FONTE DE ALIMENTAÇÃO COMUTADA

Objectivo

Utilizando o simulador **PSpice** pretende-se realizar e verificar o funcionamento de uma fonte de alimentação comutada baseada no conversor DC/DC *flyback*. A tensão de entrada não regulada deste conversor deverá ser fornecida através de um conversor AC/DC.

Conhecimentos Práticos Necessários

Este trabalho apoia-se na matéria leccionada na disciplina de Electrónica de Potência. A bibliografia recomendada é a dessa disciplina.

PSpice

É um *software* de simulação da *MicroSim Corporation* que ajuda a análise de circuitos eléctricos e electrónicos. Em anexo é fornecida informação sobre esta ferramenta e respectivas instruções.

Introdução

As fontes de tensão reguladas são necessárias para a maior parte dos sistemas electrónicos analógicos ou digitais. Pretende-se que a maior parte delas tenham os seguintes requisitos:

- Tensão de saída constante (dentro de uma tolerância pré-definida) para variações dentro da gama especificada da tensão de entrada e da carga.
- Isolamento eléctrico entre a entrada e a saída.
- Múltiplas saídas isoladas umas das outras.

Atendendo a estes requisitos, o principal objectivo é reduzir o tamanho da fonte de tensão e melhorar a sua eficiência. O avanço na tecnologia dos semicondutores permitiu-nos usar fontes de tensão comutadas, em oposição às fontes de tensão lineares. Como as fontes comutadas operam a frequências elevadas, os componentes de filtragem são menores, tornam-se menos pesadas e volumosas.

A comparação económica entre fontes lineares e comutadas depende da potência necessária.

Características gerais da fonte comutada

- Tensão de entrada 220V(RMS) a 50Hz
- Tensão de saída contínua e igual a 12V
- Corrente de saída variavel entre 0.5A e 5A.
- *Ripple* da tensão de saída na situação mais desfavorável inferior a 2%.

Características do conversor *flyback*

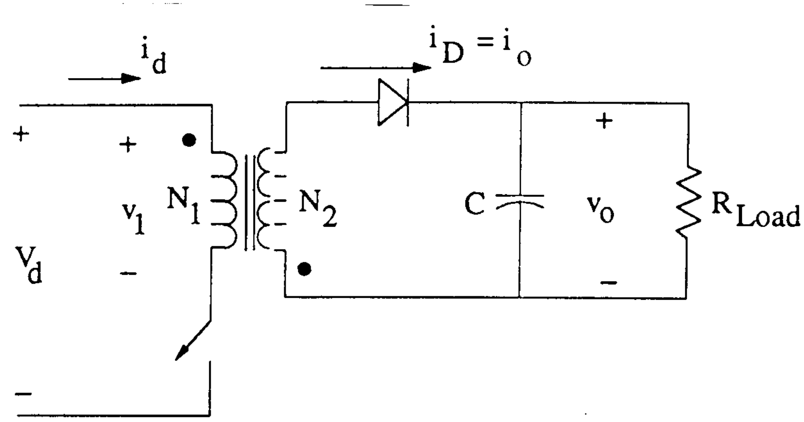
- Razão de transformação, N1/N2 = 1
- Tensão de entrada não regulada variavel entre Vd = 12V e Vd = 24V
- Frequência de comutação, fs = 200KHz
- Indutância de fugas desprezável.

Procedimento do Trabalho

1ª Parte - Conversor AC/DC

Execute o exemplo apresentado no fim do enunciado. Este exemplo representa uma ponte rectificadora monofásica com comutação natural.

2ª Parte - Conversor DC/DC *Flyback*



Dimensione o conversor, tendo em conta as características acima referidas, de modo a que o ferro do transformador nunca se encontre desmagnetizado (o equivalente ao funcionamento no modo de condução contínua).

Com o simulador PSpice, crie o ficheiro **.CIR** relativo a este circuito e execute os passos seguintes:

1º - Obtenha as formas de onda de v_1, i_d, i_D e v_o.

