

Rohrheimer-Str. 20
D-64625 Schwanheim

Benutzerhandbuch

SCEPTRE
Ein Programm zur Analyse und Simulation
nichtlinearer elektrischer Netzwerke und
dynamischer Systeme

Wolf-Rainer Novender

27. Mai 2000

©1989 – 2000. Alle Rechte vorbehalten.

Inhaltsverzeichnis

1. SCEPTRE-Anweisungen	1
1.1. Sprachvereinbarungen	1
1.2. Vorbereitungen	3
1.3. Eingabeformat	5
1.4. Eingabestruktur	5
1.5. ELEMENTS	6
1.5.1. Allgemeine Zweipolform	6
1.5.2. Gegeninduktivitäten	6
1.5.3. Widerstandsabhängige Quellen	8
1.5.4. Gesteuerte Stromquellen	8
1.5.5. Zeitliche Ableitungen von Quellen	8
1.5.6. Modellaufruf	8
1.5.7. Elemente mit Toleranzgrenzen	8
1.5.8. Quellen bei der Wechselstromanalyse	9
1.6. DEFINED PARAMETERS	9
1.7. OUTPUTS	11
1.8. INITIAL CONDITIONS	12
1.9. FUNCTIONS	12
1.9.1. EQUATION-Anweisung	12
1.9.2. TABLE-Anweisung	13
1.10. RUN CONTROLS	14
1.10.1. DC-Analyse	14
1.10.2. AC-Analyse	16
1.10.3. TR-Analyse	17
1.10.4. DC- und TR-Analyse	18
1.10.5. DC- und AC-Analyse	18
1.10.6. AC- und TR-Analyse	18

Inhaltsverzeichnis

1.10.7. Ausgabesteuerung	18
1.10.8. Wiederholungsläufe	18
1.10.9. Ablaufsteuerung	19
1.10.10. Testhilfen	20
1.11. SENSITIVITY, MONTE CARLO, WORST CASE, OPTIMIZATION	20
1.12. Modelle	21
1.12.1. Modellbeschreibung	22
1.12.2. Modellaufruf	23
1.13. RERUN DESCRIPTION	24
1.14. CONTINUE	28
1.15. RE-OUTPUT	29
1.16. END	29
2. Allgemeine Hinweise	30
2.1. Fehlerdiagnose	30
2.2. Einige Grundregeln	30
2.3. Zeitabhängige Induktivitäten und Kapazitäten	31
2.4. Tabellen	31
2.5. Arithmetische Anweisungen	31
2.6. Topologische Einschränkungen	31
2.6.1. DC-, AC- und TR-Analyse	31
2.6.2. DC-Analyse	32
2.7. "Computational Delay"	32
2.8. Wahl des Integrationsverfahrens	33
2.9. Definition abhängiger Quellen	33
2.10. Strom- und Spannungsmessungen	35
2.11. Verwendung eigener FORTRAN-Unterprogramme	35
2.12. Die Vektornotation	36
2.13. Wahl der Zustandsvariablen	36
2.14. Berechnung von Integralen	37
2.15. Simulationsstatistik	37
2.16. Druckausgabe	38
2.17. Anfangsbedingungen	39
2.17.1. Arbeitspunkt	39
2.17.2. Implizite Integration	39

2.17.3. Wiederholungsläufe	40
2.18. Periodische Funktionen	41
2.18.1. Ideale Funktionen	41
2.18.2. FPULSE	42
2.18.3. FSINSQ	43
2.19. Reservierte Variablennamen	44
2.20. Wechselstromquellen	44
3. SCEPTRE Beispiele	45
3.1. A01: Inverter-Schaltung mit RC-Last	45
3.2. A02: Emitterfolger mit Übertrager	47
3.3. A03: Darlington-Schaltung (DC-Analyse)	48
3.4. A04: Kleinsignal-Ersatzschaltung	50
3.5. A05: Differentialgleichungssystem	52
3.6. Toleranzanalysen einer Darlington-Schaltung	53
3.6.1. A07: Monte-Carlo-Analyse	53
3.6.2. A08: Empfindlichkeitsanalyse	53
3.6.3. A09: ‘Worst Case’-Analyse	56
3.6.4. A10: Optimierung	59
3.7. A11: Kleinsignal-Ersatzschaltung (AC)	59
3.8. A13: Elektro-hydraulischer Stellantrieb	62
4. Systeminformationen zur Benutzung von SCEPTRE	66
4.1. Programmorganisation	66
4.2. Aufruf unter Unix	66
4.3. Grafikschnittstellen	68
4.3.1. Das Unterprogramm INTPLT	68
4.3.2. Beispiel zur Ausgabe mittels ANSI Treibers	70
4.3.3. Das Unterprogramm POST	71
4.4. Auswertung der Ergebnisse (“Postprocessing”)	72
4.5. Das Grafikprogramm NGP	73
4.5.1. Übersicht der Kommandos	74
4.5.2. NGP und GNUPLOT	74
4.5.3. Konfigurationsdatei <code>ngp.rc</code>	75
5. Fortsetzungsläufe (CONTINUE)	76

Inhaltsverzeichnis

A. Dioden- und Transistor-Ersatzschaltbilder	77
B. Programmkapazität	80
C. Übersicht aller Anweisungen und internen Variablen	81

Abbildungsverzeichnis

1.1. Spannungsverlauf	13
2.1. Zeitmultiplexsignal	42
2.2. Die periodische Funktion FPULSE	43
2.3. Die periodische Funktion FSINSQ	43
3.1. Inverterschaltung und für SCEPTRE aufbereitetes Ersatzschaltbild	45
3.2. Spannung VC1 in Abhängigkeit von JX entsprechend TABLE 1	46
3.3. Emitterfolger mit Übertrager, Transistor-Ersatzschaltbild	47
3.4. Spannungen VL1 und VL2 am Übertrager	49
3.5. Darlington Schaltung	49
3.6. Ergebnisse der Gleichstromanalysen	51
3.7. Kleinsignal-Schaltung, Ersatzschaltbild mit h-Parametern	51
3.8. Ergebnisse der Monte-Carlo-Analyse	54
3.9. Ergebnisse der Empfindlichkeitsanalyse	57
3.10. Ergebnisse der 'Worst Case'-Analyse für die Spannung VCET1	58
3.11. Optimierungsergebnisse für die Spannung VR3	60
3.12. Kleinsignalschaltung, Transistor-Ersatzschaltbild	61
3.13. Schaltung des Regelkreises	63
3.14. Magnetstrom ILTORK als Funktion der Zeit	65
3.15. Regelgröße PDIS als Funktion der Zeit	65
4.1. Programm- und Datenfluß	67
A.1. SCEPTRE Diodenersatzschaltbild	77
A.2. SCEPTRE npn Transistor-Ersatzschaltbild	78
A.3. SCEPTRE pnp Transistor-Ersatzschaltbild	79

Tabellenverzeichnis

1.1. Klassifizierung der Elemente	3
1.2. Länge der frei wählbaren Namen	4
1.3. Beispiele für Maßeinheiten	4
1.4. Klassifizierung weiterer SCEPTRE-Größen	4
1.5. Festlegung von Strom und Spannung bei allgemeinen Zweipolen	7
1.6. Abhängige Variablen	21
1.7. Unabhängige Variablen	21
B.1. Programmkapazität	80
C.1. SCEPTRE-Anweisungen und interne Variablen	82

1. SCEPTRE-Anweisungen

1.1. Sprachvereinbarungen

Die allgemeine Form einer SCEPTRE-Anweisung wird in diesem Handbuch eingerahmt dargestellt. Die Schlüsselwörter werden dabei in Großbuchstaben geschrieben, z. B.

`RUN INITIAL CONDITIONS ONLY` .

Vom Benutzer vorzugebene Parameter werden kursiv gedruckt, z. B.

`M name, L name1 – L name2` .

Geschweifte Klammern deuten alternative Eingaben an:

`{ DE } name = value` .
`{ DJ }`

Wahlweise Angaben werden durch eckige Klammern gekennzeichnet:

`variable1 [, variable2...], PLOT (variable3)` .

Voreinstellungen sind entweder unterstrichen oder stehen neben der Umrahmung.

An verschiedenen Stellen taucht der Begriff 'Karte' auf. Er ist gleichbedeutend mit der Zeile einer Datei.

Zur Vereinfachung werden in diesem Handbuch nachstehend aufgeführte Vereinbarungen getroffen. Die englische Bezeichnung wurde im allgemeinen beibehalten:

element name: Name einer Netzwerkkomponente

Beispiel: R102, CLE, E1, LCOIL

node: Name eines Knotenpunktes

Beispiel: 0, GND, MASSE, EGON

number: numerische Konstante im E-, I- oder F-Format, mit oder ohne Vorzeichen, max. 13 Zeichen lang

Beispiel: 314, 3.14, -.314, 3.14E-3

constant: wie **number**, jedoch immer mit Dezimalpunkt

value: dazu gehören:

- **number**
- **defined parameters**

1. SCEPTRE-Anweisungen

- **TABLE...**
- **EQUATION...**
- **EXPRESSION...**
- **external function**

special value: dazu gehören:

- **value**
- **constant * IR...**, **constant * VR...**
Beispiel: 3.*IRL, .5E-7*VR2
- **value * J...**
Beispiel: PS3*JEIN, T43(VLS)*JAUS
- **DIODE TABLE...**
- **DIODE EQUATION...**

variable: dazu gehören:

- Spannungen und Ströme (**V...**, **I...**)
Beispiel: VRL, IRL, VJEIN, IEIN
- Quellen und ihre Ableitungen (**E...**, **DE...**, **J...**, **DJ...**)
Beispiel: E1, DE1, J2, DJ2
- **defined parameters**
- Wert einer Netzwerkkomponente (**element name**)
Beispiel: R1, CES, ML1L2
- **TIME**
- **interne SCEPTRE-Parameter**
Beispiel: XSTPSZ, XIR, XTMON

TABLE *name(variable)*: Netzwerkgrößen können auch in tabellarischer Form vorliegen. Die Tabelle wird unter FUNCTIONS definiert. *Variable* entspricht der unabhängigen Variablen. Fehlt die Variable, wird **TIME** als unabhängige Variable angenommen.

Verkürzte Schreibweise: *Tname(variable)*

Beispiele:

```
TABLE UIC(VC1)
```

```
TREGEL(PU4)1
```

EQUATION *name(arg1, arg2,...)*: Werden Netzwerkgrößen nach einer arithmetischen Anweisung berechnet, kann man diese Schreibweise benutzen. Sie hat die Wirkung eines Funktionsaufrufs, bei dem die Parameter *arg1, arg2,...* übergeben werden. Die mathematische Formulierung wird unter FUNCTIONS beschrieben. Die Argumente können vom Typ **variable**, **constant**, bzw. **TABLE...** sein. Die **EQUATION**-Anweisung ist dann vorteilhaft, wenn mehrere Netzwerkgrößen nach der gleichen Vorschrift mit verschiedenen Zahlenwerten berechnet werden sollen.

Verkürzte Schreibweise: *Qname(arg1, arg2,...)*

Beispiele:

```
EQUATION A1 (E1, TIME, 23.4, TABLE9)
```

```
QA1(220., TIME, RA, T9(VRA))
```

¹siehe auch Abschnitt 2.4

1. Buchstabe	Element
R	Widerstand
C	Kapazität
L	Induktivität
M	Gegeninduktivität
E	Spannungsquelle
J	Stromquelle

Tabelle 1.1.: Klassifizierung der Elemente

EXPRESSION *name* (*arithmetische Anweisung*): Diese Schreibweise hat die gleiche Wirkung wie **EQUATION**. Sie erlaubt jedoch die direkte Angabe der Berechnungsvorschrift und erweist sich als vorteilhaft beim einmaligen Berechnen von Netzwerkgrößen. Um Verwechslungen mit internen SCEPTRE-Parametern zu vermeiden, sollte *name* möglichst nur aus Ziffern bestehen. Die mathematische Formulierung ist identisch mit der in FORTRAN. Neben den 5 Grundrechenarten (+, -, /, *, **) können auch **external functions** benutzt werden. Numerische Konstanten müssen mit einem Dezimalpunkt angegeben werden.

Verkürzte Schreibweise: *Xname*(...)

Beispiele:

E1=X100(310.*DSIN(314.*TIME))

POM=X209(2.*PI*TIME)

PI=X210(4.*DATAN(1.))

CBE=X2(IR5*XTABLE(T23,VRA))²

DIODE TABLE *name* (*variable*): Liegt eine Diodenkennlinie in Tabellenform vor, so bezieht sich *variable* automatisch auf die Diodenspannung

Beispiel: JA=DIODE TABLE OY100 (VCJA)

DIODE EQUATION (x_1, x_2): Eine Diodenkennlinie kann auch in der Form $J = x_1(e^{x_2 V_J} - 1)$ vorliegen. Die Diode wird automatisch durch eine nach der o. a. Formel gesteuerten Stromquelle J ersetzt, V_J ist die Diodenspannung.

Beispiel: DIODE EQUATION (3.E-14,38.)

external function: Für arithmetische Anweisungen stehen alle Funktionen der FORTRAN Unterprogrammibliothek sowie selbstgeschriebene FORTRAN Unterprogramme zur Verfügung. Es müssen die doppelgenauen Aufrufe benutzt werden, z.B. für reelle Argumente DSIN, DCOS, DEXP, DATAN, DLOG10... Für komplexe Argumente (AC-Analyse) stehen folgende Funktionen in etwas abgeänderter Schreibweise zur Verfügung: ZSQRT, ZSIN, ZCOS, DZABS, ZLOG, ZCMPLX.

interne SCEPTRE-Parameter: Diese Programmvariablen werden von SCEPTRE benötigt; sie stehen dem Benutzer in den Eingabeanweisungen direkt und in selbstgeschriebenen FORTRAN Unterprogrammen über einen COMMON-Bereich zur Verfügung.

Beispiele: XSTPSZ, XIR, XTMON, XSTOPT.

1.2. Vorbereitungen

Zuerst wird das Ersatzschaltbild des zu untersuchenden Netzwerks gezeichnet und jedem Element ein Name zugeordnet. Dabei klassifiziert der 1. Buchstabe des Namens das Element gemäß Tabelle 1.1.

Weiterhin werden alle Knotenpunkte mit einem Namen versehen. Ein Name darf nur aus alphanumerischen Zeichen bestehen. Die Länge der frei wählbaren Namen ist begrenzt (siehe Tabelle 1.2).

²wegen der geänderten Syntax siehe Abschnitt 2.4

1. SCEPTRE-Anweisungen

	circuit	model
	description	
Knotenpunkte ‡	6	3 †
Elemente	5	2 †
Definierte Parameter	6	3 †
Tabellennamen	5	2 †
EQUATION-Namen	5	2 †
Modellnamen ‡	18	18
Ausgabevariablen	6	3 †
Modellaufrufe	3 †	-

Tabelle 1.2.: Länge der frei wählbaren Namen

Widerstand	Ω	k Ω	k Ω	k Ω
Kapazität	F	pF	μ F	nF
Induktivität	H	μ H	H	mH
Strom	A	mA	A	mA
Spannung	V	V	kV	V
Zeit	s	ns	ms	μ s
Frequenz	Hz	GHz	kHz	MHz

Tabelle 1.3.: Beispiele für Maßeinheiten

Die mit ‡ markierten Namen dürfen auch mit numerischen Zeichen beginnen (z.B. 2N3055 als Modellname oder 0 als Knotenpunktname). Die mit † versehenen Zahlenangaben sind empfohlene Werte und sollten möglichst nicht überschritten werden.

Die Maßeinheiten können frei gewählt werden. Sie müssen nur die Verknüpfungsgleichungen

$$U = IR \quad \tau = \frac{L}{R} = RC \quad f = \frac{1}{T}$$

widerspruchsfrei erfüllen (siehe Tabelle 1.3).

Außer den Netzwerkelementen können in SCEPTRE auch Tabellen, arithmetische Anweisungen und andere Größen definiert werden. Sie werden ebenfalls durch ihren 1. Buchstaben gekennzeichnet entsprechend Tabelle 1.4.

In jeden Stromzweig legt man eine Stromrichtung, für die Spannungsquellen die Polarität fest. Dabei ist zu beachten, daß SCEPTRE bei Verbrauchern Spannungs- und Strompfeile gleichsinnig, bei Quellen gegensinnig annimmt.

1.Buchstabe	Bedeutung
P	definierter Parameter, reell
W	definierter Parameter, komplex
T, TABLE	Tabelle
Q, EQUATION	arithmetische Anweisung als Formelaufruf
X, EXPRESSION	arithmetische Anweisung, direkt
V	Spannung an einem Netzwerkelement
I	Strom durch ein Netzwerkelement
D	Differential d/dt
G	vollständiges Differential
F	eigene FORTRAN-Unterprogramme

Tabelle 1.4.: Klassifizierung weiterer SCEPTRE-Größen

1.3. Eingabeformat

Alle SCEPTRE-Anweisungen müssen in Großbuchstaben zwischen den Spaltenpositionen 1 und 72 einer Textdatei geschrieben werden. Die Eingabe ist formatfrei, Leerzeichen und -zeilen können zur besseren Lesbarkeit eingefügt werden. Mehrere Anweisungen dürfen — durch Kommata getrennt — in einer Zeile geschrieben werden. Anweisungen können auf mehreren Eingabezeilen fortgesetzt werden, und zwar unmittelbar nach den Zeichen

+ - * / , . () =.

SCEPTRE ignoriert innerhalb einer Zeile alle Informationen nach einem Semikolon “;”. Die Datei darf keine Sonderzeichen, z. B. TAB, enthalten.

1.4. Eingabestruktur

Eine SCEPTRE-Eingabe kann je nach Anwendung aus folgenden Steuerkarten bestehen:

Netzwerkbeschreibung

CIRCUIT DESCRIPTION
 maximal 11 Kommentarkarten
 ELEMENTS
 DEFINED PARAMETERS
 OUTPUTS
 INITIAL CONDITIONS
 FUNCTIONS
 SENSITIVITY
 MONTE CARLO
 WORST CASE
 OPTIMIZATION
 RUN CONTROLS

Modellbeschreibung

MODEL DESCRIPTION
 MODEL name
 maximal 11 Kommentarkarten
 ELEMENTS
 DEFINED PARAMETERS
 OUTPUTS
 FUNCTIONS
 INITIAL CONDITIONS

Parametervariation

RERUN DESCRIPTION
 maximal 11 Kommentarkarten
 ELEMENTS
 DEFINED PARAMETERS
 INITIAL CONDITIONS
 FUNCTIONS
 RUN CONTROLS

Fortsetzungslauf

CONTINUE
 RUN CONTROLS

1. SCEPTRE-Anweisungen

zusätzliche Ausdruckkopien	RE-OUTPUT OUTPUTS RUN CONTROLS
letzte Karte	END

Die Reihenfolge der Steuerkarten ist beliebig. Zu jeder der o. a. Steueranweisungen gehören Datenkarten, die nachfolgend im einzelnen beschrieben werden. Fehlen Daten zu einer Steuerkarte, kann diese ebenfalls entfallen.

1.5. ELEMENTS

Alle Netzwerkkomponenten werden unter dieser Überschrift definiert, und zwar in der allgemeinen Form:

$$\boxed{\text{element, node1 - node2 = value}}$$

Dabei wird festgelegt, daß der Strom durch *element* von *node1* nach *node2* fließt.

1.5.1. Allgemeine Zweipolform

Für die entsprechenden Netzwerkelemente lautet die Eintragung (siehe auch Tabelle 1.5) :

$$\boxed{\left\{ \begin{array}{l} R \\ C \\ L \\ E \\ J \end{array} \right\} \text{ name, node1 - node2 = value}}$$

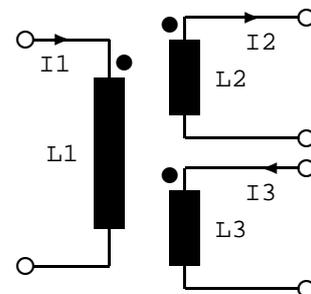
1.5.2. Gegeninduktivitäten

$$\boxed{M \text{ name, } L \text{ name1} - L \text{ name2} = \text{value}}$$

L_{name1} und L_{name2} sind die beiden miteinander gekoppelten Induktivitäten. Es muß die Bedingung $M < \sqrt{L_1 L_2}$ erfüllt sein.

