the speed potentiometer voltage which sets the nominal angular velocity \( \omega_S \) and the tachometer voltage which is proportional to angular velocity \( \omega \). We can write \( V_D \) as the difference between the angular velocities multiplied by constant \( C_4 \).

\[ V_D = C_4 (\omega_S - \omega) \]

For the controller, we use a PI characteristic which is described by:

\[ V_R = C_5 \int V_D dt + C_6 V_D \]
Subsection 2  Simulink Model

First, we model the whole system in Simulink.

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>0.50</td>
</tr>
<tr>
<td>C3</td>
<td>0.10; % N<em>m</em>s</td>
</tr>
<tr>
<td>C5</td>
<td>606.0; % 1/s</td>
</tr>
<tr>
<td>C6</td>
<td>100.0</td>
</tr>
<tr>
<td>LR</td>
<td>0.10; % H = V*s/A</td>
</tr>
<tr>
<td>RS</td>
<td>500.0; % Ohm = V/A</td>
</tr>
<tr>
<td>RR</td>
<td>0.50; % Ohm = V/A</td>
</tr>
<tr>
<td>JM</td>
<td>1.0; % kg*m^2</td>
</tr>
<tr>
<td>C2</td>
<td>0.50</td>
</tr>
<tr>
<td>C4</td>
<td>0.01; % V*s</td>
</tr>
<tr>
<td>LS</td>
<td>10.0; % H = V*s/A</td>
</tr>
</tbody>
</table>

**Simulink Model Result**

The plots for the angular velocities $\omega_S(t)$ and $\omega_R(t)$ are shown below.

The stator voltage $V_S$ is switched on at time $t=1s$. $\omega_S$ is set to 50 rad/s at 2s and the rotor starts to turn. At about $t=6s$, $\omega_R$ reaches the steady state of 50 rad/s.

At $t=7s$, a load moment of 50 Nm is applied and the rotor speed reduces to 43.6 rad/s. Therefore, the controller increases $V_R$ and the new steady state of $\omega_S=50$ rad/s is reached again at about $t=10s$. 

Examples 5-19
Subsection 3  Simulink-PSpice Model

(Ideal OPamp)

With the results of the Simulink model available for reference, we will now replace some of the Simulink blocks to PSpice circuits. Below is the system block diagram.

System block diagram

Simulink Model (Ideal OPamp)

Below is the Simulink model which includes PSpice circuits using SLPS.

Simulink model
PSpice Circuit (Ideal OPamp)

Stator, rotor, and PI controller are modeled with PSpice as seen below.

Circuit elements $V_s$, $R_s$, $L_s$ and $E_r$, $R_r$, $L_r$ implement the equation for $dl_s/dt$ and $dl_r/dt$. For the operational amplifier (OPamp) in the PI controller, we first use an ideal model with very high open-loop gain, infinite input resistance, zero output resistance, zero input offset voltage, zero input offset and bias current, and no output voltage saturation. This behavior is easily modeled in SPICE by a linear voltage-controlled voltage source. The PI controller output voltage $V(9)(\text{PIC\_OUT})$ is given by the equation:

$$ V(9) = -\left( \frac{1}{R_C} \int V_D dt + \frac{R_2}{R_1} V_D \right) $$

With the factors

$$ \frac{1}{R_C} = \frac{1}{10k\Omega \cdot 3.3\mu F} = 30.3 \quad \frac{R_2}{R_1} = \frac{50k\Omega}{10k\Omega} = 5 $$

The corner frequency $f_0$ where the I- and P- characteristics meet is

$$ f_0 = \frac{1}{2\pi R_2 C} = \frac{1}{2\pi \cdot 50k\Omega \cdot 3.3\mu F} = 0.965\,\text{Hz} $$

Examples
Voltage $V(9)$ controls the voltage-controlled voltage source $E_R$ which has a gain value of -20. Therefore, we get

$$V(3) = V_R = -20 \cdot V(9) = 606 \int V_D dt + 100V_D$$

Comparing this equation to the one for $V_R$ in the pure Simulink model, we note that both are identical because of constant values $C_5=606$ and $C_6=100$. The PSpice controller circuit therefore implements the same equation $V_R$ for as in subsystem PI-CONTROLLER in Simulink model.