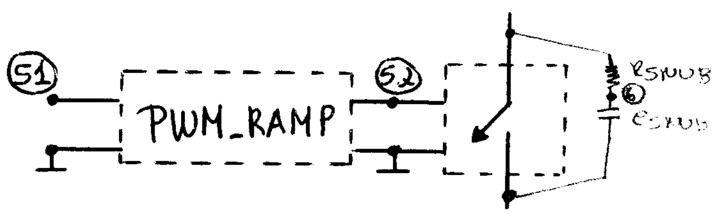
2º - Obtenha as formas de onda da corrente e tensão no *switch* e verifique se os resultados estão de acordo com a seguinte equação: $v_{SW} = V_d/(1-D)$.

3º - Modifique os valores dos componentes de modo ao conversor funcionar no modo de condução descontínuo. Repita a questão nº 1.

NOTA:

Em anexo apresentam-se os modelos dos subcircuitos utilizados neste trabalho e as suas respectivas descrições.

Switch - É modelado através de um subcircuito da livraria chamado *SWITCH* . Neste modelo, um SW AC (no qual a corrente pode fluir em qualquer sentido) está conectado em série com um díodo. Assim a corrente no *SWITCH* é unidireccional. Este SW inclui um circuito de *snubber* RC.



Definição do modelo:
XSW 1 2 52 0 SWITCH

Controlador - Uma onda em dente de serra à frequência de comutação, com um pico de 1V, é comparada com uma tensão de controlo para gerar o sinal comutado de controlo. Isto é feito pelo subcircuito da livraria PWM_RAMP.

Definição dos parâmetros:
.PARAM RISE=4.8us, FALL=0.1us, PW=0.1us, PERIOD=5us

Definição do modelo:
VCONTL 51 0 Duty
XLOGIC 51 0 52 PWM_RAMP

Transformador - É representado pelo subcircuito da livraria TRANS_NONIDEAL. Como a indutância de fugas é desprezável, assume-se que o coeficiente de acoplamento (k) é aproximadamente 1.

Definição dos parâmetros:
.PARAM L1=30uH L2=30us k=0.999

Definição do modelo:
XTRANS 1 2 3 4 TRANS_NONIDEAL

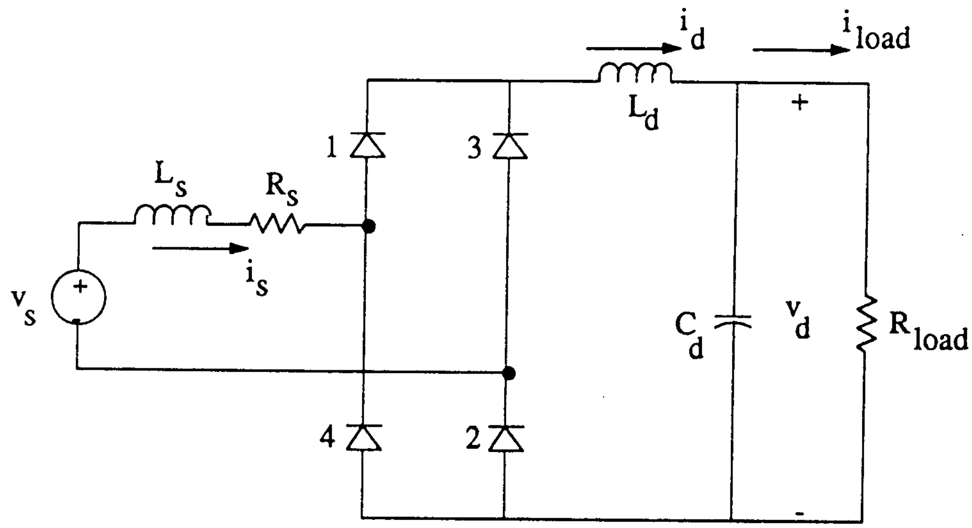
É necessário conectar uma resistência de 1MΩ entre o secundário do transformador e o nó 0 (requisitos de conectividade).