Das Vorzeichen von M ist positiv, wenn die induzierten Spannungen der Gegen- und Selbstinduktivitäten in einer Wicklungsanordnung die gleichen Richtungen haben. Für das dargestellte Beispiel gilt:

$$\begin{aligned} M_{12}, L_1 - L_2 &= - \text{value} \\ M_{13}, L_1 - L_3 &= + \text{value} \\ M_{23}, L_2 - L_3 &= - \text{value} \end{aligned}$$



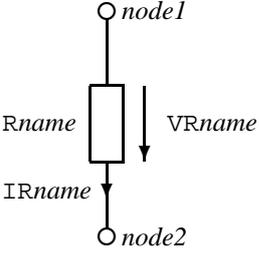
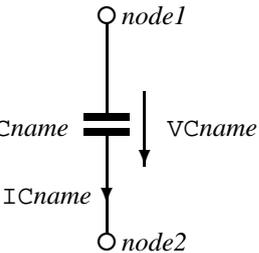
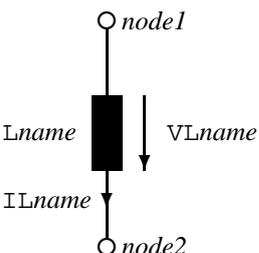
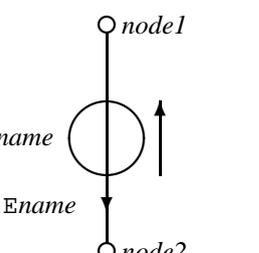
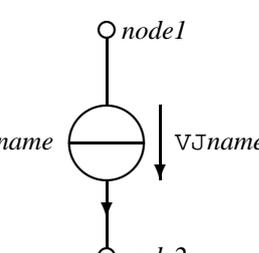
	$Rname, node1 - node2 = value$
	$Cname, node1 - node2 = value$
	$Lname, node1 - node2 = value$
	$Ename, node1 - node2 = value$
	$Jname, node1 - node2 = value$

Tabelle 1.5.: Festlegung von Strom und Spannung bei allgemeinen Zweipolen

1. SCEPTRE-Anweisungen

1.5.3. Widerstandsabhängige Quellen

$$E \text{ name, node1 - node2} = \text{constant} * VR \dots$$

$$J \text{ name, node1 - node2} = \text{constant} * IR \dots$$

1.5.4. Gesteuerte Stromquellen

Gesteuerte Stromquellen dienen vorwiegend zur Simulation von Halbleiterschaltungen.

Stromquellen 1. Ordnung

$$J \text{ name, node1 - node2} = \left\{ \begin{array}{l} \text{DIODE TABLE} \dots \\ \text{DIODE EQUATION} \dots \end{array} \right\}$$

Stromquellen 2. Ordnung

$$J \text{ name1, node1 - node2} = \text{value} * J \text{ name2}$$

Jname2 ist eine Stromquelle 1. Ordnung.

1.5.5. Zeitliche Ableitungen von Quellen

Treten in einem Netzwerk Maschen auf, die nur eine Spannungsquelle und einen Kondensator bzw. eine Stromquelle und eine Induktivität enthalten, verlangt SCEPTRE zusätzlich die zeitliche Ableitung dieser Quellen³.

$$\left\{ \begin{array}{l} DE \\ DJ \end{array} \right\} \text{ name} = \text{value}$$

1.5.6. Modellaufruf

$$\text{name, node1 - node2} - \dots = \text{MODEL modelname}$$

1.5.7. Elemente mit Toleranzgrenzen

Diese Angaben werden nur bei MONTE CARLO, WORST CASE und/oder OPTIMIZATION benötigt.

$$\left\{ \begin{array}{l} R \\ E \\ J \end{array} \right\} \text{name, node1 - node2} = \text{number1} \left\{ \begin{array}{l} (\text{number2, number3}) \\ (\text{number4}) \end{array} \right\}$$

number1 entspricht dem Nominalwert, *number2* dem unteren, *number3* dem oberen Wert, *number4* gibt die prozentuale Abweichung an.

³Diesen Fall kann man vermeiden, indem man einen geeigneten Widerstand in diese Maschen einfügt.

1.5.8. Quellen bei der Wechselstromanalyse

$$\left\{ \begin{array}{c} E \\ J \end{array} \right\} \text{ name, node1 - node2} = \left(\left\{ \begin{array}{c} \text{constant} \\ \text{def. par.} \\ \text{TABLE} \dots \\ \text{FREQ} \end{array} \right\}, \left\{ \begin{array}{c} \text{constant} \\ \text{def. par.} \\ \text{TABLE} \dots \\ \text{FREQ} \end{array} \right\}, \left\{ \begin{array}{c} \text{DEGREES} \\ \text{RADIANS} \\ \text{COMPLEX} \end{array} \right\} \right)$$

Bei der Angabe DEGREES (Phasenwinkel in Grad) oder RADIANS (Phasenwinkel im Bogenmaß) stellt das erste Argument die Amplitude, das zweite den Phasenwinkel dar. Benutzt man COMPLEX, dann enthält das erste Argument den Realteil, das zweite den Imaginärteil.

Alle R-, C-, L- und M-Werte werden bei einer Wechselstromanalyse als konstant angenommen. SCEPTRE setzt automatisch die vom Benutzer vorgegebenen Anfangsbedingungen ein oder, falls diese fehlen, berechnet sie zum Zeitpunkt $t = 0$.

Alle konstanten und zeitabhängigen Quellen werden bei der Wechselstromanalyse durch den Wert Null ersetzt.

Bei der Wechselstromanalyse ist im Gegensatz zur transienten Analyse unter bestimmten netzwerktopologischen Voraussetzungen die Angabe der zeitlichen Ableitungen einer Quelle nicht erforderlich. Sie ergibt sich automatisch durch Multiplikation mit $j\omega$.

1.6. DEFINED PARAMETERS

Unter diesem Titel können zusätzliche Zwischenvariablen definiert werden, z.B. um aus mehreren Netzwerkgrößen eine neue Größe zu berechnen. Allgemeine Form:

$$Pname = value$$

Beispiele:

$$PR5 = X10 (IR5 * VR5)$$

$$PTAU = X11 (C1 * R9)$$

Die zeitliche Ableitung einer Variablen $\frac{d}{dt}(P(t))$ kann ebenfalls angegeben werden:

$$DPname = value$$

Beispiel zur Lösung eines Differentialgleichungssystems 1. Ordnung:

$$\begin{aligned} \dot{x} &= -6x + 5y + 10 \\ \dot{y} &= 5x - 7y + 0,2z \\ \dot{z} &= 0,2y - 0,2z - 0,5 \end{aligned}$$

Anfangsbedingungen: $x(0) = 6, y(0) = 5, z(0) = 4$

$$\begin{aligned} DPX &= X1 (-6 . *PX + 5 . *PY + 10 .) \\ DPY &= X2 (5 . *PX - 7 . *PY + 0 . 2 *PZ) \\ DPZ &= X3 (0 . 2 *PY - 0 . 2 *PZ - 0 . 5) \\ PX &= 6 \\ PY &= 5 \\ PZ &= 4 \end{aligned}$$

1. SCEPTRE-Anweisungen

Zwischenvariablen können auch Toleranzgrenzen haben⁴:

$$\boxed{P \text{ name} = \text{number1} \left\{ \begin{array}{l} (\text{number2}, \text{number3}) \\ (\text{number4}) \end{array} \right\}}$$

Das vollständige Differential der Funktion $P(x, y, \dots)$, das bei SENSITIVITY, OPTIMIZATION und WORST CASE benötigt wird,

$$dP = \frac{\partial P}{\partial x} dx + \frac{\partial P}{\partial y} dy + \dots$$

wird in SCEPTRE folgendermaßen definiert:

$$\boxed{GPname = Pname1 * Dname2 + Pname3 * Dname4 + \dots}$$

Dabei stellen $Pname1$, $Pname3$ die partiellen Ableitungen $\partial P/\partial x$, $\partial P/\partial y$, $Dname2$, $Dname4$ die Differentiale dx , dy dar. Das vollständige Differential *muß* in dieser Form geschrieben werden (keine EXPRESSION- oder EQUATION-Anweisung).

Beispiele:

```
PWR = X1 ( IRL ** 2. + ILX ** 2. )
GPWR= P2 * DIRL + P3 * DILX
P2   = X76 ( 2. * IRL )
P3   = X77 ( 2. * ILX )
```

Die Differentiale $Dname2$, $Dname4 \dots$ müssen angegeben werden, wenn die Variablen $name2$, $name4 \dots$ als unabhängige Variablen definiert werden.

Für die AC-Analyse müssen die Zwischenvariablen komplex sein. Ihnen kann man reelle Werte

$$\boxed{Wname = \text{value}}$$

oder komplexe Werte

$$\boxed{Wname = (\text{value1}, \text{value2}) \text{ typ}}$$

zuordnen⁵. Komplexe Zwischenvariablen können nur zur Berechnung von Größen benutzt werden, die ausgegeben werden sollen, d.h. sie dürfen weder in einer EQUATION-, EXPRESSION- oder TABLE-Anweisung noch in einer Funktion auftreten.

⁴siehe Kapitel 1.5: Elemente mit Toleranzgrenzen

⁵Bedeutung der Parameter siehe Kapitel 1.5: Quellen bei der Wechselstromanalyse

1.7. OUTPUTS

Unter diesem Titel können alle Größen vom Typ **variable** aufgeführt werden. SCEPTRE bietet die Möglichkeit, die Ergebnisse sowohl in Abhängigkeit von der Zeit/Frequenz als auch von anderen unabhängigen Variablen auszudrucken/plotten. Allgemein erhält man einen tabellarischen Ausdruck in Abhängigkeit von der Zeit/Frequenz mit der Anweisung

$$\boxed{\text{variable1} [, \text{variable2} \dots]} .$$

Einen zusätzlichen Schnelldruckerplot, und zwar jede Variable einzeln auf einer Seite, erhält man mit

$$\boxed{\text{variable1} [, \text{variable2} \dots], \text{PLOT}} .$$

Plotterzeichnungen als Funktionen anderer unabhängiger Variablen werden angefordert mit

$$\boxed{\text{variable1} [, \text{variable2} \dots], \text{PLOT} (\text{variable3})} .$$

Außer TIME können alle Größen umbenannt werden mit

$$\boxed{\text{variable1} (\text{name1}) [, \text{variable2} (\text{name2}) \dots], \text{PLOT} (\text{variable3} (\text{name3}))} .$$

Die neuen Namen *name1*... dürfen maximal 6 Zeichen lang sein.

Beispiele:

```
E1, IE1, VRL, XSTPSZ, PLOT
E1(U1), IE1(I1), VRL(U2), IRL(I2), PLOT
VRL(U2), PLOT (IRL(I2))
```

In den o. a. Beispielen wird jede Variable einzeln auf eine Schnelldruckerseite geplottet. Bis zu 9 Größen können aber auch gleichzeitig über mehrere Seiten hinweg in ein Diagramm gezeichnet werden, jedoch nur in Abhängigkeit von TIME, und zwar indem man der Plotterzeichnung einen Namen gibt (max. 6 alphanumerische Zeichen):

$$\boxed{\text{variable1} [, \text{variable2} \dots], \text{PLOT} \text{plotname}} .$$

Es können mehrere Zeichnungen dieser Art durch verschiedene Plotternamen angefordert werden. Eine Umbenennung der Ausgabevariablen ist ebenfalls möglich. Da die Länge einer solchen Zeichnung bis zu 2000 Zeilen betragen kann, muß unter RUN CONTROLS die Anweisung

$$\boxed{\text{PLOT INTERVAL} = \text{number}}$$

eingefügt werden⁶.

Die Ergebnisse einer AC-Analyse werden normalerweise als Funktion der Frequenz ausgegeben.

$$\boxed{\text{variable1} [, \text{variable2} \dots], \left\{ \begin{array}{l} \text{DEGREES} \\ \text{RADIANS} \\ \text{COMPLEX} \\ \text{NYQUIST} \end{array} \right\} [, \text{PLOT}]}$$

Mit NYQUIST wird der Imaginärteil über dem Realteil aufgetragen, das Wort PLOT kann dabei entfallen. Mit DEGREES erhält man die Amplitude und den Phasenwinkel in Grad, mit RADIANS den Phasenwinkel im Bogenmaß und mit COMPLEX den Real- und Imaginärteil der entsprechenden Wechselstromgröße.

⁶siehe Kapitel 1.10

1.8. INITIAL CONDITIONS

Für Kapazitäten, Induktivitäten und Dioden-Stromquellen können Anfangsbedingungen gesetzt werden:

$$\left\{ \begin{array}{l} VC \\ IL \\ VJ \end{array} \right\} name = number .$$

SCEPTRE bietet zwei Möglichkeiten, die Anfangsbedingung für eine Netzwerksimulation anzugeben:

- Aufruf der Gleichstromanalyse (RUN IC). In diesem Fall werden die Anfangsbedingungen vom Programm berechnet. Die o. a. Anweisungen dienen dann nur als Anfangswerte für die Newton-Raphson-Iteration. Andernfalls beginnt die Iteration bei Null.
- Direkte Eingabe der Anfangswerte. Fehlen diese, werden alle Anfangswerte automatisch auf den Wert Null gesetzt.

1.9. FUNCTIONS

Unter dieser Überschrift werden alle Tabellen und Funktionen, die unter ELEMENTS und DEFINED PARAMETERS benötigt werden, beschrieben.

1.9.1. EQUATION-Anweisung

$$\left\{ \begin{array}{l} EQUATION \\ Q \end{array} \right\} name (Liste) = (arithmetische Anweisung)$$

Liste enthält Hilfsvariablen (Formalparameter), die maximal 6 Zeichen lang sein und nicht mit einer Zahl oder den Buchstaben I bis N beginnen dürfen⁷. Diese Hilfsvariablen werden in einer arithmetischen Anweisung entsprechend der in FORTRAN üblichen Formelfunktion miteinander verknüpft. Es können auch externe Unterprogramme aufgerufen werden. Alle numerischen Konstanten in der arithmetischen Anweisung müssen mit Dezimalpunkt geschrieben werden. Wird mit komplexen Größen gerechnet, müssen die Hilfsvariablen mit dem Buchstaben Z beginnen. Die Anzahl der Hilfsvariablen muß mit der Anzahl der Parameter im EQUATION-Aufruf übereinstimmen. Die Namen der Hilfsvariablen können in anderen EQUATION-Anweisungen wiederbenutzt werden.

Beispiele:

```
ELEMENTS
  E1, 1-0 = Q9 (220.,50.)
  ...
FUNCTIONS
  Q9 (A,B) = (A*DSQRT(2.) * DSIN (B*6.28 * TIME))
```

oder

```
DEFINED PARAMETERS
  WZE = Q107 (EIN,IEIN)
  ...
FUNCTIONS
  Q107 (Z1,ZB7X) = (Z1/ZB7X)
```

⁷wegen der Wahl der Variablenamen siehe Abschnitt 2.19, Seite 44

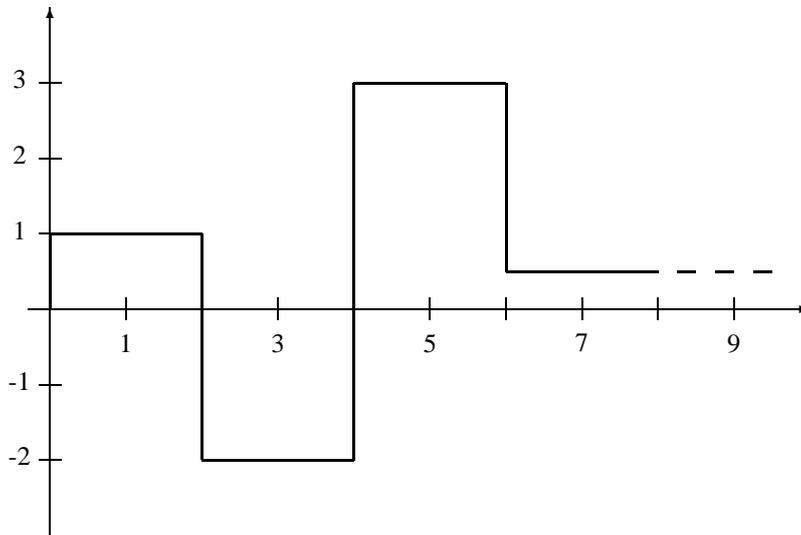


Abbildung 1.1.: Spannungsverlauf

1.9.2. TABLE-Anweisung

$$[\text{DIODE}] \left\{ \begin{array}{c} \text{TABLE} \\ \text{T} \end{array} \right\} \textit{name} \left\{ \begin{array}{c} = \\ \textit{neue} \\ \textit{Zeile} \end{array} \right\} \textit{number1}, \textit{number2}, \dots$$

Die Tabellenwerte werden paarweise, durch Kommata getrennt, eingegeben. Der erste Wert entspricht der unabhängigen, der zweite der abhängigen Variablen. Die Werte der unabhängigen Variablen müssen in monoton aufsteigender Reihenfolge vorliegen. Die Funktion muß eindeutig sein. Es können aber zu einem unabhängigen Wert zwei abhängige Werte angegeben werden, um Sprungfunktionen zu beschreiben (siehe Beispiel). Zwischenwerte in den Tabellen werden linear inter- oder extrapoliert⁸.

Beispiel: Beschreibung einer Spannungsfunktion nach Abbildung 1.1.

```
TABLE XYZ7
0,0, 0,1, 2,1, 2,-2, 4,-2, 4,3,
6,3, 6,0.5, 100,0.5
```

Wird die Tabelle als DIODE TABLE definiert, nimmt SCEPTRE an, daß der erste Wert der Diodenspannung, der zweite dem entsprechenden Diodenstrom entspricht.

⁸besonders wichtig für das letzte Wertepaar

1. SCEPTRE-Anweisungen

1.10. RUN CONTROLS

Zu diesem Kapitel gehören eine Reihe von Steuerkarten, mit denen sich der Rechenablauf einer Simulation in weiten Grenzen steuern läßt. Es stehen drei Analysearten zur Verfügung, die teilweise miteinander kombiniert werden können.

1.10.1. DC-Analyse

Soll nur eine Gleichstromanalyse durchgeführt werden, genügt die Anweisung

```
RUN INITIAL CONDITIONS ONLY .
```

In diesem Fall wird die DC-Analyse mit dem Newton-Raphson-Algorithmus durchgeführt. Die maximale Anzahl der Iterationen kann zusätzlich beeinflußt werden mit

```
NEWTON-RAPHSON PASS LIMIT = number (100).
```

Die Konvergenzkriterien für die Iteration können ebenfalls geändert werden durch die Anweisungen

```
RELATIVE CONVERGENCE = number (10-3)
```

```
ABSOLUTE CONVERGENCE = number (10-4).
```

Mit

```
RUN IC VIA IMPLICIT
```

wird anstatt des Newton-Raphson-Verfahrens ein impliziter Algorithmus aufgerufen.

Innerhalb der DC-Analyse stehen vier spezielle Verfahren zur Verfügung, die neben den Steuerkarten unter RUN CONTROLS zusätzlich noch Anweisungen unter CIRCUIT DESCRIPTION benötigen. Für MONTE-CARLO, OPTIMIZATION und WORST-CASE muß unter ELEMENTS bzw. DEFINED PARAMETERS mindestens eine Größe mit Toleranzgrenzen angegeben werden.

Unabhängig von der Reihenfolge der Steueranweisungen werden – falls verlangt – die Toleranzanalysen in der Reihenfolge

1. SENSITIVITY,
2. MONTE-CARLO,
3. WORST-CASE und
4. OPTIMIZATION

durchgeführt.

Sensitivity:

Es genügt die Anweisung

```
RUN SENSITIVITY .
```

Monte Carlo:

Der Aufruf lautet

```
RUN MONTE CARLO [= number] (10) .
```

Mit *number* kann die Anzahl der Iterationen variiert werden. Die Verteilung der Zufallszahlen wird gesteuert mit

```
DISTRIBUTION = { GAUSSIAN
                  UNIFORM }
```

Man kann die Anfangszufallszahl vorgeben mit

```
INITIAL RANDOM NUMBER = number (127263527) .
```

Diese Zahl sollte positiv, ungerade und neun Stellen lang sein (I-Format). Soll bei Wiederholungsläufen immer mit der gleichen Zufallszahl begonnen werden, erreicht man dies durch

```
INITIAL RANDOM NUMBER = DEFAULT .
```

Zusätzliche Informationen zu jeder Monte-Carlo-Iteration erhält man mit

```
LIST MONTE CARLO DETAILS .
```

Worst-Case:

Eine Worst-Case-Analyse erhält man mit

```
RUN WORST CASE [ = { LOW
                     NOMINAL
                     HIGH } ] .
```

Die Schlüsselwörter nach dem Gleichheitszeichen geben an, welche Werte als Anfangswerte für eine nachfolgende TR-Analyse eingesetzt werden sollen.