**Simulink-PSpice Model Result (Ideal OPamp)**

The simulation results are below

- $\Gamma(t)$ and $\Gamma_s(t)$ for PI controller with ideal OPamp
Subsection 4  Simulink-PSpice Model

(Device OPamp)

PSpice Circuit (Device OPamp)

We now replace the ideal OPamp model in the PI controller with a device-level model of a standard A741 OPamp.

Simulink-PSpice Result (Device OPamp)

After the excitation is turned on at 1s, the motor starts to turn in the opposite direction. Obviously, the device-level model behaves differently from the ideal OPamp model used in the controller. This is because of that the OPamp output goes into saturation because of the input offset voltage and the very high open-loop gain at DC operating point calculation at \( t=0 \).
PSpice Circuit (DC feedback)

To avoid the offset problem, a 10MΩ resistor was put in the feedback loop to limit the DC gain to 1000.

Simulink-PSpice Model Result (DC feedback)

Because the OPamp output offset voltage is reduced, the initial transient in the opposite direction is also reduced. A negative effect of the reduced DC gain is that the steady-state response of \( \dot{\theta}(t) \) slightly differs from 50 rad/s.
Chapter 6  FAQ

Can multiple SLPS blocks be used?

Only one SLPS block can be placed in a single Simulink model.
If you want to incorporate multiple circuits, use Capture to create multiple circuit diagram pages within a
project, and connect all the data lines in the Simulink model (which need to be connected to the circuit) to
a single SLPS block.

I cannot open the SLPS Setting window.

If the following error dialog appears by double-clicking on the SLPS block to open the setting window,
the Simulink model with a SLPS block has not been saved or the
MATLAB current directory is
different from the directory where the
Simulink model was saved.
→SLPS setting window

A warning appears when opening slpslib in MATLAB

R14

If the SLPS block library is displayed at the slpslib command of the MATLAB R14 command window,
there is no problem with using SLPS, although the following warning message will be displayed.

Warning: Function call slpslib invokes inexact match…….

CIR files and input/output are not listed in the SLPS

Setting window.

If CIR files are not listed after you designate Capture project, the designated project may not be a PSpice
project, or analysis settings (simulation profile) may not have been created.
If input/output is not listed, transient analysis using PSpice A/D may not have been performed from
Capture. Be sure to perform transient analysis beforehand. →Creating a CIR file
I pressed the Open Capture button, but schematic diagram does not open.

If Capture is open when you start Capture by pressing Open Capture from SLPS Setting, you cannot open a project. →SLPS setting window

Changes in schematic diagram are not reflected in SLPS.

If a schematic diagram or settings are changed with Capture, you must execute analysis once using PSpice A/D in order for the changes to be reflected in SLPS. →Creating a CIR file

A license error occurs when I try to start analysis.

If the following license error window appears, the PSpice or SLPS license was unable to be acquired. Please check if your machine/system has a proper environment for the PSpiceA/D, Unison or SLPS license, or start analysis by activating PSpiceA/D to check the situation. →System requirements

I want to use a netlist created with something other than OrCAD Capture.

SLPS uses some files created with OrCAD Capture, so you cannot directly assign net lists created with applications other than Capture, but you can use Capture to convert a SPICE net list to a symbol and incorporate it into a circuit. After inserting the symbolized net list into a circuit, perform transient analysis with PSpice A/D, and then assign it to SLPS.

Analysis speed of Simulink models containing SLPS is slow.

When an SLPS block is inserted, it may take longer to analyze the model, though calculation speed is fast enough without it. Because the PSpice engine solves all electric circuits using nonlinear operations,
Simulink waits for a response from SLPS at each time step, which affects analysis speed of the entire model. If analysis speed is too slow, please check the following points:

**PSpice internal data saving option**

If one of the items other than None is selected at the Data Saving Option in the SLPS Setting window, the PSpice engine will perform tasks to save results during analysis, so the entire speed will be down. → SLPS setting window

**Input waveform to SLPS**

The analysis speed of PSpice is affected greatly by the amount of change per unit time in the input waveform. If there is a large change in value in a short time, PSpice will attempt to get a result by using smaller internal time steps, so analysis speed will be slower. Enter a gradually-changing waveform to SLPS. You may insert a low-pass filter at the previous stage to do this.