3ª Parte (opcional) - Fonte

Junte ao conversor DC/DC um conversor AC/DC adequado aos requisitos da fonte comutada.

EXEMPLO

Ponte Rectificadora Monofásica não Controlada



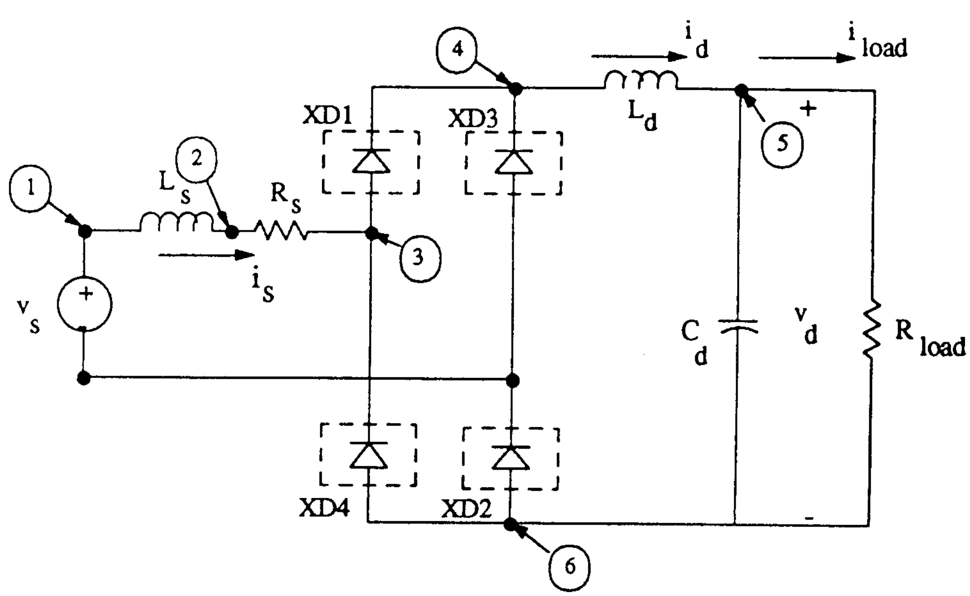
Valores Nominais: $V_{s(rms)} = 120V$ a 60Hz

$L_s = 1mH$; $R_S = 1m\Omega$

$L_d = 1\mu H$; $C_d = 1000\mu F$; $R_{load} = 20\Omega$

Procedimento:

1º - Analisar o circuito e atribuir nomes aos nós (ver figura seguinte).



2º - Criar o ficheiro **.CIR** correspondente ao circuito usando o Ps Control Shell (exemplo.cir).

```
EXEMPLO.CIR
* Ponte Rectificadora monofasica nao controlavel.
*
.LIB PWR_ELEC.LIB
.PARAM  FREQ = 60.0Hz
*
LS    1 2 1mH
RS    2 3 1m
*
LD       4 5 1uH
RLOAD  5 6 20.0
CD       5 6 1000uF IC=160V
*
XD1    3 4 DIODE_WITH_SNUB
XD3    0 4 DIODE_WITH_SNUB
XD2    6 0 DIODE_WITH_SNUB
XD4    6 3 DIODE_WITH_SNUB
*
VS    1 0 SIN(0 170V {FREQ} 0 0 0)
*
.TRAN  50us  50ms  0s  50us  UIC
.PROBE
.FOUR  60.0 i(LS)  i(LD)  i(XD1.DX)  V(5,6)
.END
```

3º - Executar o exemplo.cir, usando o comando *simulator* (tecla F11) do Control Shell, ou usando o ficheiro Pspice.exe.

4º - Obter as formas de onda de v_s, i_s e v_d.

5º - Examinar o ficheiro de saída (exemplo.out).

6º - Dos resultados da análise de Fourier contidos no ficheiro exemplo.out, calcule o *displacement power factor* (DPF) e o factor de potência da entrada (PF).