Optimization:

Diese Analyse wird mit

```
RUN OPTIMIZATION [= number] (30)
```

aufgerufen. Mit *number* kann die Anzahl der durchzuführenden Iterationen geändert werden.

Zwischenergebnisse erhält man mit

```
LIST OPTIMIZATION DETAILS .
```

1. SCEPTRE-Anweisungen

Mit den folgenden Steueranweisungen kann die Optimierung in weiten Grenzen beeinflusst werden:

`INITIAL H MATRIX FACTOR = number` (1.)

`OPTIMIZATION CRITERION = number` (10^{-7})

`OPTIMIZATION RANDOM STEPS = number` (0)

`RANDOM STEP SIZE CONTROL = number` (0.2)

`MINIMUM FUNCTION ESTIMATE = number` (0)

`PUNCH OPTIMIZATION RESULTS` .

Die Bedeutung und Handhabung dieser Anweisungen sind in [2] beschrieben.

1.10.2. AC-Analyse

Die AC-Analyse wird aufgerufen durch die Anweisung

`RUN AC` .

Soll das Netzwerk nur für eine Frequenz berechnet werden, genügt die Anweisung

`FREQUENCY = number` .

Für die Beschreibung eines Frequenzbereichs werden folgende Angaben benötigt:

`INITIAL FREQUENCY = number`

`FINAL FREQUENCY = number`

`NUMBER FREQUENCY STEPS = number` (10)

`TYPE FREQUENCY RUN = { $\frac{\text{LINEAR}}{\text{LOG}}$ }` .

Werden bei Wiederholungsläufen nur die Strom- und/oder Spannungsquellen variiert, spart man Rechenzeit mit der Anweisung

`USE FIXED AC MATRIX IN RERUNS` .

1.10.3. TR-Analyse

Für eine TR-Analyse ist *immer* die Angabe des Endzeitpunktes der Simulation erforderlich:

`STOP TIME = number` .

Weiterhin können folgende Größen gesteuert werden:

Anzahl der Integrationsschritte:

`MAXIMUM INTEGRATION PASSES = number` (20000)

Anfangszeitpunkt der TR-Analyse:

`START TIME = number` (0.0)

Auswahl des zu verwendenden Integrationsverfahrens:

`INTEGRATION ROUTINE = { XPO
TRAP
RUK
IMPLICIT }` .

Wird das implizite Integrationsverfahren gewählt, kann zusätzlich noch bestimmt werden:

`USE { DIFFERENCED
SYMBOLIC } JACOBIAN` .

Fehlt diese Anweisung, so wählt SCEPTRE automatisch den für das zu untersuchende Netzwerk besten Algorithmus aus.

SCEPTRE arbeitet mit variabler Integrationsrittweite, die nach jedem Integrationsschritt neu berechnet wird. Durch folgende Anweisungen können die Integrationsrittweiten variiert werden:

Integrationsverfahren:	XPO	TRAP	RUK	IMPLICIT
MINIMUM STEP SIZE = <i>number</i>	10^{-5}	10^{-5}	10^{-5}	10^{-14}
MAXIMUM STEP SIZE = <i>number</i>	2×10^{-2}	2×10^{-2}	2×10^{-2}	2×10^{-2}
STARTING STEP SIZE = <i>number</i>	10^{-3}	10^{-3}	10^{-3}	10^{-8}
	× STOP TIME			

Die Integrationsgenauigkeit kann durch weitere Anweisungen beeinflusst werden. Die Bedeutung dieser Parameter wird in [2] beschrieben.

Integrationsverfahren:	XPO	TRAP	RUK	IMPLICIT
MINIMUM ABSOLUTE ERROR = <i>number</i>	10^{-4}	5×10^{-5}	5×10^{-5}	10^{-4}
MAXIMUM ABSOLUTE ERROR = <i>number</i>	5×10^{-3}	10^{-3}	5×10^{-3}	-
MINIMUM RELATIVE ERROR = <i>number</i>	2×10^{-4}	5×10^{-4}	5×10^{-5}	-
MAXIMUM RELATIVE ERROR = <i>number</i>	5×10^{-3}	10^{-2}	5×10^{-3}	-

1. SCEPTRE-Anweisungen

1.10.4. DC- und TR-Analyse

Mit der Anweisung

```
RUN INITIAL CONDITIONS
```

wird zuerst eine DC-Analyse und anschließend eine TR-Analyse durchgeführt (STOP TIME nicht vergessen!). Die Ergebnisse der DC-Analyse werden als Anfangsbedingungen in die TR-Analyse eingesetzt⁹. Es können alle Steueranweisungen der DC- und TR-Analyse verwendet werden.

1.10.5. DC- und AC-Analyse

Es gilt sinngemäß dasselbe wie bei der **DC- und TR-Analyse**. Es können alle Steueranweisungen der DC- und AC-Analyse benutzt werden.

1.10.6. AC- und TR-Analyse

Diese Kombination ist nicht möglich.

1.10.7. Ausgabesteuerung

Die unter OUTPUTS angeforderten Ergebnisse liegen zu jedem Integrationsschritt vor. Die Anzahl der tatsächlich auszudruckenden Ergebnisse läßt sich mit

```
MAXIMUM PRINT POINTS = number (1000)
```

steuern. Unterdrücken der Druckausgabe erreicht man, indem man *number* gleich Null setzt.

Zusätzlich kann man sich die Ergebnisse zu bestimmten Zeitintervallen ausdrucken lassen:

```
PRINT INTERVAL = number .
```

Mehrfachplotterzeichnungen werden unter OUTPUTS angefordert. Die Länge dieser Zeichnungen in Zeilen wird durch

```
PLOT INTERVAL = number
```

gesteuert. Sie ergibt sich aus dem Quotienten STOP TIME / PLOT INTERVAL und kann maximal 2000 Zeilen betragen. Fehlt diese Anweisung, wird für jede Ausgabegröße eine einfache Plotterzeichnung angefertigt.

1.10.8. Wiederholungsläufe

Bei Wiederholungsläufen (siehe Kapitel 1.13) werden die Anfangsbedingungen jeweils neu berechnet. Sie können aber auch explizit für jeden Lauf angegeben werden. Es besteht außerdem die Möglichkeit, als Anfangsbedingung die Ergebnisse des 1. Laufs oder vorherigen Laufs einzusetzen, und zwar mit der Anweisung

```
IC FOR RERUNS = { MASTER } RESULTS .  
                  { PRECEDING }
```

⁹Es werden immer die Ergebnisse der *letzten* DC-Analyse als Anfangsbedingungen eingesetzt. Die Toleranzanalysen werden unabhängig von der Reihenfolge der Steueranweisungen in der Reihenfolge SENSITIVITY, MONTE-CARLO, WORST-CASE und OPTIMIZATION durchgeführt.

1.10.9. Ablaufsteuerung

Man kann eine Simulation in Abhängigkeit vom Netzwerkverhalten abbrechen oder unterbrechen. Dies geschieht mit der Anweisung

$$\left\{ \begin{array}{c} \text{TERMINATE} \\ \text{STOP} \end{array} \right\} \text{ IF } \left(\left\{ \begin{array}{c} \text{variable} \\ \text{constant} \end{array} \right\} \text{ .ro. } \left\{ \begin{array}{c} \text{variable} \\ \text{constant} \end{array} \right\} \right) .$$

Dabei steht *.ro.* für die Vergleichsoperatoren *.LE.*, *.LT.*, *.GT.*, *.GE.*, *.EQ.* und *.NE.*. Sie haben die gleiche Bedeutung wie in FORTRAN. Die Bedingungen können noch durch die logischen Operatoren AND und OR erweitert werden. Benutzt man TERMINATE, so wird der aktuelle Lauf abgebrochen, die evtl. noch anstehenden Wiederholungsläufe werden jedoch durchgeführt. Mit STOP wird der ganze Rechenlauf beendet. Beispiele:

```
TERMINATE IF (PWR .GT. 0.3)
STOP IF ((VRL .LE. E3) .OR. (IL9 .GT. 20.))
```

SCEPTRE schreibt während eines Rechenlaufs in regelmäßigen Zeitabständen interne Programmdateien in eine Datei. Endet ein Lauf abnorm (Systemzusammenbruch etc.), so braucht der Lauf nicht von Anfang an wiederholt zu werden, sondern kann mit den zuletzt gesicherten Programmdateien fortgesetzt werden (siehe Kapitel 1.14). Mit

$$\boxed{\text{COMPUTER SAVE INTERVAL} = \textit{number}} \quad (15)$$

kann das Zeitintervall geändert werden (Angaben in Minuten).

Eine weitere Vorsichtsmaßnahme bieten die Abfragen nach der bisher verbrauchten CPU-Zeit sowie der Verweilzeit des Programms im Rechner während der Ausführungsphase (elapsed time). Die maximale CPU-Zeit in Sekunden setzt man mit

$$\boxed{\text{SOLUTION TIME LIMIT} = \textit{number}} .$$

Wenn diese Zeit verbraucht ist, druckt und plottet das Programm alle bis zu diesem Zeitpunkt errechneten Ergebnisse und endet ordnungsgemäß. Es wird nur die Rechenzeit der Ausführungsphase gezählt. Die maximale Verweilzeit in Minuten ändert man mit

$$\boxed{\text{COMPUTER TIME LIMIT} = \textit{number}} \quad (600) .$$

Die Verweilzeiten von Programmen hängen weitgehend von der Recherauslastung ab. Wird die angegebene Zeit während der Ausführungsphase überschritten, werden alle bis zu diesem Zeitpunkt vorhandenen Ergebnisse ausgegeben und das Programm ordnungsgemäß abgeschlossen.

Sollen die SCEPTRE-Anweisungen nur verarbeitet und nicht ausgeführt werden, z. B. zwecks Syntaxprüfung, erreicht man das mit

$$\boxed{\text{EXECUTE SETUP PHASE ONLY}} .$$

1. SCEPTRE-Anweisungen

1.10.10. Testhilfen

Eine Übersicht aller Netzwerkknoten und dazugehörigen Zweige erhält man durch

```
LIST NODE MAP .
```

Die von SCEPTRE generierten FORTRAN-Programme werden ausgedruckt mit der Anweisung

```
WRITE SIMUL8 DATA .
```

Die Netzwerkmatrix, die die Topologie des Netzwerks beschreibt, erhält man mit

```
PRINT B MATRIX .
```

Bei einer AC-Analyse lassen sich mit den folgenden Anweisungen weitere Informationen ausdrucken:

```
PRINT A MATRIX
```

```
PRINT EIGENVALUES
```

```
PRINT EIGENVECTORS .
```

1.11. SENSITIVITY, MONTE CARLO, WORST CASE, OPTIMIZATION

Die Toleranzanalysen werden durch Anweisungen gleichen Namens aufgerufen.

```
SENSITIVITY
```

```
MONTE CARLO
```

```
WORST CASE
```

```
OPTIMIZATION
```

Jeder dieser Anweisungen folgt eine Parameterliste in der Form

```
(abhängige Variablenliste / unabhängige Variablenliste) .
```

Die abhängigen Variablen definieren die Zielfunktion. Mehrere Variablen werden durch Kommata getrennt. Es müssen mindestens eine abhängige und eine unabhängige Variable vorhanden sein. Für die Monte-Carlo-, Worst-Case- und Optimization-Analysen müssen die unabhängigen Variablen mit Toleranzgrenzen angegeben werden. Tabellen 1.6 und 1.7 zeigen die Größen, die als abhängige und unabhängige Variablen auftreten können. Wird *Pname* als abhängige Variable in der Sensitivity-, Optimization-, oder Worst-Case-Analyse benutzt, muß das vollständige Differential *GPname* ebenfalls angegeben werden. *Pname* als unabhängige Variable kann nur als Faktor in einer Stromquelle 2. Ordnung auftreten, z. B. $Jname1 = Pname * Jname2$. Beispiel:

<i>VCname</i>	Spannung an einem Kondensator
<i>IEname</i>	Strom durch eine unabhängige Spannungsquelle
<i>Jname</i>	abhängige Stromquelle 1. Ordnung
<i>VJname</i>	Spannung an einer unabhängigen Stromquelle
<i>ILname</i>	Strom durch eine Induktivität
<i>VRname</i>	Spannung an einem Widerstand
<i>IRname</i>	Strom durch einen Widerstand
<i>Pname</i>	definierter Parameter

Tabelle 1.6.: Abhängige Variablen

<i>Rname</i>	Widerstand
<i>Ename</i>	unabhängige Spannungsquelle
<i>Jname</i>	unabhängige Stromquelle
<i>Pname</i>	definierter Parameter

Tabelle 1.7.: Unabhängige Variablen

SENSITIVITY

(IL3 , PX / IX , EY)
 (VR1 , IR1 , VCX / P1 , E1)

Es werden folgende partiellen Ableitungen berechnet:

$$\frac{\partial IL3}{\partial IX}, \frac{\partial IL3}{\partial EY}, \frac{\partial PX}{\partial IX}, \frac{\partial PX}{\partial EY}$$

$$\frac{\partial VR1}{\partial P1}, \frac{\partial VR1}{\partial E1}, \frac{\partial IR1}{\partial P1}, \frac{\partial IR1}{\partial E1}, \frac{\partial VCX}{\partial P1}, \frac{\partial VCX}{\partial E1}$$

Beispiel:

OPTIMIZATION

(IR1 , P1 / R1 , P2 , E1)
 (VC1 / I1 , R2)

IR1 wird optimiert bezogen auf R1, P2 und E1.

P1 wird optimiert bezogen auf R1, P2 und E1.

VC1 wird optimiert bezogen auf I1 und R2.

1.12. Modelle

Netzwerkteile oder Ersatzschaltbilder, die häufig benutzt werden, können permanent oder temporär gespeichert werden und bei Bedarf in das zu untersuchende Netzwerk eingefügt werden. Dabei ist es möglich, einzelne Parameter im Modell zu ändern. Die Modellbeschreibung erfolgt unter der Überschrift

```
MODEL DESCRIPTION [ ( [ INITIAL , ] PRINT ) ]
```

und geht der Netzwerkbeschreibung voraus. Werden keine Modelle benutzt, kann dieser Paragraph entfallen. INITIAL wird nur dann benötigt, wenn das *erste Modell permanent* gespeichert werden soll. PRINT veranlaßt, die Modellbeschreibung auszudrucken. Die Befehle PRINT und INITIAL sind nur für permanent gespeicherte Modelle erlaubt.

1. SCEPTRE-Anweisungen

1.12.1. Modellbeschreibung

Folgende Anweisungen sind für eine Modellbeschreibung erlaubt:

```
MODEL modelname [ ( { PERM } ) ] ( node1 - node2 - ... )
```

PERM bedeutet, daß das Modell permanent gespeichert werden soll. Mit TEMP wird es am Ende der Simulation wieder gelöscht. *modelname* kann maximal 18 Zeichen lang sein. Ein Modell darf bis zu 25 Anschlußpunkte (*node1* - *node2* - ...) haben. Mit

```
MODEL modelname (DELETE)
```

wird ein permanent gespeichertes Modell gelöscht. Nach der MODEL Anweisung können bis zu 11 Kommentarzeilen folgen. Danach sind Anweisungen zu folgenden Überschriften möglich:

- ELEMENTS
- DEFINED PARAMETERS
- OUTPUTS
- FUNCTIONS
- INITIAL CONDITIONS.

Für die dazugehörigen Anweisungen gelten die gleichen Regeln wie unter CIRCUIT DESCRIPTION. Beispiel:

```
MODEL DESCRIPTION (INITIAL,PERM)
MODEL 2N1734B (PERM) (B-E-C)
ELEMENTS
  CE, B-E = Q1 (5., 80., TABLE 1 (VCE))
  CC, B-C = Q1 (10., 200., TABLE 2 (VCC))
  J1, B-E = DIODE TABLE 1
  JA, E-B = 0.1 * J2
  J2, B-C = DIODE TABLE 2
  JB, C-B = 0.98 * J1
OUTPUTS
  VCE, VCC, J1, PLOT
FUNCTIONS
  DIODE TABLE 1 = 0,0, .3,0, .65,.5, ....
  DIODE TABLE 2 = 0,0, .58,0, .62,.4, ....
  EQUATION 1 (A,B,C) = (A + B * C)
```

Ob es sinnvoll ist, die OUTPUTS-Anweisung in die Modellbeschreibung einzufügen, muß man von Fall zu Fall entscheiden.

1.12.2. Modellaufruf

Ein Modell wird unter CIRCUIT DESCRIPTION/ELEMENTS mit folgender Anweisung in das Netzwerk eingefügt:

$$name, node1 - node2 - \dots = MODEL modelname [(\left\{ \begin{array}{c} PERM \\ TEMP \end{array} \right\})] \dots$$

$$\dots [, CHANGE \left\{ \begin{array}{c} element \\ def.-par. \end{array} \right\} = value [, \left\{ \begin{array}{c} \dots \\ \dots \end{array} \right\} = value]] \dots$$

$$\dots [, SUPPRESS \left\{ \begin{array}{c} var1, var2, \dots \\ ALL \end{array} \right\}] [PRINT]) .$$

PERM oder TEMP gibt an, ob es sich um ein permanent oder temporär gespeichertes Modell handelt. *modelname* ist der Name des Modells, unter dem es gespeichert oder definiert wurde.

Alle internen Modellnamen (Elemente, Knotenpunkte, usw.) werden um den Bezugsnamen *name* erweitert (siehe folgendes Beispiel). Damit ist jeder interne Modellname eindeutig gekennzeichnet, auch wenn das gleiche Modell mehrere Male eingefügt wird. SCEPTRE behandelt dann dieses Modell so, als ob es direkt unter CIRCUIT DESCRIPTION definiert worden wäre. Da unter CIRCUIT DESCRIPTION die Namenslänge begrenzt ist, empfiehlt es sich, die im Modell internen, frei wählbaren Namen sowie den Bezugsnamen möglichst kurz zu wählen entsprechend der Angaben in Tabelle 1.2. Beispiel: Aufruf des o. a. Modells 2N1734B:

```
T1, 13-4-GND = MODEL 2N1734B (PERM)
```

Will man sich auf eine Größe in dem eingefügten Modell T1 beziehen, muß diese Größe um den Bezugsnamen T1 erweitert werden, z.B. CET1, CCT1, J1T1, ..., VCET1, ICET1, usw..

Mit CHANGE können die Werte von Netzwerkelementen und definierten Parametern in dem eingefügten Modell geändert werden. Die folgenden Beispiele beziehen sich wieder auf das Modell 2N1734B.

```
T1, 13-4-GND = MODEL 2N1734B (PERM, CHANGE CC=50)
T1, 13-4-GND = MODEL 2N1734B (PERM, CHANGE CC=TABLE 7(VCCT1),
    JA=0.2*J2T1, JB=DIODE TABLE 4)
```

Die Werte für die Kapazität CC ergeben sich aus einer neuen Tabelle (TABLE 7), die Stromquelle JB wird mittels einer Diodenkennlinie berechnet (DIODE TABLE 4). Beide Tabellen müssen unter CIRCUIT DESCRIPTION/FUNCTIONS definiert sein. Die gesteuerte Stromquelle JA wird mit einer anderen mathematischen Anweisung berechnet.

```
T1, 13-4-GND = MODEL 2N1734B (PERM, CHANGE CC=EQUATION 5 (VCCT1, VCX))
```

Die Kapazitätswerte für CC werden durch eine neue Formelfunktion (EQUATION 5) berechnet, die unter CIRCUIT DESCRIPTION/FUNCTIONS definiert sein muß. Dabei werden als Parameter die Spannungen an den Kondensatoren CC und CX übergeben. Der Kondensator CX gehört nicht zu dem Modell 2N1734B.

```
T1, 13-4-GND = MODEL 2N1734B (PERM, CHANGE TABLE 1 = TABLE 7)
```

1. SCEPTRE-Anweisungen

Die Tabelle 7 ersetzt die Tabelle 1 in dem Modell und muß unter CIRCUIT DESCRIPTION/FUNCTIONS definiert werden. Die Anzahl der Punktepaare in beiden Tabellen braucht nicht übereinzustimmen¹⁰.

```
T1, 13-4-GND = MODEL 2N1734B (PERM, CHANGE Q1=Q2)
```

Es wird die Funktion EQUATION 1 durch eine neue Funktion EQUATION 2 ersetzt, die unter CIRCUIT DESCRIPTION/FUNCTIONS definiert sein muß. Die neue Funktion muß die gleichen Parameter haben¹¹.