**Effects of nonlinear devices used in PSpice circuits**

In addition to the input waveform, nonlinear elements in the circuit may affect analysis speed. This phenomenon occurs if analysis is executed using PSpice A/D in Capture. Before passing a circuit to SLPS, replace the model with a model which converges to a solution or take other countermeasures.

**Effects of Simulink time step**

The parameter relating to the internal time step of PSpice is based on a value designated in Simulink analysis settings. The maximum step size of Simulink should be adjusted to their optimal values, but it may be possible to obtain better results by setting it to Auto. → How to determine PSpice parameters

**Convergence errors occur frequently.**

To avoid convergence errors, refer to the above section. In almost all cases, the same methods as how to speed up calculations in SLPS apply.

**The direction of current output is opposite.**

In the current sign in PSpice, the direction from pin 1 to pin 2 is defined as positive if you use a 2-terminal device. Check the direction of the device. → Designating SLPS output
A value is not obtained when a digital node voltage is designated.

A digital node value cannot be taken in the current version of SLPS. You need to change the digital node to an analog node by creating a path from the node to ground through a large resistance, or inserting a resistance with a small value into the node.

I want to assign a value from SLPS to a global parameter in the circuit.

This function is not available in the current version of SLPS. The initial values given in the circuit diagram are assigned to global parameter values.

If there is no input for SLPS to be given to a circuit...

At least one input must be designated for SLPS, but you may want to assign a circuit with only output to SLPS (i.e. an oscillator). In this case, create a dummy circuit (for example, with a power source and resistance) within the circuit diagram page, do not connect it with the original circuit, and allocate some data to this power source.

The amount of changes in SLPS outputs does not effect to the step size of Simulink when variable step is used.

Although a variable step can be used with Simulink in the current version of SLPS, the Simulink step cannot be controlled by the amount of changes in SLPS. Please adjust the maximum step size to avoid overlooking output waveform phenomena of SLPS. → Adjusting maximum step size
Can I use real-time workshop (RTW) ?

SLPS is an interface with PSpice. The PSpice engine has not been incorporated into Simulink models, so you cannot use the real time workshop.

Note concerning models including feedback in SLPS

When the SLPS block is inserted in the feedback loop, calculation in the SLPS block is performed by PSpice, which operates in a different memory space than that of Simulink. In principle, a delay of one Simulink cycle step will occur between the SLPS block’s input and its output. To minimize this effect, Simulink’s time steps must be sufficiently small.

Simulation stops when activating Hyper-Threading (HT) function

SLPS may stop a simulation when activating the Hyper-Threading (HT) function on a PC which has a CPU supporting HT. SLPS does not support the HT function, so you need to disable the HT function when running SLPS.
Chapter 7 Detailed information

Section 1 Data exchange between Simulink and PSpice

When a SLPS is placed in a Simulink model, two different analysis engines are used for simulation, where transient analysis will be done at their respective time step. Data exchange between Simulink and PSpice is done at the time steps of Simulink, which manages the entire model.

In Simulink, you cannot check the internal phenomena of PSpice which has smaller time step than that of Simulink, as you can see from the above. To check the internal results of PSpice, designate one of the items other than None at the Data Saving Option in the SLPS Setting window, and use PSpice A/D to open the PSpice data file SLPS_***.dat after completing an analysis.
Section 2  How to determine PSpice transient analysis parameters

In general, the internal time steps of PSpice are much smaller than the Simulink time steps. If PSpice uses the minimum/maximum time step determined by Simulink as is, a convergence error will be likely to occur; that is, no result can be found. To solve this problem, SLPS calculates values suited to PSpice based on Simulink parameters, and then passing these to PSpice.