7º - Utilize a análise de Fourier do ficheiro exemplo.out e obtenha as formas de onda de i_s, is1, i_{s3} e i_{s5} (veja o ficheiro exemploa.cir). Comente o observado.

8º - Obtenha as formas de onda da corrente e tensão associadas com o díodo XD1. Calcule o valor médio e eficaz dessa corrente.

9º - Varie L_s para investigar a sua influência no DPF, no factor de potência, na percentagem total da distorção harmónica (%THD) e no *ripple* pico a pico de V_d (ver ficheiro exemplob.cir). Justifique.

10º - Varie o condensador de filtragem C_d para investigar a sua influência no *ripple* de V_d, no DPF e na %THD (ver exemploc.cir). Comente.

11º - Varie a potência da carga para investigar a sua influência no valor médio da tensão DC (ver exemplod.cir).

12º - Obtenha vs, i_s e v_d durante o transitório de arranque quando o condensador de filtragem não está inicialmente carregado. Justifique o pico da corrente (ver exemploe.cir).

13º - Substitua o lado DC por uma fonte de corrente I_d=10A (corresponde a uma elevada Ld). Faça L_s muito proximo de zero. Obtenha as formas de onda mais relevantes (ver exemplof.cir).

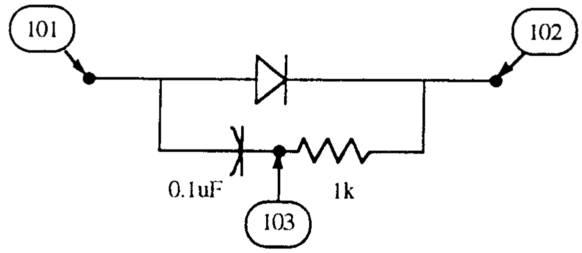
14º - Ao circuito inicial, retire o condensador e analise o conteudo harmónico da corrente do lado AC. Coloque L_d de modo a que a corrente na carga não se anule (ver exemplog.cir).

15º - Substitua os díodos por tiristores e justifique a diferença do conteudo harmónico com o observado na alínea anterior (ver exemploh.cir). Varie L_d ou R_{load} observando o seu efeito em v_d.

ANEXO - Modelos dos Subcircuitos

SUBCKT DIODE_WITH_SNUB

Objective: This subcircuit is used to represent line-frequency diodes.

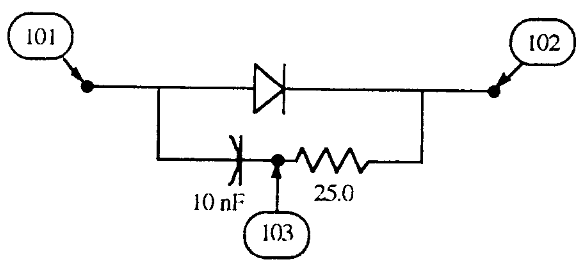


Input Circuit File Listing:

```
.SUBCKT     DIODE_WITH_SNUB    101     102
* Power Electronics: Simulation, Analysis & Education.....by N. Mohan.
DX          101     102     POWER_DIODE
RSNUB       102     103     1000.0
CSNUB       103     101     0.1uF
.MODEL      POWER_DIODE          D(RS=0.01, CJO=100pF)
.ENDS
```

SUBCKT SW_DIODE_WITH_SNUB

Objective: This subcircuit is used to represent sitching-frequency diodes.

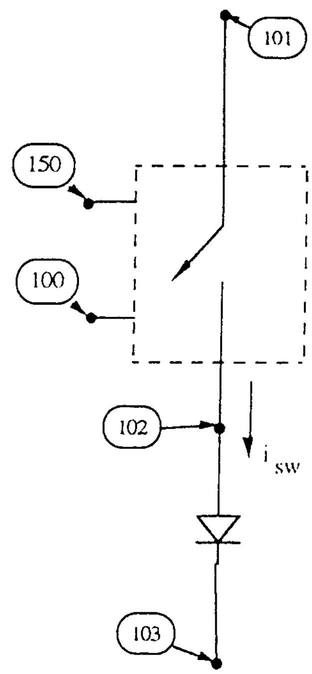


Input Circuit File Listing:

```
.SUBCKT     SW_DIODE_WITH_SNUB      101     102
* Power Electronics: Simulation, Analysis & Education.....by N. Mohan.
DX          101     102     POWER_DIODE
RSNUB       102     103     100.0
CSNUB       103     101     10nF
.MODEL      POWER_DIODE          D(CJO=0.001fF, IS=1E-6, RS=0.01)
.ENDS
```