Die Ausgabe von Ergebnissen, die man unter MODEL DESCRIPTION/OUTPUTS anfordern kann, läßt sich ganz oder teilweise mit dem Befehl SUPPRESS unterdrücken.

```
T1, 13-4-GND = MODEL 2N1734B (PERM, SUPPRESS J1)
T1, 13-4-GND = MODEL 2N1734B (PERM, SUPPRESS ALL)
```

Mit dem Befehl PRINT wird die Modellbeschreibung aufgelistet.

```
T1,13-4-GND=MODEL 2N1734B(PERM,CHANGE CC=50,CE=30,SUPPRESS J1,VCE,PRINT)
```

Die beim Modellaufruf möglichen Befehle können miteinander kombiniert werden.

Anfangsbedingungen setzt man vorteilhafter unter CIRCUIT DESCRIPTION/INITIAL CONDITIONS.

```
VCCT1 = 0.53
```

1.13. RERUN DESCRIPTION

SCEPTRE bietet die Möglichkeit, einen bereits durchgeführten Lauf mit veränderten Parametern zu wiederholen. Dabei brauchen nur noch die sich ändernden Werte angegeben zu werden. Diese Änderungen werden unter der Überschrift

```
RERUN DESCRIPTION [ ( n ) ]
```

zusammengefaßt. Die Zahl *n* gibt die Anzahl der modifizierten Rechenläufe an. Der Klammerausdruck entfällt bei nur einem Wiederholungslauf. Im Gegensatz zur CIRCUIT DESCRIPTION dürfen bei der RERUN DESCRIPTION nur die Werte *einer* Variablen pro Eingabezeile geändert werden.

Die konstanten Variablen unter ELEMENTS ändert man mit

```
elementname = number1, number2, ... .
```

Im 1. Wiederholungslauf wird *elementname* der Wert *number1*, im 2. Lauf der Wert *number2* usw. zugewiesen.

Entsprechend werden den konstanten Variablen unter DEFINED PARAMETERS neue Werte zugeordnet.

```
def. par. name = number1, number2, ...
```

Beispiel:

¹⁰Achtung: es ist nicht möglich, eine Diodenkennlinie direkt in eine andere Diodenkennlinie abzuändern (CHANGE DIODE TABLE 1 = DIODE TABLE 9).

¹¹Achtung: es ist nicht möglich, in einem Modell eine Funktion in eine Tabelle oder umgekehrt abzuändern (CHANGE EQUATION 1 = TABLE 4). Es können dagegen Netzwerkelemente und definierte Parameter durch Konstanten, Tabellen oder Funktionen ersetzt werden.

```
RERUN DESCRIPTION (3)
ELEMENTS
  RLOAD = 25, 200, 20E3
DEFINED PARAMETERS
  PSI = 13.4, 13.2, 13.1
```

Ebenso können Toleranzgrenzen neu definiert werden.

$$\left\{ \begin{array}{l} \text{def. par. name} \\ \text{element name} \end{array} \right\} = \text{number1} \left(\left\{ \begin{array}{l} \text{number2, number3} \\ \text{number4} \end{array} \right\} \right), \dots$$

Beispiel:

```
R3 = 4.6, 9.(8.,10.5)
```

Im 1. Lauf wird dem Widerstand R3 der Wert 4.6 ohne Toleranzgrenzen zugewiesen, im 2. Lauf der Nominalwert 9. mit der Unter- und Obergrenze von 8. bzw. 10.5.

Die Anfangsbedingungen unter INITIAL CONDITIONS können geändert werden mit

$$\left\{ \begin{array}{l} \text{VC} \\ \text{IL} \\ \text{VJ} \end{array} \right\} \text{name} = \text{number1, number2, ...}$$

Unter FUNCTIONS können die Tabellenwerte geändert werden. Die neue Tabelle muß die gleiche Länge besitzen wie die ursprüngliche Tabelle. Pro RERUN DESCRIPTION-Anweisung können die unabhängigen (x-Werte) einer Tabelle nur einmal überschrieben werden. Die dazugehörenden Tabellenwerte (y-Werte) können beliebig oft geändert werden. Die allgemeine Anweisung lautet:

$$\text{TABLE name} = X1, Y11, Y12, \dots Y1N, X2, Y21, Y22, \dots Y2N, \dots$$

Der Index *N* entspricht der Anzahl der gewünschten Wiederholungsläufe.

Beispiel:

```
CIRCUIT DESCRIPTION
.....
FUNCTIONS
  TABLE 7 = 0,0, 1,10, 2,27, 4,47
.....
RERUN DESCRIPTION (2)
FUNCTIONS
  TABLE 7 = 0,0,0, 1,3,5, 3,4,8, 5,7,9
```

Tabelle 7 wird dann folgendermaßen belegt:

Originallauf		1. Wiederholung		2. Wiederholung	
X	Y	X	Y	X	Y
0	0	0	0	0	0
1	10	1	3	1	5
2	27	3	4	3	8
4	47	5	7	5	9

1. SCEPTRE-Anweisungen

Man erkennt, daß die x-Werte der Tabelle im 2. Wiederholungslauf die gleichen Werte aufweisen wie im ersten Lauf. Will man die x-Werte bei der 2. Wiederholung ebenfalls ändern, muß man eine zweite RERUN DESCRIPTION Überschrift einfügen.

Beispiel:

```
RERUN DESCRIPTION
FUNCTIONS
TABLE 7 = 0,0, 1,3, 3,4, 5,7
RERUN DESCRIPTION
FUNCTIONS
TABLE 7 = 0,0, 2,4, 3,5, 4,9
```

Unter RUN CONTROLS können alle Anweisungen geändert werden mit Ausnahme von STOP und TERMINATE. Bezüglich der Wahl der Analysearten gibt es einige Einschränkungen:

- Die Topologie eines Netzwerkes kann nicht geändert werden.
- Die Angaben unter OUTPUTS können nicht geändert werden.
- Konstanten müssen durch Konstanten, Tabellen durch Tabellen und EQUATION-Anweisungen durch EQUATION-Anweisungen ersetzt werden.
- EXPRESSION- und EQUATION-Anweisungen können nur dann geändert werden, wenn die in ihnen enthaltenen Variablen unter DEFINED PARAMETERS definiert werden. Man ändert dann diese Parameter.

Beispiel:

```
CIRCUIT DESCRIPTION
ELEMENTS
JE, 1-8 = DIODE EQUATION (PX1,PX2)
.....
DEFINED PARAMETERS
PX1 = ....., PX2 = .....
.....
RERUN DESCRIPTION
DEFINED PARAMETERS
PX1 = .....
PX2 = .....
```

- Die Änderungen von Netzwerkelementen erfordern u. U. eine Neuberechnung der Anfangsbedingungen.
- Im allgemeinen kann die Analyseart nicht geändert (gewechselt) werden.
- Es werden keine neuen Anfangsbedingungen berechnet, auch wenn sie im ursprünglichen Lauf durchgeführt wurden. Um dies auch für die Wiederholungsläufe zu erreichen, muß unter RUN CONTROLS die Anweisung RUN INITIAL CONDITIONS erneut angegeben werden. Es werden dann bei *jedem* Wiederholungslauf die Anfangsbedingungen berechnet.
- Folgende Steueranweisungen sind in Wiederholungsläufen nur gültig, wenn die Anfangsbedingungen im ursprünglichen Lauf berechnet wurden:

- RUN SENSITIVITY
- RUN MONTE CARLO

1.13. RERUN DESCRIPTION

- RUN WORST CASE
 - RUN OPTIMIZATION
 - RUN INITIAL CONDITIONS
 - RUN INITIAL CONDITIONS ONLY.
- Die folgenden Anweisungen sind in Wiederholungsläufen nicht erlaubt:
 - NO ELEMENT SORT
 - PUNCH PROGRAM
 - WRITE SIMUL8 DATA
 - VECTOR EQUATIONS
 - WRITE DEBUG
 - PRINT B MATRIX
 - LIST NODE MAP
 - IC FOR RERUNS
 - USE DIFFERENCED JACOBIAN
 - USE SYMBOLIC JACOBIAN
 - EXECUTE SETUP PHASE ONLY
 - USE FIXED AC MATRIX IN RERUNS
 - IMPULSE RESPONSE BUFFER
 - INPUT FUNCTION BUFFER.
- Folgende Anweisungen werden vom ursprünglichen Lauf nicht mit in den Wiederholungslauf übernommen, sie müssen bei Bedarf neu angegeben werden:
 - RUN AC
 - RUN SENSITIVITY
 - RUN MONTE CARLO
 - RUN WORST CASE
 - RUN OPTIMIZATION
 - LIST MONTE CARLO DETAILS
 - LIST OPTIMIZATION DETAILS
 - PUNCH OPTIMIZATION RESULTS
 - OPTIMIZATION RANDOM STEPS
 - PRINT A MATRIX
 - PRINT EIGENVALUES
 - MINIMUM FUNCTION ESTIMATE
 - INITIAL RANDOM NUMBER.
- Folgende Anweisungen widersprechen sich und führen zu einem Programmabbruch:

1. SCEPTRE-Anweisungen

```
CIRCUIT DESCRIPTION
.....
IC FOR RERUNS = MASTER RESULTS
oder
IC FOR RERUNS = PRECEDING RESULTS
RERUN DESCRIPTION
.....
RUN INITIAL CONDITIONS [ ONLY ]
oder
RUN IC VIA IMPLICIT
```

Für Änderungen in Modellen gelten die gleichen Regeln. Es ist nur darauf zu achten, daß die Elementnamen um den Bezugsnamen erweitert werden.

Beispiel:

```
MODEL DESCRIPTION
MODEL ABCDEFG (PERM) (1-7-8-9)
ELEMENTS
.....
R1, 1-2 = 1
DEFINED PARAMETERS
P1 = 2.7
FUNCTIONS
TABLE 1 = ...
CIRCUIT DESCRIPTION
ELEMENTS
...
T8, 4-9-1-2 = MODEL ABCDEFG
...
RERUN DESCRIPTION
ELEMENTS
R1T8 = 0.98
DEFINED PARAMETERS
P1T8 = 3.1
FUNCTIONS
TABLE 1 T8 = ...
```

1.14. CONTINUE

Die CONTINUE-Anweisung erlaubt dem Benutzer, einen ordnungsgemäß abgeschlossenen Rechenlauf fortzusetzen¹². Sie kann nur auf TR-Analysen angewandt werden. Bei Wiederholungsläufen wird der letzte durchgeführte Lauf fortgesetzt. Eine Änderung im zu untersuchenden Netzwerk ist nicht möglich. Es können nur Anweisungen unter RUN CONTROLS geändert werden, und zwar:

- STOP TIME
- COMPUTER TIME LIMIT

¹²siehe auch Kapitel 1.10: Ablaufsteuerung

- MAXIMUM INTEGRATION PASSES
- INTEGRATION ROUTINE
- USE ...JACOBIAN
- MINIMUM/MAXIMUM STEP SIZE
- STARTING STEP SIZE
- MINIMUM/MAXIMUM ABSOLUTE ERROR
- MINIMUM/MAXIMUM RELATIVE ERROR
- MAXIMUM PRINT POINTS
- PRINT INTERVAL
- PLOT INTERVAL

Beispiel:

```
CONTINUE
  RUN CONTROLS
    STOP TIME = 200
END
```

1.15. RE-OUTPUT

Eine weitere Möglichkeit, Rechenzeit zu sparen, bietet die RE-OUTPUT-Anweisung. Sie erlaubt es, zusätzliche Ausdruckkopien zu einem vorher abgeschlossenen Rechenlauf zu erzeugen, ohne diesen Lauf noch einmal zu wiederholen. Dabei kann die Ausgabe der Ergebnisse in Grenzen gesteuert werden. Unter OUTPUTS wird eine Liste der Ausgabevariablen angegeben, die ausgedruckt bzw. geplottet werden sollen. Diese Liste kann nur Ausgabevariablen enthalten, die im ursprünglichen Lauf unter OUTPUTS angefordert wurden¹³. Zusätzlich kann unter RUN CONTROLS die Anzahl der Ausgabepunkte mit MAXIMUM PRINT POINTS gesteuert werden.

Bei der AC-Analyse ist außerdem darauf zu achten, daß Plotterzeichnungen nur in den Einheiten ausgegeben werden können¹⁴, die im ursprünglichen Lauf angefordert wurden.

Beispiel:

```
RE-OUTPUT
  OUTPUTS
    VR1, VR2
    VL1, VCC, PLOT
  RUN CONTROLS
    MAXIMUM PRINT POINTS = 1500
END
```

1.16. END

Die Anweisung END markiert das Ende der SCEPTRE Eingabe und muß als letzte Anweisung stehen.

¹³unabhängig von der Art der Ausgabe

¹⁴RADIANS, DEGREES, COMPLEX

2. Allgemeine Hinweise

Dieses Kapitel enthält wichtige Hinweise zur Benutzung von SCEPTRE. Außerdem wird auf mögliche Fehlerquellen und deren Ursache eingegangen.

2.1. Fehlerdiagnose

SCEPTRE verfügt über eine umfangreiche Fehlerdiagnose. Die Fehlermeldungen werden in 3 Klassen eingeteilt:

warning only: es handelt sich dabei im allgemeinen um unkritische Fehler, z. B. Syntaxfehler, die SCEPTRE berichtigen kann; die Simulation wird ausgeführt.

simulation deleted: Fehler dieser Kategorie verhindern eine anschließende Simulation; die Eingabedaten werden jedoch weiter geprüft.

execution terminated: bei schwerwiegenden Fehlern wird die Fehlerdiagnose abgebrochen.

Neben Syntaxfehlern und elementaren Verstößen gegen die Eingabestruktur werden auch Fehler aufgedeckt, die sich aus der Gesamtheit der Schaltung ergeben, z. B. "computational delay", unzulässige Netzwerktopologien, etc.. Im allgemeinen sind die Fehlernachrichten so gezielt, daß man unschwer die Ursache entdecken kann. Es gibt aber auch Fälle, bei denen eine unrichtige Schreibweise (z. B. die Null statt des Buchstabens O) eine Flut von verwirrenden Fehlermeldungen auslösen kann.

Neben diesen Eingabefehlern gibt es auch logische Fehler, die man erst anhand der Ergebnisse erkennt. Hier kann SCEPTRE keine Hilfe bieten¹.

2.2. Einige Grundregeln

- Physikalische Einheiten sollten so gewählt werden, daß sich die maximalen und minimalen Werte um nicht mehr als 5–6 Zehnerpotenzen unterscheiden und sich in der Nähe von 1 bewegen. Weit auseinander liegende Werte verschlechtern oder verhindern sogar die Konvergenz der iterativen Verfahren.
- Werden Modelle benutzt (z. B. aus einer permanenten Bibliothek), muß unbedingt kontrolliert werden, ob das Modell und die Schaltung, in die das Modell eingefügt werden soll, das gleiche Einheitensystem verwenden.
- Bei größeren bzw. komplexen Schaltungen empfiehlt es sich, die Schaltung schrittweise in Betrieb zu nehmen und anhand von Einzelsimulationen die Richtigkeit der Modelle und Ersatzschaltungen zu überprüfen.
- Nichtlinearitäten sollten möglichst durch gesteuerte Quellen beschrieben werden statt passive Elemente (R,C,L) zu benutzen.

¹Es stehen jedoch eine Reihe von Anweisungen zur Verfügung, die in manchen Situationen hilfreich sein können (siehe Kapitel 1.10: Testhilfen).

2.3. Zeitabhängige Induktivitäten und Kapazitäten

Der verwendete Lösungsansatz liefert falsche Resultate, wenn bei der TR-Analyse Induktivitäten bzw. Kapazitäten von der Zeit abhängen (siehe auch [3, Kapitel 1.7]). Gilt für eine Kapazität $C = f(t)$, muß eine Stromquelle $J = V_C \frac{dC}{dt}$ dem Kondensator parallel geschaltet werden. Für eine zeitabhängige Induktivität $L = f(t)$ korrigiert eine in Reihe geschaltete Spannungsquelle $E = I_L \frac{dL}{dt}$ den Fehler. Zeitabhängige Widerstände, stromabhängige Induktivitäten und spannungsabhängige Kapazitäten werden richtig berechnet.

2.4. Tabellen

Werden Tabellen zur Beschreibung von Nichtlinearitäten benutzt, empfiehlt es sich, die Kurven möglichst zu glätten bzw. zusätzliche Stützstellen einzufügen, um Knicke zu vermeiden. Besonders die ersten und letzten Wertepaare sind kritisch, da SCEPTRE außerhalb liegende Werte aus diesen Wertepaaren extrapoliert. Beschreibt man passive Elemente (R,C,L) mit Tabellen, ist darauf zu achten, daß diese für *keinen* unabhängigen Wert Null werden dürfen. Grundsätzliche sollte man geschlossene Funktionen Tabellen vorziehen.

Werden Tabellen innerhalb einer EXPRESSION- oder EQUATION-Anweisung verwendet, muß eine andere Syntax benutzt werden:

```
..... = X4711 ( ..... XTABLE(Tname,variable) ..... )
```

Für name wird der Tabellename eingesetzt, die unabhängige Variable *muß* als zweiter Parameter angegeben werden, z. B. TIME.

2.5. Arithmetische Anweisungen

Verwendet man arithmetische Anweisungen zur Beschreibung abhängiger Elemente oder Parameter, sollte man den Gültigkeitsbereich der Formeln überprüfen. Die Standardfunktionen MIN, MAX, ABS etc. in FORTRAN sind z. B. gut geeignet, Divisionen durch Null oder Überläufe zu verhindern. Beispiele:

```
CC, B-5 = X1 (PX/(0.8-MIN(0,VCC))**0.33333)
JD, C-B = X2 (PS*EXP(MIN(88,PQ*VJD))-1)
```

2.6. Topologische Einschränkungen

Enthält ein Netzwerk Maschen, bestehend aus Spannungsquellen und Kapazitäten, bzw. Knotenpunkte, die nur über Stromquellen und Induktivitäten erreichbar sind, muß bei variablen Quellen die erste zeitliche Ableitung vorgegeben werden.

2.6.1. DC-, AC- und TR-Analyse

Bei keiner der drei Analysearten dürfen in dem Netzwerk Maschen vorhanden sein, die ausschließlich aus Spannungsquellen bestehen (*voltage source loop*). Ebenso sind Knotenpunkte verboten, die nur über Stromquellen zu erreichen sind (*all-current source cut set*).

2. Allgemeine Hinweise

2.6.2. DC-Analyse

Sollen Anfangsbedingungen berechnet werden, dürfen in dem Netzwerk keine Maschen auftreten, die nur aus Spannungsquellen und Induktivitäten bestehen. Ebensovienig sind Knotenpunkte zugelassen, die nur über Kapazitäten und Stromquellen zu erreichen sind. Dem Problem im letzten Fall kann man dadurch begegnen, daß man den fraglichen Knotenpunkt über einen Widerstand, der das Verhalten der Schaltung nicht beeinflußt, z. B. mit Masse verbindet.

Man kann diese Einschränkung umgehen, indem man die Anfangsbedingungen nicht über eine Gleichstromanalyse sondern über eine Transientanalyse berechnet. Die notwendige Syntax lautet dann:

```
RUN INITIAL CONDITIONS [ ONLY ]  
RUN IC VIA IMPLICIT
```

Man kann dieses Verfahren auch anwenden, falls die Gleichstromanalyse nicht konvergiert (siehe Kapitel 2.17).

2.7. "Computational Delay"

Eine der überragenden Fähigkeiten von SCEPTRE besteht in der Modellierung von Nichtlinearitäten. Dies wird u. a. dadurch ermöglicht, daß der Benutzer jedes Netzwerkelement in Abhängigkeit von anderen Größen beschreiben kann. SCEPTRE berechnet zu jedem Integrationsschritt die zeitlichen Ableitungen der unabhängigen Variablen (Zustandsvariablen) und daraus mit den Netzwerkgleichungen die anderen Netzwerkvariablen. Voraussetzung ist dafür allerdings, daß die unabhängigen Variablen zu Beginn eines Integrationsschrittes bekannt sind. Andernfalls verwendet SCEPTRE den Wert der unabhängigen Variablen vom vorherigen Integrationsschritt. Es wird jedoch eine Warnung

```
THE TERM '.....' WILL CAUSE A COMPUTATIONAL DELAY
```

ausgegeben. Die Größe des Fehlers läßt sich schwer abschätzen und ist von dem Grad der Nichtlinearität der Funktion und der Integrationsschrittweite abhängig. Dieser Integrationsfehler tritt nicht bei Abhängigkeiten von der Zeit, Spannungen an Kondensatoren und Strömen durch Induktivitäten auf, da diese Größen immer unabhängige Variablen sind. Auch Widerstandsspannungen und -ströme können unabhängig sein, vorausgesetzt, daß sie mit der richtigen Syntax² definiert werden.