The information explained in this section is useful for understanding how PSpice internal time steps are determined, and taking countermeasures against a convergence error.

PSpice parameter types

PSpice uses the following four internal parameters when performing transient analysis:

PSpice internal step parameters (The name used for explanation in this Chapter is given in parentheses.)

- **TSTOP** (PS_TSTOP) : PSpice stop time
- **Delmin** (PS_Delmin) : Minimum step size
- **Delmax** (PS_Delmax) : Maximum step size
- **Delnew** (PS_Delnew) : Initial step size

In an ordinary PSpice simulation, the only parameter directly designated by the user is TSTOP, and the other parameters are determined as follows:

\[
\text{Del min} = \frac{TSTOP}{(10 \times \text{RELTOL})}
\]

* RELTOL is an analysis option parameter. Its default value is 0.001

\[
\text{Del max} = \frac{TSTOP}{50}
\]

Delnew = 2 x Delmin or Delmax, whichever is smaller
Simulink parameter types

Simulink has the following parameters:

- **STOP (SL_STOP)**: Simulink stop time
- **Initial Step (SL_Initial)**: Initial step size
- **Min Step (SL_Min)**: Minimum step size
- **Max Step (SL_Max)**: Maximum step size

For **STOP**, you can designate infinity as Inf, but in this case, analysis will continue until the user stops simulation.

For other parameters, you can designate auto, so the parameter values are determined automatically.

How to determine PSpice parameters

When PSpice called from SLPS begins analysis, together with Simulink, SLPS determines the transient analysis parameters based on the analysis parameters of Simulink. However, if some Simulink parameters are set to auto, their values must be determined using other parameters. The following gives how to determine PSpice parameters in each case.

**If SL_STOP was infinite**

\[ PS_{\text{TSTOP}} = \text{Maximum real number value (INT–1)} \]

**If all of SL_Initial, SL_Min and SL_Max were auto**

All PSpice internal parameters are based on the values designated in Capture analysis settings.

**If only SL_Initial was designated**

\[
\begin{align*}
PS_{\text{Delmin}} &= \text{Use calculation formula for ordinary PSpice analysis} \\
PS_{\text{Delnew}} &= SL_{\text{Initial}} / 4 \\
PS_{\text{Delmax}} &= SL_{\text{Initial}} \times 100
\end{align*}
\]

**If only SL_Min was designated**

Calculate \( PS_{\text{Delmin}} \) using the calculation formula for ordinary PSpice analysis.

Only when \( PS_{\text{Delmin}} > SL_{\text{Min}} \):

\[
\begin{align*}
PS_{\text{Delmin}} &= SL_{\text{Min}} / 10 \\
PS_{\text{Delnew}} &= 2 \times PS_{\text{Delmin}} \\
PS_{\text{Delmax}} &= SL_{\text{Min}} \times 10000
\end{align*}
\]
If SL Initial and SL Min were designated
   Calculate PS Delmin using the calculation formula for ordinary PSpice analysis.
   Only when PS Delmin>SL Min:
       PS Delmin = SL Min / 10
   PS Delnew = SL Initial / 4
   PS Delmax = SL Initial×100

If only SL Max was designated
   Calculate PS Delmin using the calculation formula for ordinary PSpice analysis.
   PS Delnew = 2×PS Delmin
   PS Delmax = SL Max×1.1

If SL Max and SL Initial were designated
   Calculate PS Delmin using the calculation formula for ordinary PSpice analysis.
   PS Delnew = SL Initial / 4
   PS Delmax = SL Max×1.1

If SL Max and SL Min were designated
   Calculate PS Delmin using the calculation formula for ordinary PSpice analysis.
   PS Delnew = 2×PS Delmin
   Only when PS Delmin>SL Min:
       PS Delmin = SL Min / 10
   PS Delmax = SL Max×1.1

If all parameters were designated
   Calculate PS Delmin using the calculation formula for ordinary PSpice analysis.
   Only when PS Delmin>SL Min:
       PS Delmin = SL Min / 10
   PS Delnew = SL Initial / 4
   PS Delmax = SL Max×1.1