SUBCKT SWITCH

Objective: This subcircuit is used to represent voltage-controlled switches through which the current is unididirectional.

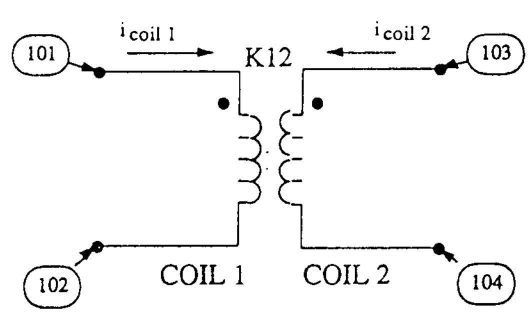


Input Circuit File Listing:

```
.SUBCKT     SWITCH     101    103    150    100
* Power Electronics: Simulation, Analysis & Education.....by N. Mohan.
SW          101    102    150    100    AC_SWITCH
DSW         102    103    POWER_DIODE
.MODEL      AC_SWITCH    VSWITCH ( RON=0.01 )
.MODEL      POWER_DIODE       D(CJO=0.001fF, IS=1E-6, RS=0.01)
.ENDS
```

SUBCKT TRANS_NONIDEAL

Objective: A nonideal tranformer.

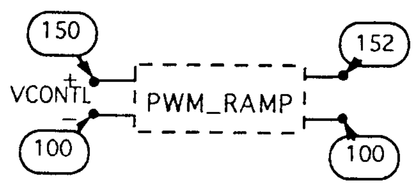


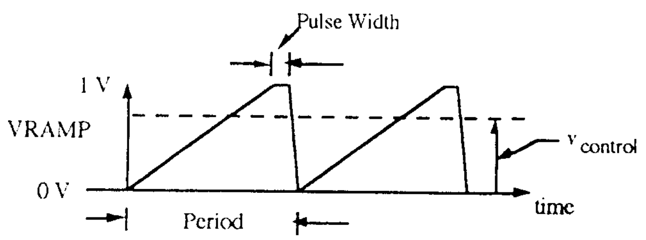
Input Circuit File Listing:

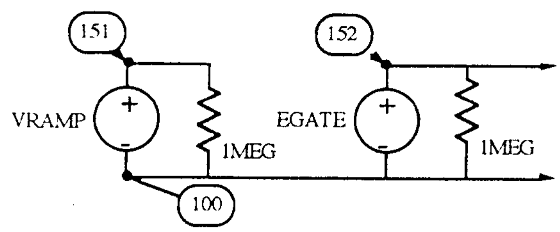
```
.SUBCKT     TRANS_NONIDEAL  101  102  103  104  PARAMS: ICOIL1=1A  ICOIL2=1A
* Power Electronics: Simulation, Analysis & Education.....by N. Mohan.
LCOIL1      101    102    {L1}    IC={ICOIL1}
LCOIL2      103    104    {L2}    IC={ICOIL2}
K12         LCOIL1        LCOIL2        {k}
.ENDS
```