Dieses Problem tritt z. B. bei Diodenstromquellen auf, die spannungsgesteuert sind. Durch Parallelschaltung einer Kapazität — die ja in Wirklichkeit auch vorhanden ist — bildet die Kondensatorspannung die unabhängige Variable.

Falls diese Methode versagt, kann man die von SCEPTRE generierten FORTRAN Programme untersuchen. Mit der Anweisung `WRITE SIMUL8 DATA` unter `RUN CONTROLS` druckt SCEPTRE das Unterprogramm `SIMUL8` aus. Dem Listing kann man die Reihenfolge entnehmen, in der die abhängigen Variablen berechnet werden. Unter Umständen ist es möglich, die in Frage kommende Größe durch Netzwerkvariablen zu beschreiben, die zu diesem Integrationszeitpunkt bereits bekannt sind (siehe auch Kapitel 2.13). Lautet die Beschreibung unter `ELEMENTS` z. B.

```
L2, ... = X1 (P1*IR2)
```

und erhält man die Warnung

```
THE TERM 'IR2' WILL CAUSE A COMPUTATIONAL DELAY
```

²siehe Kapitel 1.5: Gesteuerte Quellen

kann man diese Warnung vermeiden, in dem man z. B. L2 als Funktion von VR2 definiert, vorausgesetzt, VR2 ist zu diesem Zeitpunkt bereits berechnet worden:

$$L2, \dots = X1 (P1*VR2/R2)$$

2.8. Wahl des Integrationsverfahrens

SCEPTRE bietet wahlweise drei explizite Integrationsverfahren an: das Exponentialverfahren (XPO), das Runge-Kutta-Verfahren 4. Ordnung (RUK) und ein modifiziertes Trapezverfahren (TRAP), sowie ein implizites (IMPLICIT). Das Exponentialverfahren, eine Eigenentwicklung von IBM, stellt einen Kompromiß zwischen Schnelligkeit und Genauigkeit dar. Es wird deshalb automatisch benutzt, falls der Benutzer kein Verfahren angibt. Das implizite Verfahren wird vorteilhaft immer dann angewandt, wenn in einem Netzwerk gleichzeitig sehr große und sehr kleine Zeitkonstanten auftreten.

Jedes Integrationsverfahren hat für die Schrittweitenbegrenzung und die Fehlerabschätzung einen Satz eigener Parameter³. Nach jedem Integrationsschritt wird der Fehler abgeschätzt und daraus die neue Schrittweite berechnet. Wird dabei die unterste Schranke unterschritten, bricht das Programm ab mit der Fehlermeldung

SMALLER MINIMUM STEP SIZE REQUIRED

und dem Index der verursachenden Zustandsvariablen⁴. Diese Fehler treten auf, wenn die Zeitkonstanten des Netzwerkes sehr viel kleiner sind als die gesamte Simulationszeit (STOP TIME). Man kann den Fehler dadurch beheben, daß man

- die Simulationszeit verkleinert,
- die Elemente, die zur Bildung der Zeitkonstanten beitragen, überprüft und die Zeitkonstanten evtl. vergrößert,
- die minimale Schrittweite verkleinert (MINIMUM STEP SIZE), voraussichtlich wird man die maximale Anzahl der Integrationsschritte vergrößern müssen (MAXIMUM INTEGRATION PASSES).

Treten in einer Schaltung Impulse mit sehr geringer Impulsbreite auf, so muß die Schrittweite soweit verringert werden, daß SCEPTRE diesen Impuls erkennen kann.

Es empfiehlt sich — vor allem bei rechenintensiven Simulationen — vorher Testläufe mit jeweils einem der vier Integrationsverfahren durchzuführen und die Ergebnisse sowie die Rechenzeit kritisch miteinander zu vergleichen.

Es gibt Simulationsprobleme, bei denen das eine oder andere Integrationsverfahren versagt oder unverhältnismäßig viel Rechenzeit benötigt. Auf die Problematik der numerischen Integrationsverfahren wird in [2] und besonders in [9, Kapitel 3.5] eingegangen.

2.9. Definition abhängiger Quellen

Es ist wichtig, daß abhängige Quellen richtig in SCEPTRE definiert werden, damit sie verzögerungsfrei bei einer Transientanalyse zur Verfügung stehen⁵. Es gibt vier Arten von abhängigen Quellen:

³siehe Kapitel 1.10: TR-Analyse

⁴SCEPTRE listet stets alle Zustandsvariablen zusammen mit dem zugehörigen Index auf.

⁵siehe Kapitel 2.7

2. Allgemeine Hinweise

1. widerstandsabhängige Spannungsquellen:
 $E_{xx}, \text{node1} - \text{node2} = \text{constant} * V_{Ryy}$
2. widerstandsabhängige Stromquellen:
 $J_{xx}, \text{node1} - \text{node2} = \text{constant} * I_{Ryy}$
3. Dioden (Stromquellen 1. Ordnung):
 $J_{xx}, \text{node1} - \text{node2} = \text{DIODE EQUATION (x1,x2)}$
 $J_{xx}, \text{node1} - \text{node2} = \text{DIODE TABLE xy}$
4. diodenabhängige Stromquellen (2. Ordnung):
 $J_{xx}, \text{node1} - \text{node2} = \text{variable} * J_{yy}$
wobei J_{yy} eine Stromquelle 1. Ordnung sein muß.

Diese Quellen müssen exakt so definiert werden, andernfalls erkennt SCEPTRE diese Quellen nicht richtig. So kann z. B. die Schreibweise

$$E_{xx}, \text{node1} - \text{node2} = X1 (\text{constant} * V_{Ryy})$$

zu einem "computational delay" führen, da SCEPTRE die Spannungsquelle E_{xx} nicht als widerstandsabhängig erkennt.

Die richtige Definition von abhängigen Quellen führt im allgemeinen zu kürzeren Rechenzeiten bei der DC- und TR-Analyse.

Nicht richtig definierte Quellen werden bei der AC-Analyse wie Gleichstromquellen behandelt und auf Null gesetzt.

Vielfach läßt sich eine anders definierte abhängige Quelle in eine Quelle der o. a. Form umwandeln. Z. B. statt

$$\begin{aligned} E1, N1 - N2 &= X1 (3. * VC1) \\ C1, NA - NB &= \dots \end{aligned}$$

kann man einen großen Widerstand parallel zu C1 schalten und dann schreiben

$$\begin{aligned} E1, N1 - N2 &= 3. * VR1 \\ C1, NA - NB &= \dots \\ R1, NA - NB &= \text{grosser Wert} \end{aligned}$$

Ähnlich kann man eine stromabhängige Spannungsquelle

$$\begin{aligned} E1, N1 - N2 &= X1 (3. * IR1) \\ R1, NA - NB &= 2. \end{aligned}$$

in die richtige Form bringen unter Einbeziehung des Widerstandes R1.

$$\begin{aligned} E1, N1 - N2 &= 1.5 * VR1 \\ R1, NA - NB &= 2. \end{aligned}$$

2.10. Strom- und Spannungsmessungen

Den Strom durch eine Komponente erhält man, indem man dem Komponentennamen den Buchstaben I voranstellt. Sinngemäß erhält man die Spannung an einer Komponente durch Vorstellen des Buchstabens V ⁶. Will man Spannungen zwischen weit auseinanderliegenden Knotenpunkten messen, kann man mit Hilfe einer Nullstromquelle zwischen diesen Punkten die Spannung messen, ohne die Schaltung zu beeinflussen.

```
Jxx, NA - NZ = 0
VJxx, PLOT
```

Entsprechend lassen sich auch Ströme in Zweigen mit einer Nullspannungsquelle messen, ohne daß ein Shuntwiderstand eingefügt werden muß.

```
Exx, NM - NN = 0
IExx, PLOT
```

2.11. Verwendung eigener FORTRAN-Unterprogramme

Das Einfügen eigener FORTRAN Unterprogramme gibt dem Benutzer Gelegenheit, spezielle Berechnungen durchzuführen, die nicht direkt in SCEPTRE möglich sind. Es können Werte zwischen SCEPTRE und dem Unterprogramm in beiden Richtungen ausgetauscht werden.

FORTRAN-Unterprogramme ruft man mit der SCEPTRE-Anweisung

$$\text{Netzwerkvariable} = F \text{ name } (\text{Parameterliste})$$

auf. Im allgemeinen wird *Netzwerkvariablen* ein Wert zugewiesen.

Der Unterprogrammname *Fname* darf maximal 6 Zeichen lang sein. Das entsprechende FORTRAN Unterprogramm muß mit der folgenden Anweisung beginnen:

```
DOUBLE PRECISION FUNCTION F name (Parameterliste) .
```

Alle in der Parameterliste vorkommenden Variablen müssen doppelt genau definiert werden⁷.

Zusätzlich können Unterprogramme über den COMMON-Block /CNTRLS/⁸ interne SCEPTRE-Variablen abfragen.

Achtung: die SCEPTRE-Variablen in der Parameterliste dürfen in dem Unterprogramm nicht verändert werden.

Möchte man bei jedem Integrationsschritt FORTRAN-Unterprogramme aufrufen ohne Daten auszutauschen, um z. B. Daten in eine externe Datei zu schreiben oder eine eigene Graphik zu erzeugen, muß man auch die obige Schreibweise verwenden und eine Hilfsvariable benutzen, z. B.

```
DEFINED PARAMETERS
  PDUMM = Fname (Parameterliste)
```

⁶vgl. Tabelle 1.1

⁷REAL*8, COMPLEX*16

⁸siehe Anhang C

2. Allgemeine Hinweise

SCEPTRE prüft aber beim Generieren der FORTRAN-Anweisungen, wo die Variable PDUMM benötigt wird. Gibt es keine Verwendung für die Variable, wird die Anweisung zum Berechnen von PDUMM nicht erzeugt. *Damit das Unterprogramm bei jedem Integrationsschritt aufgerufen wird, muß die Hilfsvariable irgendwo in der Eingabe noch ein zweites Mal auftauchen*, am besten unter OUTPUTS. Das Beispiel muß also um die Anweisungen

```
OUTPUTS
  PDUMM, . . . .
```

ergänzt werden.

2.12. Die Vektornotation

SCEPTRE verwendet in den generierten FORTRAN-Unterprogrammen die gleichen Variablennamen wie sie in der Eingabe vom Benutzer gewählt wurden (z. B. PC, PL, C001 etc.). Werden mehr als 70 Elemente benutzt, schaltet SCEPTRE in eine Vektornotation um, d. h. alle Elemente werden in einem Vektor X und die definierten Parameter in einem Vektor P abgespeichert. Zusätzlich wird eine Korrespondenzliste ausgedruckt.

Eine Möglichkeit, Formelfunktionen zu verwenden, zeigt das folgende, allerdings nicht gute Beispiel:

```
ELEMENTS
  CX, . . . . = QC (VCX)
DEFINED PARAMETERS
  PL = 4711
FUNCTIONS
  QC(B) = (PL/(1.+B)*DLOG(2.+B))
```

Die Kapazität CX wird mittels der Funktion QC aus den Parametern VCX und PL berechnet. Da PL nicht als Argument beim Formelaufwurf auftritt, wird nach FORTRAN-Konvention der Wert von der lokalen Variablen PL eingesetzt. Solange SCEPTRE die Variablenbezeichnungen beibehält, d. h. bis zu 70 Elementen, ist der Parameter PL im FORTRAN-Unterprogramm bekannt und der Formelaufwurf funktioniert. Schaltet SCEPTRE in die Vektornotation um, wird der Parameter PL in das Vektorelement z. B. P(2) umbenannt, in der Formelfunktion steht aber PL, d. h. der FORTRAN-Compiler setzt für PL Hausnummern ein. Man umgeht diese Gefahr, wenn man alle benötigten Parameter in der Funktion als Argumente übergibt.

```
ELEMENTS
  CX, . . . . = QC (PL,VCX)
FUNCTIONS
  QC(A,B) = (A/(1.+B)*DLOG(2.+B))
```

2.13. Wahl der Zustandsvariablen

SCEPTRE wählt die Zustandsvariablen nach eigenen Kriterien aus. Die Reihenfolge der Netzwerkbeschreibungen unter ELEMENTS spielt dabei eine gewisse Rolle. Durch Umsortierung der Eingabeanweisungen wählt SCEPTRE unter Umständen andere Zustandsvariablen aus. Zusätzlich sollte man unter RUN CONTROLS die Anweisung

```
NO ELEMENT SORT
```

einfügen. Von der Wahl der Zustandsvariablen und den Zeitkonstanten im Netzwerk hängt die Integrationsschrittweite ab. Unter Umständen können günstigere Integrationsschrittweiten erzielt werden. Bei umfangreichen Simulationen empfiehlt es sich daher — um Rechenzeit zu sparen — diese Methoden vorher auszuprobieren.

2.14. Berechnung von Integralen

Da die Ableitungen definierter Parameter spezifiziert werden können, gibt es eine einfache Möglichkeit, Integrale zu berechnen. Soll die Funktion $f(t)$ mit dem Anfangswert $f(0) = \textit{number}$ integriert werden, lauten die Anweisungen unter DEFINED PARAMETERS

$$\boxed{DPname = f(t)}$$

$$\boxed{Pname = \textit{number}}$$

Am Ende der Simulation enthält Pname den Wert

$$Pname = \int_0^{\text{STOP TIME}} f(t)dt + \textit{number}.$$

Damit man die Integrale quantitativ einfacher auswerten kann, werden am Ende einer TR-Analyse der erste und letzte Wert jeder Ausgabevariablen ausgedruckt.

Hinweis: Als Anfangswert dürfen keine Variablen benutzt werden.

2.15. Simulationsstatistik

Am Ende einer AC- und TR-Analyse druckt SCEPTRE eine Statistik und einige Kontrollmeldungen aus.

INTEGRATION ROUTINE

Es wird das verwendete Integrationsverfahren ausgegeben.

CURRENT SIMULATION TIME

Es wird die Zeit ausgedruckt, bis zu der die Simulation tatsächlich durchgeführt wurde.

INTEGRATION STEP COUNTER

Dieser Zähler gibt die Zahl der erfolgreichen Integrationsschritte an und entspricht somit der Anzahl der zur Verfügung stehenden Ergebnisse.

INTEGRATION PASS COUNTER

Dieser Zähler gibt die Anzahl der Integrationsschritte plus der Integrationsversuche an, d. h. er enthält auch die Integrationsschritte, die mit verkleinerten Schrittweiten wiederholt wurden.

ELAPSED EXECUTION TIME [MIN]

Diese Zeitangabe bezieht sich auf die Verweilzeit der Simulation im Rechner, sie ist sehr stark von der Rechenauslastung abhängig.

CPU EXECUTION TIME [SEC]

2. Allgemeine Hinweise

Diese Zeitangabe entspricht der tatsächlichen Rechenzeit, jedoch nur für die Simulation. Die Rechenzeiten für das Verarbeiten der Eingabedaten, die FORTRAN-Übersetzung und das Linken sind nicht darin enthalten.

TERMINATION CONDITION

Diese Information ist die wichtigste und sollte deshalb nach jeder Simulation zuerst kontrolliert werden.

Folgende Meldungen sind möglich:

- STOP TIME EXCEEDED (NORMAL STOP)
Die Simulation ist normal verlaufen.
- MAXIMUM INTEGRATION PASSES EXCEEDED
Die Simulation ist vorzeitig abgebrochen worden, da mehr Integrationsschritte notwendig waren, als vorgegeben.
- SMALLER MINIMUM STEP SIZE REQUIRED
siehe Kapitel 2.8.
- SOLUTION TIME LIMIT EXCEEDED
Die vorgegebene Rechenzeit ist aufgebraucht, die Simulation ist vorzeitig abgebrochen worden.
- OPTIONAL TERMINATION CONDITION MET
Ein vom Benutzer vorgegebenes Abbruchkriterium ist erfüllt worden⁹.

Bei der AC-Analyse enthält die Statistik folgende Angaben, die sich selbst erklären:

INITIAL FREQUENCY
LAST CALCULATED FREQUENCY
NUMBER FREQUENCY STEPS
TYPE FREQUENCY RUN

2.16. Druckausgabe

Jede Variable, die man unter OUTPUTS angibt, kann auch tabellarisch ausgedruckt werden. Voreingestellt ist MAXIMUM PRINT POINTS = 0, d. h. der Ausdruck wird unterdrückt.

Benutzt man den Befehl

PRINT INTERVAL = number

dann wird diese Druckausgabe *zusätzlich* erzeugt. Auch hier kann man die Standard-Druckausgabe mit dem o. a. Kommando unterbinden.

Die Anweisungen

PLOT INTERVAL =
PRINT INTERVAL =

gehören zu der Überschrift RUN CONTROLS und nicht OUTPUTS.

⁹siehe Kapitel 1.10: Ablaufsteuerung

2.17. Anfangsbedingungen

2.17.1. Arbeitspunkt

Zu Beginn jeder TR-Analyse haben die Spannungen an allen Elementen und die Ströme durch alle Elemente den Wert Null. Dies trifft auch auf Dioden-Stromquellen zu. Enthält eine Schaltung Energiespeicher und/oder Dioden-Stromquellen, stellt sich der Arbeitspunkt der Schaltung erst nach einer hinreichend langen Einschaltzeit der Versorgungsspannung ein.

SCEPTRE bietet die Möglichkeit, diesen Arbeitspunkt automatisch berechnen zu lassen mit der Anweisung `RUN INITIAL CONDITIONS [ONLY]` (siehe Seite 14 und Seite 18).

Vorsicht ist bei zeitabhängigen Tabellen geboten. Es werden grundsätzlich die Werte zur Zeit $t = 0$ eingesetzt. Liegt zu dieser Zeit eine Sprungfunktion vor, wird der erste Wert angenommen. Betrachtet man dazu die Funktion in Abb. 1.1 (Seite 13) und die dazugehörige Tabelle `TABLE XYZ7` (Abschnitt 1.9.2), würde SCEPTRE zum Zeitpunkt $t = 0$ den Wert 0 annehmen. Läßt man das erste Wertepaar (0,0) in der Tabelle weg, ergibt sich für die Anfangsbedingung der Wert 1. Der Spannungsverlauf ändert sich dadurch nicht.

Alternativ lassen sich die Anfangsbedingungen auch manuell unter `INITIAL CONDITIONS` vorgeben (siehe Seite 12).

Man kann auch beide Verfahren gleichzeitig anwenden. Die Werte unter `INITIAL CONDITIONS` werden dann als Startwerte für die Berechnung des Arbeitspunktes genommen.

Die Anfangsbedingungen lassen sich nicht berechnen, wenn Ströme durch Widerstände als unabhängige Variablen in Tabellen oder Gleichungen auftreten (statt dessen kann man I_x durch V_x/R_x ausdrücken, siehe dazu auch Abschnitt 2.9) oder wenn Größen in Abhängigkeit von Kapazitäten oder Induktivitäten verwendet werden, da diese bei der Berechnung der Anfangsbedingungen noch nicht existieren.

Achtung: enthält das zu untersuchende Problem Differentialgleichungen (`P . . . , DP . . .`), kann man die Anfangsbedingungen nur mittels der impliziten Integration bestimmen (siehe nächster Abschnitt).

2.17.2. Implizite Integration

Standardmäßig werden die Anfangsbedingungen nach der Newton-Raphson Methode iterativ berechnet. Unter Umständen konvergiert das Verfahren jedoch nicht oder das Netzwerk enthält für dieses Verfahren eine ungültige Topologie (E-L Maschen, J-C Knoten). In diesem Fall könnte man eine TR-Analyse mit "eingefrorenen" Anregungsfunktionen durchführen und so lange warten, bis sich das Netzwerk auf die Anfangswerte eingependelt hat. Diese Methode ist nicht besonders effektiv, besonders dann, wenn die Zeitkonstanten im Netzwerk wertemäßig weit auseinander liegen.