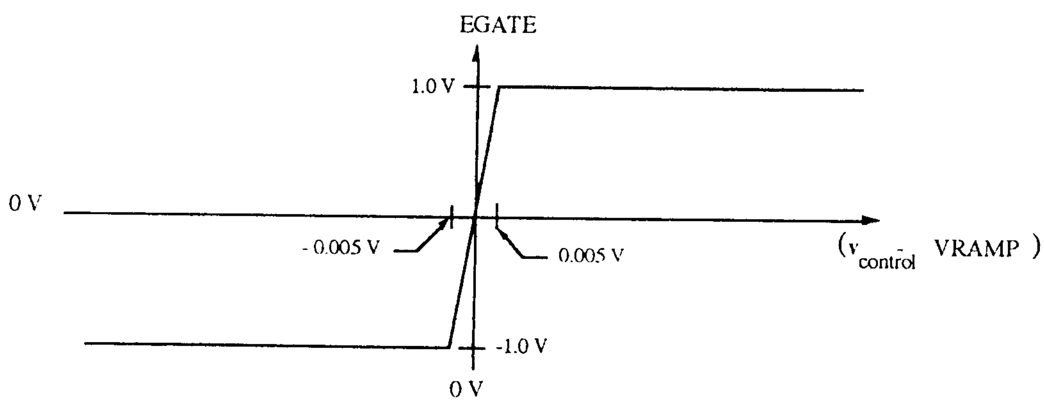
PWM_RAMP

Objective: This subcircuit is used for PWM control of single-switch dc-dc converters.









Input Circuit File Listing:

```
.SUBCKT      PWM_RAMP   150     100     152
* Power Electronics: Simulation, Analysis & Education.....by N. Mohan.
RCNTL        150     100     1MEG
RRAMP        151     100     1MEG
VRAMP        151     100     PULSE(0 1V 0 {RISE} {FALL} {PW} {PERIOD})
EGATE        152     100     TABLE { V(150) -  V(151) } = (-1.0,-1.0) (-0.005,-1.0) (0.0,0.0)
+                            (0.005,1.0) (1.0  ,       1.0)
RGATE        152     100     1MEG
.ENDS
```

Note: A finite pulse width PW is used to avoid voltage convergence problems.

ANEXO

Informação sobre o PSpice

Beginner's Instruction Set for PSpice* in Power Electronics Simulation

(* PSpice is a registered trademark of MicroSim Corporation.)

This beginner's instruction set shows **one out of many possible ways** to represent power electronic circuits in PSpice. The examples refer to the accompanying Power Electronics Simulation package. This instruction set is a very small subset of the complete PSpice User's Guide which can be obtained from the MicroSim Corporation by calling (714) 770-3022.

SPICE: Simulation Program with Integrated Circuit Emphasis (Univ.ersity of California - Berkeley)

PSpice: A version of SPICE developed by MicoSim Corporation.

Objective: To perform a transient analysis on a given power electronic circuit.

Simulation Steps:

1. Assign names to nodes and components in your circuit on paper.
2. Create an Input Circuit File (.CIR) using a text editor like Edit in MS-DOS 5.00 or higher.
3. Execute PSpice using the Input Circuit File (.CIR).
4. Examine the Output List File (.OUT) for the printed output using a text editor.
5. Plot the raw plot data points in the Data File (.DAT) using PROBE included with PSpice.

Contents of the Input Circuit File (.CIR):

1. Title Statement
2. Comment Statements (optional but desirable)
3. Library Statement (if necessary)
4. Data Statements
5. Solution Control Statements
6. Output Control Statements
7. .END Statement

Conventions in Input Circuit File (.CIR):

The user has to give each node in the circuit a name(See Exercise). The name can consist of characters or numbers. However, one node in the circuit must be ground, named 0. Other nodes can be named arbitrarily and don't need to be in a sequence 1, 2, 3, etc.

The Title Statement has to be the first line of the Input Circuit File and .END statement has to be the very last line. All other statements can be in any random sequence. PSpice is case insensitive, that is, uppercase and lowercase letters mean the same and therefore can both be used in any statement. The user has the option of using the following suffix letters to indicate various power-of-ten: f for 1e-15, p for 1e-12, n for 1e-9, u for 1e-6, m for 1e-3, k for 1e3, meg for 1e6, g for 1e9, and t for 1e12. If the suffix has other letters, for example in 10mH and 170V in Exercise , H and V will be ignored.)

Statements in the Input circuit file:

1. **Title Statement:** The first line of the Input Circuit File has to be any text which describes the circuit. It can be very brief or it can even be a blank line.

2. **Comment Statements:** Marked by * in column 1, followed by any text.

3. **Library Statement:** Subcircuits (discussed later on), for example representing a diode with an R-C snubber, can be put into a library file . If such a subcircuit is used, the Input Circuit File must point to that library file. (See Exercise)

   .LIB library file name

4. **Data Statements:** Note that the direction of current through a component is always defined to be from the +node to the -node.

   a. Parameter Statements: See Exercise where 60-Hz frequency is defined as a parameter FREQ. On using expressions to define parameters, see Exercise . Parameters defined this way are passed on to the subcircuits as well.

   .PARAM name = {a value or an expression}

   b. Passive Components:

Resistor:	Rname	+node	-node	value	
Inductor:	Lname	+node	-node	value	[IC= initial value of current]
Capacitor:	Cname	+node	-node	value	[IC= initial value of Voltage]
	(See Exercise for the initial value of voltage.)				

   Coupled Inductors: Kname Lname1 Lname2 value of mutual coupling coefficient k_{12}

   c. Independent Voltage and Current Sources for Transient Analysis

   Vname +node -node *transient specifications*
   Iname +node -node *transient specifications*

   Transient specifications :

   pulse (initial voltage, peak voltage, initial delay time, rise time, fall time, pulse width, pulse period)

   sin(0, amplitude, frequency, 0, 0, phase in degrees)
   (See Exercise .)

   d. Dependent sources

   Voltage-Controlled, Voltage Source:

   Ename +node -node +controlling node -controlling node gain

   Current-Controlled Current Source:

   Fname +node -node name of the current-measuring voltage source gain

e. Using Models built-in within PSpice: To use these models, we need two statements. One defines the component connections and the model name associated with the specific values of the parameters of the built-in PSpice model. The second statement defines the parameter values of the PSpice built-in model associated with that model name.

Diode: See the subcircuit DIODE_WITH_SNUB in Appendix . Note the difference between the name of the diode DX , and the *model* name POWER_DIODE. The same model can be used for any number of diodes in the circuit.

Dname +node -node model_name

.MODEL model_name D(CJO=0.001fF, RS=0.01)

Note that all parameters within the brackets are optional. If a value is not specified, then the default values are used by the program.

Voltage-Controlled Switch: This switch is represented by a very low resistance RON in its on-state, when the control voltage is greater than VON. When the control voltage is less than VOFF, the switch is represented by a very high resistance ROFF. The default values are as follows: RON=1 ohm, ROFF=1e+6 ohm, VON=1 V, VOFF= 0 V. Note that the current through the switch can flow in either direction when the switch is on.

Sname +node -node +controlling node -controlling node model_name

.MODEL model_name VSWITCH(RON=0.01)

Note that a large resistance is internally connected between the controlling nodes to keep them from floating. Internally, the controlling nodes are electrically isolated from the switch nodes.

h. Subcircuits: See Exercise . The use of a subcircuit requires a call to the subcircuit. The definition of the subcircuit can be a part of the Input Circuit File (.CIR) Otherwise, the Input Circuit File must point to the library, by means of a .LIB statement, where the subcircuit definition is located.

Call to a Subcircuit: The parameter values defined in the call statement are passed on to the subcircuit. This way, the same subcircuit can be used with different parameter values.

Xname node 1 node 2 ...node n subckt_name PARAMS: name 1=value, name 2=value,.......

Subcircuit Definition: The subcircuit can be given any node and component names, without interference with the names in the calling circuit. The only exception is 0 which is treated as ground everywhere. This definition must end with a .ENDS statement.