Hier bietet SCEPTRE alternativ das implizite Integrationsverfahren an, das eine "Pseudo"-TR-Analyse mit folgenden Parametern durchführt:

```
MINIMUM STEP SIZE = 0
MAXIMUM STEP SIZE = 1E74 (oder 1E37)
STARTING STEP SIZE = 1E-8
```

Die "Integration" wird beendet, wenn für die Ableitungen des Zustandsvektors y gilt

$$(2.1) \quad |\dot{y}| \leq 10^{-8}|y| + 10^{-6}.$$

Um statt der Newton-Raphson Methode das implizite Verfahren zu aktivieren, müssen je nach Analyseart unter `RUN CONTROLS` folgende Anweisungen eingefügt werden:

2. Allgemeine Hinweise

- nur DC-Analyse (mit oder ohne Wiederholungsläufe)

```
RUN CONTROLS
  RUN INITIAL CONDITIONS ONLY
  RUN IC VIA IMPLICIT
```

- DC- und TR-Analyse

```
RUN CONTROLS
  RUN IC VIA IMPLICIT
  STOP TIME = ...
```

- DC- und TR-Analyse, Wiederholungsläufe nur mit TR-Analyse

```
RUN CONTROLS
  RUN IC VIA IMPLICIT
  STOP TIME = ...
RERUN DESCRIPTION
  ...
```

- DC- und TR-Analyse, Wiederholungsläufe ebenfalls mit DC- und TR-Analyse

```
RUN CONTROLS
  RUN IC VIA IMPLICIT
  STOP TIME = ...
RERUN DESCRIPTION
RUN CONTROLS
  RUN IC VIA IMPLICIT
  ...
```

2.17.3. Wiederholungsläufe

DC-Analyse

Die iterative Berechnung der Anfangswerte nach der Newton-Raphson Methode benutzt als Startwerte die unter INITIAL CONDITIONS angegebenen Strom- und Spannungswerte. Fehlende Werte werden zu Beginn der Iteration auf Null gesetzt. Die Vorgabe geeigneter Startwerte kann unter Umständen die Konvergenz beschleunigen und die Berechnung erheblich verkürzen.

Das gleiche Problem tritt bei Wiederholungsläufen auf. Es läßt sich Rechenzeit sparen, wenn man die Endwerte der vorherigen DC-Analyse als Startwerte für die weiteren DC-Analysen einsetzt. Bei mehr als einem Wiederholungslauf hat man die Wahl, welche Startwerte benutzt werden sollen: sollen die DC-Analysen aller Wiederholungsläufe mit den Endwerten der 1. DC-Analyse beginnen, lauten die Anweisungen *im 1. Lauf*:

```
RUN CONTROLS
  RUN INITIAL CONDITIONS ONLY
  (RUN IC VIA IMPLICIT)
  IC FOR RERUNS = MASTER RESULTS
```

```
RERUN DESCRIPTION (n)
...
```

es können auch die Werte aus dem vorherigen Lauf übernommen werden:

```
RUN CONTROLS
  RUN INITIAL CONDITIONS ONLY
  (RUN IC VIA IMPLICIT)
  IC FOR RERUNS = PRECEDING RESULTS
RERUN DESCRIPTION (n)
...
```

Das implizite Verfahren ist optional (Klammern weglassen!).

TR-Analyse

Statt die Anfangsbedingungen bei Wiederholungsläufen über die DC-Analyse zu berechnen, kann man ebenso die Endwerte einer TR-Analyse als Startwerte weiterer TR-Analysen verwenden. Auch hier können die Startwerte entweder aus den Ergebnissen der 1. TR-Analyse (IC FOR RERUN = MASTER RESULTS) oder aus der vorherigen TR-Analyse (IC FOR RERUNS = PRECEDING RESULTS) eingesetzt werden.

2.18. Periodische Funktionen

Die Erzeugung von beliebigen, periodischen Funktionsverläufen kann man in SCEPTRE auf verschiedene Arten realisieren: durch arithmetische Anweisungen, Tabellen und eigene FORTRAN Unterprogramme. Dabei dürfen diese Funktionen auch springen, d. h. zu einem unabhängigen Wert können zwei abhängige Werte vorgegeben werden (z. B. ideale Rechteckfunktion).

2.18.1. Ideale Funktionen

Mit einer einfachen EXPRESSION-Anweisung lassen sich ideale, periodische Funktionen generieren. Haben diese Funktionen keine direkte Einwirkung auf die Netzwerkdynamik, z. B. als Referenzfunktion, können Ungenauigkeiten vor allem an den Sprungstellen auftreten, da sich die Integrationsschrittweite nur nach der Dynamik des Netzwerkes richtet und u. U. der Integrationszeitpunkt nicht mit dem Sprungzeitpunkt zusammenfällt.

Rechteckfunktion

Durch Kombination der Vorzeichen- und Sinus-/Kosinusfunktionen lassen sich ideale Rechteckimpulse erzeugen. Beispiel: mit der Anweisung

```
EXXX, ..... = X1 (SIGN (311.,SIN (314.15*TIME) ) )
```

wird eine Rechteckspannung mit einer Frequenz von 50 und einer Amplitude von ± 311 Einheiten erzeugt. Durch Hinzufügen eines Offsets und einer Phasenverschiebung lassen sich die Rechteckimpulse in weiten Grenzen den Erfordernissen anpassen.

2. Allgemeine Hinweise

Rampenfunktion

Beispiel: zur Erzeugung einer "Kippspannung" kann folgende Anweisung benutzt werden:

```
ERAMP, ..... = X2 ( 3.8 + (8.2-3.8)*MOD ( (TIME/400E-6) ,1D0) )
```

Die Spannung setzt bei 3.8 ein und steigt innerhalb von 400E-6 Zeiteinheiten linear auf den Wert von 8.2. Danach wiederholt sich der gesamte Zyklus.

Getastete Sinusfunktion (Zeitmultiplex-Signal)

Abbildung 2.1 zeigt eine getastete Sinusfunktion, die mit folgender Anweisung erzeugt wurde:

```
..... = X12 ( SIN(6.281*TIME) * (0.5+SIGN(0.5, SIN(75.500*TIME))) )
```

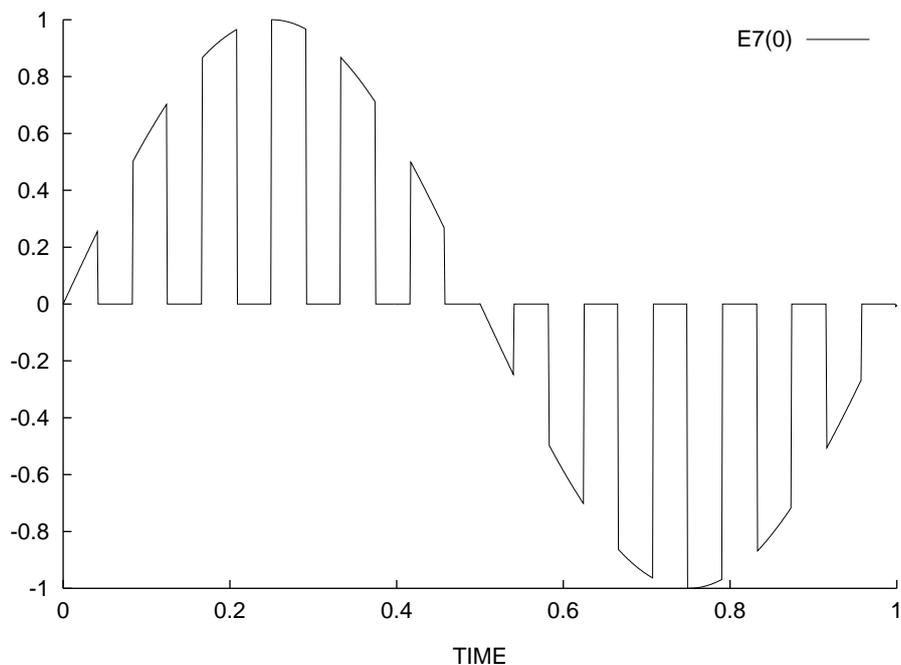


Abbildung 2.1.: Zeitmultiplexsignal

2.18.2. FPULSE

Häufig werden Rechteckimpulse zum Testen oder Ansteuern von Schaltungen benutzt. Sprungstellen in den Funktionsverläufen verschlechtern im allgemeinen drastisch die Konvergenz des Integrationsverfahrens. Aus diesem Grund verwendet man statt der idealen Funktion vorteilhafter eine Trapezfunktion mit endlichen Anstiegs- und Abfallzeiten.

Die Funktion FPULSE erzeugt einen periodischen Impuls entsprechend Abbildung 2.2. Aufgerufen wird diese

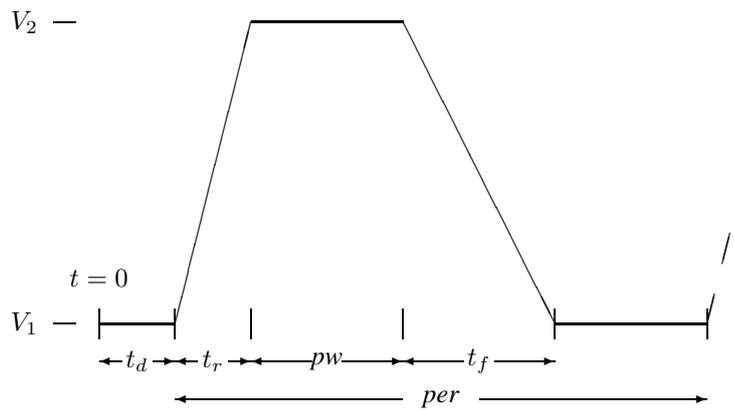


Abbildung 2.2.: Die periodische Funktion FPULSE

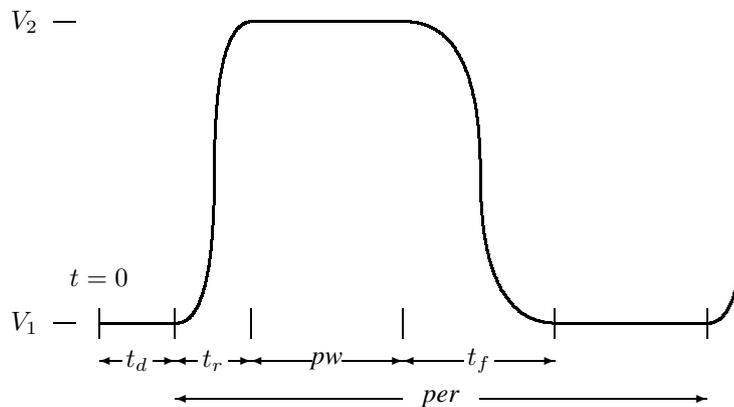


Abbildung 2.3.: Die periodische Funktion FSINSQ

Funktion wie eine eigene FORTRAN Funktion:

```
..... = FPULSE ( v1, v2, td, tr, tf, pw, per ) .
```

Die Bedeutung der einzelnen Parameter geht aus der Abbildung 2.2 hervor. Der Parameter *td* stellt eine einmalige Verzögerungszeit dar.

2.18.3. FSINSQ

Ersetzt man die Flanken eines idealen Rechteckimpulses durch \sin^2 -Verläufe, erhält man die Funktion FSINSQ entsprechend der Abbildung 2.3.

```
..... = FSINSQ ( v1, v2, td, tr, tf, pw, per )
```

Die Bedeutung der Parameter geht aus der Abbildung 2.3 hervor. Der Parameter *td* stellt eine einmalige Verzögerungszeit dar.

2.19. Reservierte Variablennamen

Die vom Benutzer gewählten Namen können unter unglücklichen Umständen zu Konflikten mit SCEPTRE-internen Programmvariablen führen. Diese Probleme werden erst durch den FORTRAN-Compiler entdeckt.

Folgende Variablennamen sollte man vermeiden:

PS zusammen mit DPS.

SCEPTRE unterscheidet bei den frei wählbaren Namen zwischen Groß- und Kleinschreibung. Je nach FORTRAN-Compiler werden u. U. alle kleingeschriebenen Variablen in Großbuchstaben umgewandelt. Damit entfällt das Unterscheidungsmerkmal und es kommt zu Laufzeitfehlern.

Beispiel:

```
PMI = .....  
Pmi = .....
```

Je nach FORTRAN-Compiler führen bestimmte Dummy-Argumente bei der EQUATION-Anweisung zu Übersetzungsfehlern (sie werden mit programminternen Variablen gleichen Namens verwechselt).

Beispiel:

```
Q9 (A,B,C,....) = .....
```

besser

```
Q9 (AA,BB,CC,....) = .....
```

2.20. Wechselstromquellen

Bei der Angabe von Wechselstromquellen (siehe Abschnitt 1.5.8) müssen rechts vom Gleichheitszeichen *zwei* Angaben, durch Komma getrennt, *in runden Klammern* angegeben werden (Abweichung vom Format für die TR-Analyse). Andernfalls wird die Quelle *ohne Fehlernachricht* ignoriert. Beispiel:

```
E1, Gnd - 2.2 = (325.27,0.)
```

3. SCEPTRE Beispiele

3.1. A01: Inverter-Schaltung mit RC-Last

Abb. 3.1 zeigt eine einfache Inverterschaltung, die mit einem RC Netzwerk belastet wird, sowie die für SCEPTRE aufbereitete Ersatzschaltung. Es soll der Einfluß von γ -Strahlung auf die Funktion der Schaltung untersucht werden. Der durch die Strahlung ausgelöste Ladungstransport im Halbleiter wird durch eine zeitabhängige Stromquelle JX simuliert. Der Stromverlauf in Abhängigkeit von der Zeit wird durch Tabelle 1 beschrieben. Die Anfangsbedingungen (Arbeitspunkt) werden explizit vorgegeben. Es handelt sich hier um eine reine Zeitanalyse. Für den Transistor wird die Ersatzschaltung nach Ebers-Moll benutzt. Die Beschreibung der beiden Dioden J1 und J2 erfolgt durch spannungsgesteuerte Stromquellen mit der in SCEPTRE eingebauten Funktion DIODE EQUATION (x1, x2). Dabei errechnet sich der Diodenstrom J direkt aus der Diodenspannung VJ nach der Beziehung $J = x_1(\exp(x_2 VJ) - 1)$. Um den Einfluß verschiedener Stromverläufe von JX beurteilen zu können, wird die Schaltung in zwei Wiederholungsläufen mit veränderter Tabelle 1 untersucht. Abb. 3.2 zeigt einen Teil der Ergebnisse.

CIRCUIT DESCRIPTION

A01: INVERTER CIRCUIT LOADED WITH RC NETWORK

ELEMENTS

```

E1, 7-1 = 10
E2, 1-6 = 10
CE, 3-1 = EQUATION 1 (5., 70., J1)
CC, 3-4 = EQUATION 1 (8., 370., J2)
C1, 5-1 = 500
R1, 1-2 = 2
R2, 2-7 = 17
R3, 6-5 = 1.5
R4, 5-1 = 18.5
RB, 2-3 = .3
    
```

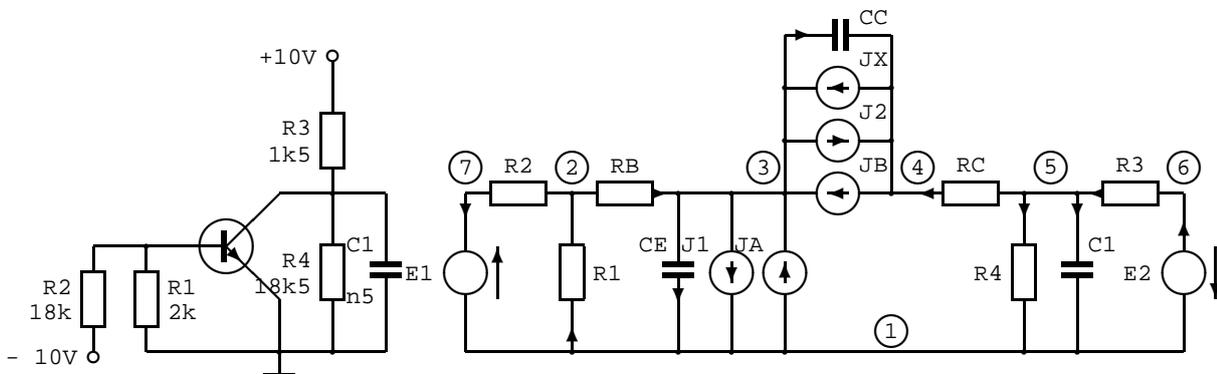


Abbildung 3.1.: Inverterschaltung und für SCEPTRE aufbereitetes Ersatzschaltbild

3. SCEPTRE Beispiele

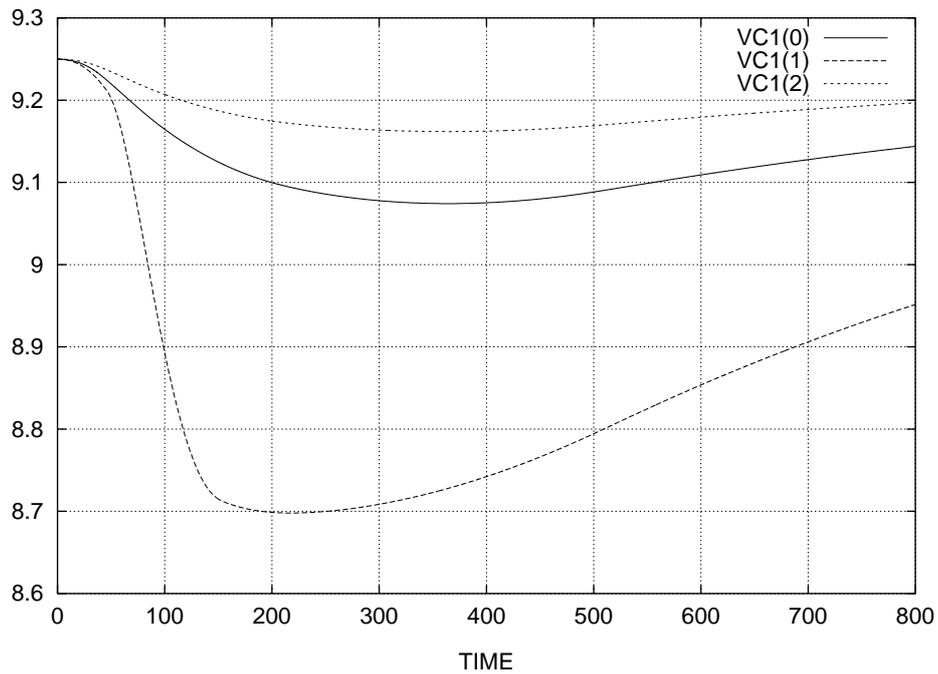


Abbildung 3.2.: Spannung VC1 in Abhängigkeit von JX entsprechend TABLE 1

```

RC, 5-4 = .015
J1, 3-1 = DIODE EQUATION (1.E-7, 35.)
J2, 3-4 = DIODE EQUATION (5.E-7, 37.)
JA, 1-3 = .1 * J2
JB, 4-3 = .98 * J1
JX, 4-3 = TABLE 1 (TIME)
OUTPUTS
  VCE, VCC, VC1, IR3, J1, PLOT
INITIAL CONDITIONS
  VC1 = 9.25
  VCE = -1
  VCC = -10.25
FUNCTIONS
  TABLE 1
    0, 0
    40, .8
    100, .5
    200, .25
    500, 0
    600, 0
  EQUATION 1 (A,B,C) = (A + B * C)
RUN CONTROLS
  MAXIMUM PRINT POINTS = 0
  INTEGRATION ROUTINE = TRAP
  STOP TIME = 800
RERUN DESCRIPTION (2)
FUNCTIONS
  TABLE 1 = 0, 0, 0
             40, 1.2, .4

```

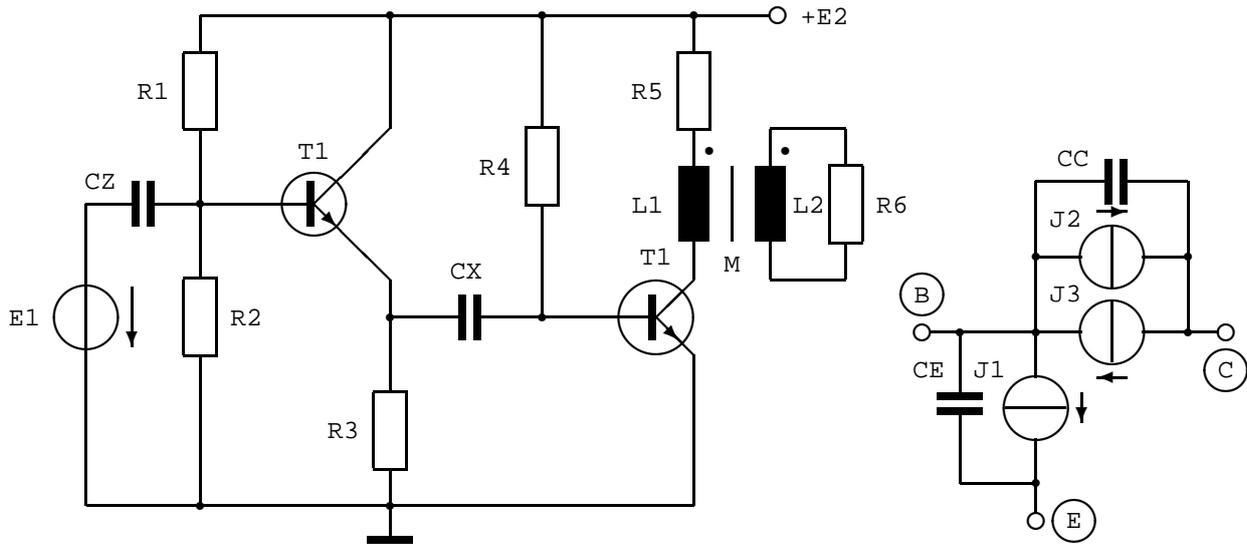


Abbildung 3.3.: Emitterfolger mit Übertrager, Transistor-Ersatzschaltbild

100,	.75,	.25
200,	.375,	.125
500,	0,	0
600,	0,	0

END

3.2. A02: Emitterfolger mit Übertrager

Eine kombinierte Kollektor- und Emitterschaltung, die einen mit einem Widerstand belasteten Übertrager treibt, zeigt Abb. 3.3. Die Schaltung wird durch eine rampenförmige Spannung E1 angesteuert. Das für die Simulation ausreichende Transistormodell ist ebenfalls wiedergegeben. Es wird für beide Transistoren verwendet und permanent gespeichert. Die Strom-Spannungscharakteristiken der beiden Dioden J1 und J2 werden durch die Diodentabellen 1 und 2 wiedergegeben. Der Transistor T2 arbeitet mit einer geringeren Stromverstärkung als T1 (0.975 statt 0.98). Der Übertrager besteht aus den Spulen L1 und L2, die über die Gegeninduktivität M miteinander gekoppelt sind. Die Simulation soll an einer betriebsbereiten Schaltung durchgeführt werden, d. h. es interessiert hier nicht der Einschwingvorgang, den das Einschalten der Spannungsversorgung verursacht. Deshalb muß vorher der Arbeitspunkt der Schaltung bestimmt werden (RUN INITIAL CONDITIONS), der dann als Ausgangszustand für die nachfolgende Transient-Analyse dient (STOP TIME). Die Spannungsverläufe VL1 und VL2 sind in Abbildung 3.4 wiedergegeben.

```

MODEL DESCRIPTION
MODEL 2N9999AA (PERM) (B-E-C)
ELEMENTS
CE,B-E=EQUATION 1(5.,40.,TABLE 1(VCE))
CC,B-C=EQUATION 1(10.,400.,TABLE 2(VCC))
J1,B-E=DIODE TABLE 1
J2,B-C=DIODE TABLE 2
J3,C-B=P1*J1
DEFINED PARAMETERS
P1=.98
    
```

3. SCEPTRE Beispiele

```
FUNCTIONS
  EQUATION 1(A,B,C)=(A+B*C)
  DIODE TABLE 1
    0,0,.3,0,.65,.05,.7,.6,.72,1.4,.73,2,.74,3.4,.77,10,.8,22
  DIODE TABLE 2
    0,0,.58,0,.62,.4,.64,1,.66,2,.67,3,.69,7,.7,12
CIRCUIT DESCRIPTION
EXAMPLE 2: TRANSFORMER COUPLED AMPLIFIER
ELEMENTS
  E1,1-2=TABLE 1(TIME)
  DERIVATIVE E1=TABLE DE1
  E2,1-4=20
  CZ,2-3=1E3
  CX,5-6=1E3
  R1,4-3=30
  R2,3-1=20
  R3,5-1=2
  R4,4-6=240
  R5,4-7=3.3
  R6,9-1=1.8
  T1,3-5-4=MODEL 2N9999AA (TEMP)
  T2,6-1-8=MODEL 2N9999AA (TEMP,CHANGE P1=.975)
  L1,7-8=100
  L2,9-1=900
  M,L1-L2=299.7
OUTPUTS
  VR6,VL1,VL2,PLOT
FUNCTIONS
  TABLE 1
    0,0,50,.5,100,.5
  TABLE DE1
    0,.01,50,.01,50,0,100,0
RUN CONTROLS
  STOP TIME=500
  RUN INITIAL CONDITIONS
  INTEGRATION ROUTINE=IMPLICIT
  MAXIMUM PRINT POINTS=21
END
```

3.3. A03: Darlington-Schaltung (DC-Analyse)

Das Prinzip einer Darlington Schaltung geht aus Abb. 3.5 hervor. Gesucht sind die Spannungen, die sich an den Widerständen R4 und R5 einstellen, sowie die Leistungsaufnahme der Schaltung, bei unterschiedlicher Stromverstärkung des Transistors T1. Dazu genügt eine Gleichstromanalyse (RUN INITIAL CONDITIONS ONLY). Die Beschreibung des Modells für den Transistor 2N706A existiert schon in der Modellbibliothek und erscheint deshalb nicht in den SCEPTRE-Eingabedaten. Sie ist jedoch der Übersicht wegen aufgelistet.

```
MODEL 2N706A (PERM) (B-E-C)
ELEMENTS
  CE, 1-E = Q1(5.,70.,J1)
  CC, 1-2 = Q1(8.,370.,J2)
  RB, B-1 = .3
  RC, C-2 = .015
```

3.3. A03: Darlington-Schaltung (DC-Analyse)

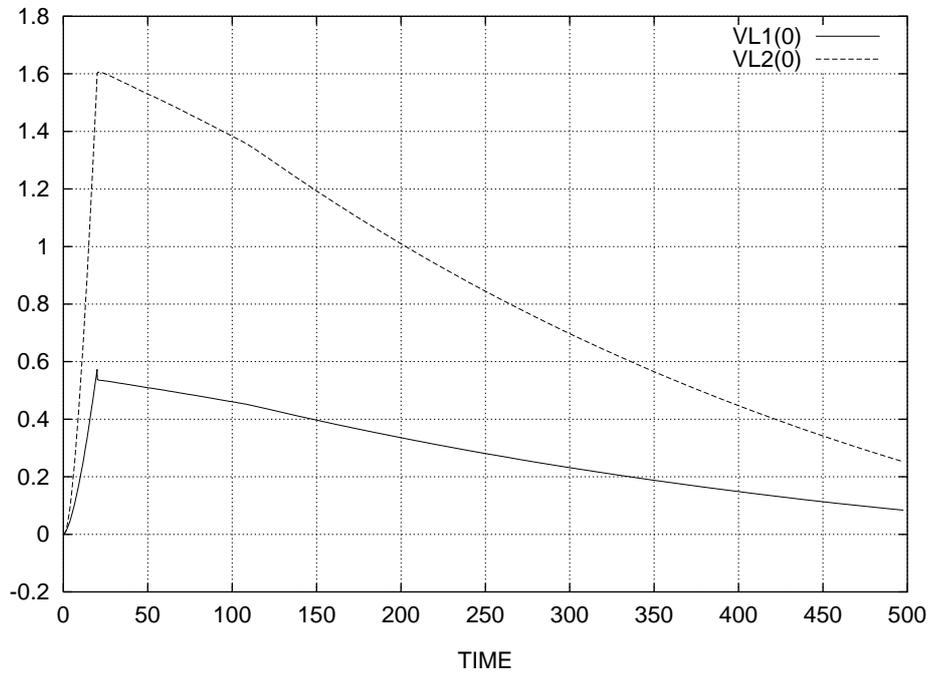


Abbildung 3.4.: Spannungen VL1 und VL2 am Übertrager

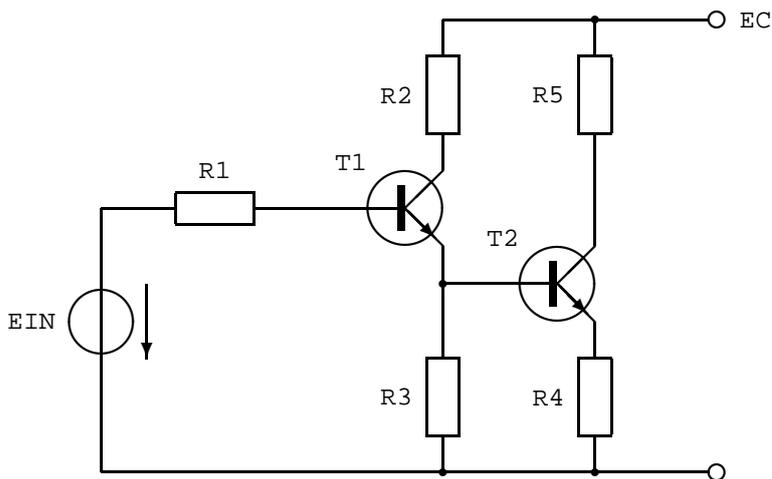


Abbildung 3.5.: Darlington Schaltung

3. SCEPTRE Beispiele

```
J1, 1-E = DIODE EQUATION(1.E-7,35.)
J2, 1-2 = DIODE EQUATION(5.E-7,37.)
JA, E-1 = .1*J2
JB, 2-1 = P1*J1
JX, 2-1 = 0
DEFINED PARAMETERS
P1 = 0.98
OUTPUTS
VCE,VCC,J1,PLOT
FUNCTIONS
Q1(A,B,C) = (A+B*C)
```

Die Stromverstärkung wird durch den Parameter P1 im Modell 2N706A dargestellt. Der Bezug zu der Stromverstärkung des Transistors T1 erfolgt durch Anhängen des Bezugsnamens T1 an den internen Modellnamen, also P1T1.

```
CIRCUIT DESCRIPTION
Example 3: Darlington pair
ELEMENTS
EC , 1-6 = 10
EIN , 1-2 = 1
R1 , 2-3 = 20
R2 , 6-5 = 5
R3 , 4-1 = 200
R4 , 8-1 = .4
R5 , 6-7 = 1
T1 , 3-4-5 = MODEL 2N706A (PERM)
T2 , 4-8-7 = MODEL 2N706A (PERM)
DEFINED PARAMETERS
PEC = X1 (EC*IEC)
PEIN= X2 (EIN*IEIN)
OUTPUTS
PEC,PEIN,VR3,VR4
RUN CONTROLS
RUN INITIAL CONDITIONS ONLY
RERUN DESCRIPTION (3)
DEFINED PARAMETERS
P1T1= .96, .93, .9
END
```

Die Ergebnisse der Gleichstromanalysen sind in Abb. 3.6 zusammengefaßt. Die Ausgabe der Ergebnisse setzt sich aus den unter OUTPUTS angegebenen Größen zusammen, und zwar sowohl von der Hauptschaltung als auch von der Modellbeschreibung.

3.4. A04: Kleinsignal-Ersatzschaltung

Dieses Beispiel soll den Einsatz von SCEPTRE bei der Analyse einer Kleinsignal-Schaltung nach Abbildung 3.7 zeigen. Es handelt sich dabei um einen RC gekoppelten, linearen, zweistufigen Verstärker, dessen Verhalten bei einer sinusförmigen Eingangsspannung E1 von 1 mV und 100 kHz untersucht werden soll. Für die Transistoren wird als temporäres Modell das Ersatzschaltbild mit h-Parametern benutzt. Da nur das Wechselstromverhalten

3.4. A04: Kleinsignal-Ersatzschaltung

I/C TRANSIENT VALUES AT TIME EQUALS ZERO:
 VCET1 = 3.3818865E-01
 VCCT1 = -8.9376695E+00
 VCET2 = 4.4221916E-01
 VCCT2 = -8.8226781E+00

RESULTS OF INITIAL CONDITION COMPUTATIONS
 PEC 5.3007344E+00
 PEIN 2.7598821E-04
 VR3 6.5620879E-01
 VR4 2.1082735E-01

I/C TRANSIENT VALUES AT TIME EQUALS ZERO:
 VCET1 = 3.3696571E-01
 VCCT1 = -8.9570427E+00
 VCET2 = 4.4077602E-01
 VCCT2 = -8.8603292E+00

RESULTS OF INITIAL CONDITION COMPUTATIONS
 RERUN 2
 PEC 5.0340188E+00
 PEIN 9.2654405E-04
 VR3 6.4422545E-01
 VR4 2.0044292E-01

I/C TRANSIENT VALUES AT TIME EQUALS ZERO:
 VCET1 = 3.3769191E-01
 VCCT1 = -8.9456210E+00
 VCET2 = 4.4163421E-01
 VCCT2 = -8.8381601E+00

RESULTS OF INITIAL CONDITION COMPUTATIONS
 RERUN 1
 PEC 5.1910074E+00
 PEIN 5.4289718E-04
 VR3 6.5128728E-01
 VR4 2.0655488E-01

I/C TRANSIENT VALUES AT TIME EQUALS ZERO:
 VCET1 = 3.3626101E-01
 VCCT1 = -8.9679012E+00
 VCET2 = 4.3993970E-01
 VCCT2 = -8.8813264E+00

RESULTS OF INITIAL CONDITION COMPUTATIONS
 RERUN 3
 PEC 4.8854795E+00
 PEIN 1.2915639E-03
 VR3 6.3752024E-01
 VR4 1.9466077E-01

SIMULATION PROGRAM HAS TERMINATED

Abbildung 3.6.: Ergebnisse der Gleichstromanalysen

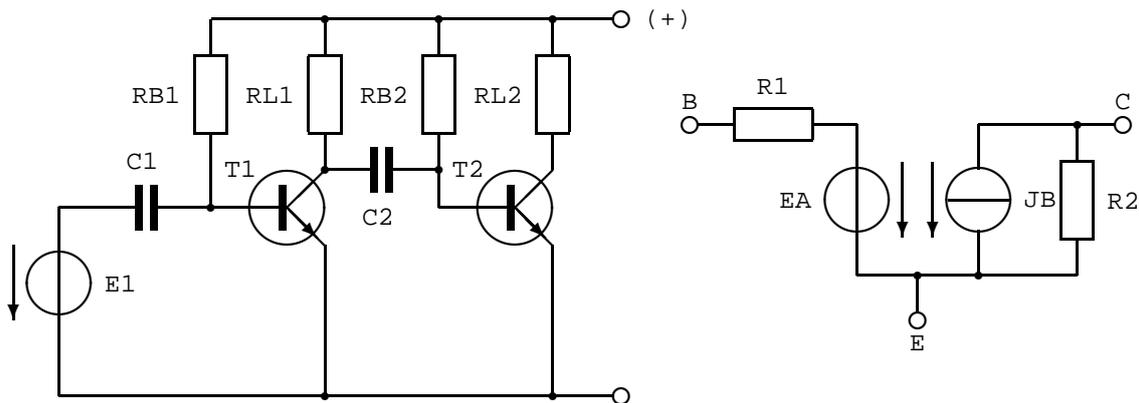


Abbildung 3.7.: Kleinsignal-Schaltung, Ersatzschaltbild mit h-Parametern

3. SCEPTRE Beispiele

interessiert, schließt man die Gleichspannungsquelle kurz. Die Simulation wird entweder nach 30000 ns beendet oder, wenn die Ausgangsspannung am Lastwiderstand RL2 größer als 20 V wird.

```
MODEL DESCRIPTION
MODEL SS1 (TEMP) (B-E-C)
ELEMENTS
  EA, E-X = .0005 * VR2
  R2, C-E = 2000
  R1, B-X = .3
  JB, C-E = 50. * IR1
CIRCUIT DESCRIPTION
A04 : small signal equivalent circuit
ELEMENTS
  E1, 1-2 = X1(.001*DSIN(.000628*TIME))
  C1, 2-3 = 5E6
  C2, 4-5 = 5E6
  RB1,3-1 = 100
RL1,1-4 = 1
RB2,5-1 = 100
RL2,1-6 = 1
T1, 3-1-4 = MODEL SS1
T2, 5-1-6 = MODEL SS1
OUTPUTS
  VRL1, VRL2, VC1, VC2, PLOT
RUN CONTROLS
  STOP TIME = 30000
  INTEGRATION ROUTINE = TRAP
  MAXIMUM PRINT POINTS = 21
  TERMINATE IF (VRL2 .GE. 20.)
END
```

3.5. A05: Differentialgleichungssystem

Dieses Beispiel zeigt die Anwendung der definierten Parameter zur Lösung eines Systems von Differentialgleichungen erster Ordnung. Dieses Gleichungssystem ist vollkommen unabhängig von einem elektrischen Netzwerk.

$$\begin{aligned}\dot{x} &= -6x + 5y && +10 \\ \dot{y} &= 5x - 7y && +2z \\ \dot{z} &= 0.2y - 0.2z && -0.5\end{aligned}$$

Anfangsbedingungen: $x(0) = 6, y(0) = 5, z(0) = 4$. Die SCEPTRE-Eingabe sieht folgendermaßen aus:

```
CIRCUIT DESCRIPTION
Example A05: Solution of simultaneous differential equations
DEFINED PARAMETERS
  DPX = EQUATION 1 (PX, PY)
  DPY = EQUATION 2 (PX, PY, PZ)
  DPZ = EQUATION 3 (PY, PZ)
  PX = 6
  PY = 5
  PZ = 4
OUTPUTS
  PX(X), PY(Y), PZ(Z), XSTPSZ, PLOT
FUNCTIONS
  EQUATION 1 (A,B) = (-6.*A+5.*B+10.)
  EQUATION 2 (A,B,C) = (5.*A-7.*B+2.*C)
  EQUATION 3 (A,B) = (.2*A-.2*B-.5)
RUN CONTROLS
  INTEGRATION ROUTINE = TRAP
  STOP TIME = 100
  MAXIMUM PRINT POINTS = 21
END
```

Die Anfangsbedingungen werden durch die Parameter PX, PY und PZ gesetzt. Nach jedem Integrationschritt werden ihnen neue Werte zugewiesen.

3.6. Toleranzanalysen einer Darlington-Schaltung

Die folgenden Beispiele befassen sich mit Toleranzanalysen der Darlington-Schaltung aus dem Beispiel A03. Obwohl alle Toleranzanalysen in einem Simulationslauf durchgeführt werden können, werden sie hier einzeln vorgestellt.

3.6.1. A07: Monte-Carlo-Analyse

Als unabhängige Variable für die Monte-Carlo-Analyse wird der Stromverstärkungsfaktor P1 im Modell 2N706A gewählt, dessen Nominalwert bei 0.96 liegt, aber Schwankungen von 0.9 bis 0.98 aufweisen soll. Unter der Überschrift MONTE CARLO werden alle zu beobachtenden Variablen angegeben (vor dem Schrägstrich), gefolgt von den Parametern, die entsprechend ihrer Toleranzgrenzen variiert werden sollen (P1T1, P1T2). Es wird eine Gauß-Verteilung angenommen (Voreinstellung).

```

MODEL DESCRIPTION
MODEL 2N706A (TEMP) (B-E-C)
ELEMENTS
  CE, 1-E = Q1(5.,70.,J1)
  CC, 1-2 = Q1(8.,370.,J2)
  RB, B-1 = .3
  RC, C-2 = .015
  J1, 1-E = DIODE EQUATION(1.E-7,35.)
  J2, 1-2 = DIODE EQUATION(5.E-7,37.)
  JA, E-1 = .1*J2
  JB, 2-1 = P1*J1
  JX, 2-1 = 0
DEFINED PARAMETERS
  P1 = 0.96 (0.98, 0.9)
FUNCTIONS
  Q1(A,B,C) = (A+B*C)
OUTPUTS
  VCE, VCC, J1
CIRCUIT DESCRIPTION
A07: Darlington pair, MONTE CARLO
ELEMENTS
  EC, 1-6 = 10
  EIN, 1-2 = 1
  R1, 2-3 = 20
  R2, 6-5 = 5
  R3, 4-1 = 200
  R4, 8-1 = .4
  R5, 6-7 = 1
  T1, 3-4-5 = MODEL 2N706A
  T2, 4-8-7 = MODEL 2N706A
MONTE CARLO
  (VCET1, VCCT1, J1T1, VCET2, VCCT2,
  J1T2, PEC, PEIN, VR3, VR4/P1T1, P1T2)
DEFINED PARAMETERS
  PEC = X1 (EC*IEC)
  PEIN= X2 (EIN*IEIN)
OUTPUTS
  PEC, PEIN, VR3, VR4
RUN CONTROLS
  RUN MONTE CARLO = 10
  RUN INITIAL CONDITIONS ONLY
END

```

Man beachte, wie man durch Anhängen des Modellbezugsnamens jedes Element und jede Variable in der Gesamtschaltung ansprechen kann. Das Ergebnis der statistischen Analyse zeigt Abbildung 3.8.