.SUBCKT subckt_name node 1 node 2 ...node n PARAMS: name 1=value, name 2=value,.....
..
..
.ENDS

g. Analog Controller Modeling:

g.1 Instantaneous Transfer Functions (algebraic relationships):

By means of VALUE:

Ename +node -node VALUE = {an expression}

By means of TABLE:

Ename +node -node TABLE { expression } = point-by-point description of relationship

g.2 Laplace Transforms:

Ename +node -node LAPLACE { expression } = { transform }

h. Parametric Analysis: See Example . It requires two statements; a .PARAM statement and a .STEP statement to carryout the analysis with a specified list of values of a parameter to be varied. For a linear variation where a start value, a final value, and a linear increment value are specified in this order,

.PARAM parameter_to_be_varied =1

.STEP PARAM parameter_to_be_varied LIST a list of values in ascending or descending order

5. Solution Control Statements:

a. Transient Analysis: See Exercise where UIC, which stands for Use Initial Conditions, must be used. Otherwise, a dc steady state analysis will be performed and the resulting voltages and currents will be used as initial conditions for the transient analysis.

.TRAN print_step final_time results_delay step_ceiling UIC

Note that the print_step dictates the frequency at which the output variables are printed in the .OUT file. Both the print ouput and the raw plot points are saved starting at results_delay. The minimum time step is not under user control but the maximum time step is dictated by step_ceiling.

b. Fourier Analysis: See Exercise where the fundamental frequency must be specified. Also note how the current through a subcircuit component is referred.

.FOUR fundamental frequency output_variables

Note that the Fourier analysis printed in the .OUT file is based on the values in the last fundamental-frequency cycle. For each harmonic h, the Fourier Component and the Phase in degrees are the peak amplitude and the phase angle in the equation, for example, $i_h = I_{h,\ peak} \sin(2\pi f_h + \phi_h)$.

6. Output Control Statements:

a. Printed output in the .OUT File:

.PRINT TRAN output_variables

b. Raw Plot Points for PROBE: Only the variables specified on the list are stored. Otherwise, all voltages and currents are stored, possibly resulting in an extremely large .DAT file.

.PROBE output_variables

7. End Statement:

.END

Executing PSpice: In the directory where the PSpice program, the Input Circuit File, and the library file are located, issue the following command:

PSPICE name of the input circuit file

Examine the Printed Output: The printed output will have the same name as the input circuit file except for the extension of .OUT. This can be examined by means of any text editor. It should be done, especially if the PSpice run aborted due to errors.

Plotting Using PROBE: If the Input Circuit File contained a .PROBE statement, the raw plot points are stored in a plot file with the same name as the input circuit file except for the extension of .DAT. Also, the program automatically goes into a plotting mode using PROBE. For plotting at a later time, issue the following command:

PROBE name of the plot file

Just follow the instructions within PROBE for plotting. To obtain the list of variables available for plotting after having selected ADD TRACE, press the F4 key. To obtain the list of subcircuit variables available for plotting, press the F4 key again and select Show_internal_subcircuit_nodes.

Hints:

Floating Nodes: There must be a dc path to ground from each node. Otherwise, an error message in .OUT file will indicate that a particular node is floating. A simple remedy is to connect a large resistance, for example a 1e7 ohm resistance, from that node to ground which does effect the circuit performance.

Inductor Loops: If there is a loop involving inductors with zero resistance, an error message will be printed in the .OUT file. Insert a small resistance anywhere in this loop.

Convergence: In modeling of power electronic circuits, convergence and the speed of simulation are two of the biggest problems. Always include R-C snubbers across diodes and switches to avoid a sharp discontinuity which results in PSpice proceeding with a very small time step. In case of convergence problems, increase the error tolerance VNTOL from its default value of 1μV and ABSTOL from its default value of 1pA to possibly 1μA by means of the .Options statement below. ITL5=0 sets the total iteration limit for all points in the transient analysis to infinity.

```
.Options  abstol = 1uA    itl5= 0    vntol=1e-3
```

Floating Nodes: At each node, there must be atleast two connections, otherwise the run will abort with an error message. A large resistance from that node to ground will correct this problem.

Step_Ceiling in the .TRAN Statement: To avoid very large time steps which result in crude plots, the user should provide the value of the maximum time step. An arbitrarily small value may result in a long simulation time.