3.6.2. A08: Empfindlichkeitsanalyse

Die Empfindlichkeitsanalyse dient dazu, den Einfluß einzelner (unabhängiger) Elemente und Parameter auf andere (abhängige) Netzwerkvariablen — ausgedrückt durch partielle Ableitungen — abschätzen zu können. Für die Darlington-Schaltung werden zwei Sätze von partiellen Ableitungen gesucht, die unter der Überschrift SENSITIVITY angegeben werden. Toleranzangaben sind nicht erforderlich. Während SCEPTRE für die abhängigen Netzwerkelemente die partiellen Ableitungen selber berechnen kann, muß der Benutzer für eine definierte, abhängige Funktion Pname das vollständige Differential GPname vorgeben.

3. SCEPTRE Beispiele

I/C TRANSIENT VALUES AT TIME EQUALS ZERO:

OVCET1 = 3.5123970E-01
 OVCCT1 = -8.9126370E+00
 OVCET2 = 4.3874672E-01
 OVCCT2 = -8.9197511E+00

INPUTS TO MONTE CARLO

NAME	NOMINAL	LBOUND	UBOUND	MEAN	SIGMA
P1T1	9.6000000E-01	9.0000000E-01	9.8000000E-01	9.4000000E-01	1.3333333E-02
P1T2	9.6000000E-01	9.0000000E-01	9.8000000E-01	9.4000000E-01	1.3333333E-02

INITIAL RANDOM NUMBER = 127263527
 DISTRIBUTION IS GAUSSIAN
 NORMAL MONTE CARLO TERMINATION AFTER 10 ITERATIONS
 FINAL RANDOM NUMBER = 1007121511

OBSERVED STATISTICS

INDEPENDENT VARIABLE	NOMINAL VALUE	MINIMUM VALUE	MAXIMUM VALUE	SAMPLE MEAN	SAMPLE SIGMA
P1T1	9.6000000E-01	9.3082453E-01	9.7906312E-01	9.5388296E-01	1.6480857E-02
P1T2	9.6000000E-01	9.1926269E-01	9.5900296E-01	9.4255822E-01	1.1320930E-02

DEPENDENT VARIABLE	NOMINAL VALUE	MINIMUM VALUE	MAXIMUM VALUE	SAMPLE MEAN	SAMPLE SIGMA
VCET1	3.5123970E-01	3.5109792E-01	3.6510355E-01	3.5782344E-01	4.0157705E-03
VCCT1	-8.9126370E+00	-8.9210128E+00	-8.8596399E+00	-8.8927857E+00	2.2013686E-02
J1T1	2.1824751E-02	2.1716713E-02	3.5455596E-02	2.7726820E-02	3.9307541E-03
VCET2	4.3874672E-01	4.3440367E-01	4.3795162E-01	4.3649493E-01	1.4850834E-03
VCCT2	-8.9197511E+00	-9.0300613E+00	-8.9392260E+00	-8.9773819E+00	3.5350405E-02
J1T2	4.6675032E-01	4.0093037E-01	4.5394055E-01	4.3189852E-01	2.2252503E-02
PEC	4.6903307E+00	4.0210736E+00	4.5616323E+00	4.3372422E+00	2.2675435E-01
PEIN	8.7254006E-04	6.3257750E-04	1.8368401E-03	1.2575918E-03	4.3076374E-04
VR3	6.3104773E-01	6.0275509E-01	6.2660595E-01	6.1664745E-01	9.7793613E-03
VR4	1.8670015E-01	1.6037217E-01	1.8157624E-01	1.7275943E-01	8.9010012E-03

RESULTS OF INITIAL CONDITION COMPUTATIONS

VCET1	3.6510355E-01
VCCT1	-8.8596399E+00
J1T1	3.5455596E-02
VCET2	4.3445810E-01
VCCT2	-9.0300613E+00
J1T2	4.0169492E-01
PEC	4.0323994E+00
PEIN	1.4793579E-03
VR3	6.0486548E-01
VR4	1.6067799E-01

Abbildung 3.8.: Ergebnisse der Monte-Carlo-Analyse

3.6. Toleranzanalysen einer Darlington-Schaltung

```

MODEL DESCRIPTION
MODEL 2N706A (TEMP) (B-E-C)
ELEMENTS
  CE, 1-E = Q1(5.,70.,J1)
  CC, 1-2 = Q1(8.,370.,J2)
  RB, B-1 = .3
  RC, C-2 = .015
  J1, 1-E = DIODE EQUATION(1.E-7,35.)
  J2, 1-2 = DIODE EQUATION(5.E-7,37.)
  JA, E-1 = .1*J2
  JB, 2-1 = P1*J1
  JX, 2-1 = 0
DEFINED PARAMETERS
  P1 = 0.96
FUNCTIONS
  Q1(A,B,C) = (A+B*C)
OUTPUTS
  VCE, VCC, J1
CIRCUIT DESCRIPTION
Example A08: Darlington pair, SENSITIVITY
ELEMENTS
  EC, 1-6 = 10
  EIN, 1-2 = 1
  R1, 2-3 = 20
  R2, 6-5 = 5
  R3, 4-1 = 200
  R4, 8-1 = .4
  R5, 6-7 = 1
  T1, 3-4-5 = MODEL 2N706A
  T2, 4-8-7 = MODEL 2N706A
SENSITIVITY
  (VCET1,VCCT1,J1T1,PR3,PRR3/P1T1,P1T2,R3)
  (VCET2,VCCT2,PEIN,VR4/P1T1,P1T2,R3,R4)
DEFINED PARAMETERS
  PEIN= X2 (EIN*IEIN)
  GPEIN=P4*DIEIN
  P4 = X4 (EIN)
  PR3 = XR3 (R3*IR3*IR3)
  GPR3 = PDR3*DR3+PDIR3*DIR3
  PDR3 = XDR3 (IR3*IR3)
  PDIR3= XDIR3 (2.*IR3*R3)
  PRR3 = XRR3 (IR3*VR3)
  GPRR3= PDIRR3*DIR3+PDVRR3*DVR3
  PDIRR3 = XDIRR3 (VR3)
  PDVRR3 = XDVRR3 (IR3)
OUTPUTS
  VR3,VR4
RUN CONTROLS
  RUN SENSITIVITY
  RUN INITIAL CONDITIONS ONLY
END

```

Bei der vorliegenden Schaltung soll der Einfluß auf die Eingangsleistung PEIN und der Verlustleistung PR3 untersucht werden. Dabei wird die Leistung im Widerstand R3 sowohl über den Spannungsabfall VR3 als auch über den Strom IR3 berechnet. Aus

$$P_{EIN} = I_{EIN} E_{IN}$$

folgt für das vollständige Differential

$$dP_{EIN} = \frac{\partial P_{EIN}}{\partial I_{EIN}} dI_{EIN} + \frac{\partial P_{EIN}}{\partial E_{IN}} dE_{IN}.$$

Da $E_{IN} = const$ und nicht zu den unabhängigen Variablen gehört, ergibt sich in der SCEPTRE-Schreibweise:

```

PEIN = X1 (EIN * IEIN)
P2    = X2 (EIN)
GPEIN= P2 * DIEIN

```

Für die Leistung PR3 erhält man

$$P_{R3} = I_{R3}^2 R_3$$

$$dP_{R3} = \frac{\partial P_{R3}}{\partial I_{R3}} dI_{R3} + \frac{\partial P_{R3}}{\partial R_3} dR_3$$

und in der SCEPTRE-Schreibweise

3. SCEPTRE Beispiele

```

PR3 = X3 (R3 * IR3 * IR3)
P4 = X4 (2. * IR3 * R3)
P5 = X5 (IR3 * IR3)
GPR3 = P4 * DIR3 + P5 * DR3

```

Bezieht man die Leistung auf den Spannungsabfall VR3, so ergibt sich

$$P_{RR3} = I_{R3} V_{R3}$$

$$dP_{RR3} = \frac{\partial P_{RR3}}{\partial I_{R3}} DI_{R3} + \frac{\partial P_{RR3}}{\partial V_{R3}} DV_{R3}$$

und in der SCEPTRE-Schreibweise

```

PRR3 = X6 (IR3 * VR3)
P7 = X7 (VR3)
P8 = X8 (IR3)
GPRR3 = P7 * DIR3 + P8 * DVR3

```

Abbildung 3.9 zeigt die Ergebnisse der Empfindlichkeitsanalyse. Wie man erkennt, ergeben sich für die Leistungen PR3 und PRR3 die gleichen partiellen Ableitungen bzw. Empfindlichkeiten.

3.6.3. A09: 'Worst Case'-Analyse

Aufgabe der Worst-Case-Simulation ist es, Extremwerte einer Zielfunktion (=abhängige Netzwerkvariable) zu finden für Werte einer oder mehrerer unabhängiger Netzwerkvariablen, die Toleranzen haben müssen.

```

MODEL DESCRIPTION
MODEL 2N706A (TEMP) (B-E-C)
ELEMENTS
  CE, 1-E = Q1(5.,70.,J1)
  CC, 1-2 = Q1(8.,370.,J2)
  RB, B-1 = .3
  RC, C-2 = .015
  J1, 1-E = DIODE EQUATION(1.E-7,35.)
  J2, 1-2 = DIODE EQUATION(5.E-7,37.)
  JA, E-1 = .1*J2
  JB, 2-1 = P1*J1
DEFINED PARAMETERS
  P1 = 0.96 (0.98, 0.9)
FUNCTIONS
  Q1(A,B,C) = (A+B*C)
OUTPUTS
  VCE, VCC, J1
CIRCUIT DESCRIPTION
Example A09: Darlington pair, WORST CASE
ELEMENTS
  EC, 1-6 = 10
EIN, 1-2 = 1
R1, 2-3 = 20
R2, 6-5 = 5
R3, 4-1 = 200 (195,205)
R4, 8-1 = .4 (0.35,0.45)
R5, 6-7 = 1
T1, 3-4-5 = MODEL 2N706A
T2, 4-8-7 = MODEL 2N706A
WORST CASE
  (VCET1, VCCT1, J1T1, PEIN/P1T1, P1T2, R3)
  (VCET2, VR4/P1T2, R3, R4)
DEFINED PARAMETERS
  PEIN= X2 (EIN*IEIN)
  GPEIN=P4*DIEIN
  P4 = X4 (EIN)
OUTPUTS
  VR3,VR4
RUN CONTROLS
  RUN WORST CASE
  RUN INITIAL CONDITIONS ONLY
END

```

Für die Zielfunktion von PEIN muß das vollständige Differential vorgegeben werden. Ein Ausschnitt aus den Ergebnissen ist in Abbildung 3.10 wiedergegeben. Für die Nominalwerte von P1T1, P1T2 und R3 beträgt die Spannung VCCT1 -8.9126, sie nimmt ihren größten Wert von -8.8243 an, wenn P1T1 den Wert 0.98, P1T2 den Wert 0.9153 und R3 den Wert 200 annehmen. Der kleinste Wert für VCCT1 von -8.949 ergibt sich für P1T1=0.951, P1T2=0.98 und R3=200.

3.6. Toleranzanalysen einer Darlington-Schaltung

SENSITIVITY CALCULATION(S)

SET NUMBER	DEPENDENT VARIABLE	INDEPENDENT VARIABLE	PARTIAL DERIVATIVE	NORMALIZED SENSITIVITY
1.	VCET1	P1T1	2.23659125E-01	6.11299797E-01
		P1T2	-4.07175175E-02	-1.11288150E-01
		R3	-3.75988248E-06	-2.14092110E-03
	VCCT1	P1T1	1.23628942E+00	-1.33163490E-01
		P1T2	-1.24486175E-01	1.34086834E-02
		R3	-1.14951356E-05	2.57951394E-04
	J1T1	P1T1	1.70846451E-01	7.51498104E+00
		P1T2	-3.11028820E-02	-1.36811486E+00
		R3	-2.87206067E-06	-2.63192979E-02
	PR3	P1T1	2.07019909E-01	9.98134165E+01
		P1T2	5.27852161E-01	2.54500777E+02
		R3	5.68080410E-06	5.70617891E-01
	PRR3	P1T1	1.03764443E-01	5.00294086E+01
		P1T2	2.64134241E-01	1.27350752E+02
		R3	-2.11814175E-06	-2.12760299E-01
2.	VCET2	P1T1	1.04334237E-02	2.28288587E-02
		P1T2	2.66473139E-02	5.83056702E-02
		R3	7.88042611E-07	3.59224392E-04
		R4	-1.11083071E-02	-1.01273074E-02
	VCCT2	P1T1	2.44690518E-01	-2.63351403E-02
		P1T2	1.09869936E+00	-1.18248970E-01
		R3	1.84816183E-05	-4.14397623E-04
		R4	2.06232105E-01	-9.24833451E-03
	PEIN	P1T1	-1.49908934E-02	-1.64935209E+01
		P1T2	-1.24411528E-03	-1.36882044E+00
		R3	-1.14882427E-07	-2.63328717E-02
		R4	-7.18255438E-03	-3.29271044E+00
	VR4	P1T1	6.81772684E-02	3.50563074E-01
		P1T2	1.74127029E-01	8.95349841E-01
		R3	5.14946906E-06	5.51629887E-03
R4		3.94163071E-01	8.44483683E-01	

RESULTS OF INITIAL CONDITION COMPUTATIONS

VCET1	3.5123970E-01
VCCT1	-8.9126370E+00
J1T1	2.1824751E-02
VCET2	4.3874672E-01
VCCT2	-8.9197511E+00
J1T2	4.6675032E-01
VR3	6.3104773E-01
VR4	1.8670015E-01

Abbildung 3.9.: Ergebnisse der Empfindlichkeitsanalyse

3. SCEPTRE Beispiele

```

WORST CASE COMPUTATION - NOMINAL VALUE
OBJECTIVE FUNCTION VCET1 = 3.51239705E-01

INDEPENDENT VARIABLE      VALUE      GRADIENT COMPONENT      LOWER BOUND      UPPER BOUND
    P1T1      9.60000000E-01      2.23659125E-01      9.00000000E-01      9.80000000E-01
    P1T2      9.60000000E-01      -4.07175175E-02      9.00000000E-01      9.80000000E-01
    R3        2.00000000E+02      -3.75988248E-06      1.95000000E+02      2.05000000E+02

LOW VALUE LOCATED AT DISTANCE -6.09861798E-02 ALONG GRADIENT = LOWER BOUND 9.00000000E-01
OF INDEPENDENT VARIABLE P1T1

HIGH VALUE LOCATED AT DISTANCE 2.03287266E-02 ALONG GRADIENT = UPPER BOUND 9.80000000E-01
OF INDEPENDENT VARIABLE P1T1

WORST CASE COMPUTATION - HIGH VALUE
OBJECTIVE FUNCTION VCET1 = 3.53942361E-01

INDEPENDENT VARIABLE      VALUE      GRADIENT COMPONENT
    P1T1      9.80000000E-01      2.80346005E-01
    P1T2      9.56358967E-01      -4.73777819E-02
    R3        2.00000000E+02      -4.32150693E-06

WORST CASE COMPUTATION - LOW VALUE
OBJECTIVE FUNCTION VCET1 = 3.42972486E-01

INDEPENDENT VARIABLE      VALUE      GRADIENT COMPONENT
    P1T1      9.00000000E-01      1.36506341E-01
    P1T2      9.70923100E-01      -3.20266276E-02
    R3        2.00000001E+02      -3.01932958E-06

```

Abbildung 3.10.: Ergebnisse der 'Worst Case'-Analyse für die Spannung VCET1

3.6.4. A10: Optimierung

In diesem Beispiel werden die Zielfunktionen von PEC und VR3 bezüglich der unabhängigen Variablen P1 und EC, die mit Toleranzgrenzen versehen sein müssen, minimiert.

```

MODEL DESCRIPTION
MODEL 2N706A (TEMP) (B-E-C)
ELEMENTS
  CE, 1-E = Q1(5.,70.,J1)
  CC, 1-2 = Q1(8.,370.,J2)
  RB, B-1 = .3
  RC, C-2 = .015
  J1, 1-E = DIODE EQUATION(1.E-7,35.)
  J2, 1-2 = DIODE EQUATION(5.E-7,37.)
  JA, E-1 = .1*J2
  JB, 2-1 = P1*J1
DEFINED PARAMETERS
  P1 = 0.96 (0.98, 0.9)
FUNCTIONS
  Q1(A,B,C) = (A+B*C)
OUTPUTS
  VCE, VCC, J1
CIRCUIT DESCRIPTION
Example A10: Darlington pair, OPTIMIZATION
ELEMENTS
  EC , 1-6 = 10 (9.5, 10.5)
EIN , 1-2 = 1
R1 , 2-3 = 20
R2 , 6-5 = 5
R3 , 4-1 = 200
R4 , 8-1 = .4
R5 , 6-7 = 1
T1 , 3-4-5 = MODEL 2N706A
T2 , 4-8-7 = MODEL 2N706A
OPTIMIZATION
  (VR3,PEC/P1T1,P1T2,EC)
DEFINED PARAMETERS
  PEC = XEC (EC*IEC)
  GPEC= P3*DIEC+P5*DEC
  P3 = X3 (EC)
  P5 = X5 (IEC)
OUTPUTS
  VR3,VR4
RUN CONTROLS
  RUN OPTIMIZATION = 40
  RUN INITIAL CONDITIONS ONLY
END

```

Bei diesem Beispiel treten definierte Parameter sowohl als abhängige (PEC) als auch unabhängige Variablen (P1) auf. Für die abhängige Variable muß das vollständige Differential vorgegeben werden. Als unabhängige Variablen dürfen definierte Parameter nur dann verwendet werden, wenn sie Quellen erster und zweiter Ordnung beschreiben. Abbildung 3.11 zeigt einen Teil der Optimierungsergebnisse. Erklärungen über die Bedeutung der in den Ergebnissen vorkommenden Parametern findet man in den entsprechenden Kapiteln von [1, 2].

3.7. A11: Kleinsignal-Ersatzschaltung (AC)

Abbildung 3.12 zeigt eine Transistorschaltung, deren Frequenzverhalten in der Umgebung ihres Arbeitspunktes untersucht werden soll.

```

MODEL DESCRIPTION
MODEL 101M (B-C-E)
ELEMENTS
  RBC, BR-B = 50.
  RC, C -BR = 2.4E9
  RE, E -BR = 5.7E8
  CC, C -BR = Q1 (5.5E-12, .8, VCC, .3, 2.88E-7, J2, 3.12E-12)
  CE, E -BR = Q2 (3.5E-12, 17E-9, J1)
  JA, BR-E = .332 * J2
  JB, BR-C = .986 * J1
  J1, E -BR = DIODE EQUATION (60E-9, 38.4)
  J2, C -BR = DIODE EQUATION (54.3E-8, 38.4)
FUNCTIONS

```

3. SCEPTRE Beispiele

INITIAL VALUES OF OPTIMIZATION PARAMETERS:

OBJECTIVE FUNCTION	VR3
NUMBER OF INDEPENDENT VARIABLES	3
NUMBER OF RANDOM STEPS	0
INITIAL H MATRIX FACTOR	1.00000000E+00
CONVERGENCE CRITERION	1.00000000E-07
RANDOM STEP SIZE CONTROL	2.00000000E-01
MINIMUM FUNCTION ESTIMATE	0.00000000E+00
H MATRIX DETERMINANT	1.00000000E+00

INITIAL VALUES OF INDEPENDENT VARIABLES:

VARIABLE	NOMINAL VALUE	LOWER BOUND	UPPER BOUND
P1T1	9.60000000E-01	9.00000000E-01	9.80000000E-01
P1T2	9.60000000E-01	9.00000000E-01	9.80000000E-01
EC	1.00000000E+01	9.50000000E+00	1.05000000E+01

INITIAL APPROXIMATION TO H MATRIX

1.00000000E+00	0.00000000E+00	0.00000000E+00
0.00000000E+00	1.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00	1.00000000E+00

OPTIMIZATION RESULTS:

TERMINATION CONDITION	OPTIMIZATION RUN COMPLETED
OBJECTIVE FUNCTION	VR3
VALUE OF OBJECTIVE FUNCTION AT TERMINATION	5.65021325E-01
OPTIMIZATION PASS COUNTER	13
MAXIMUM PASSES SPECIFIED	40

INDEPENDENT VARIABLE VALUES AT TERMINATION:

VARIABLE	VALUE	GRADIENT COMPONENT	TRANSFORMED VALUE (RADIANS)	TRANSFORMED GRADIENT
P1T1	9.00000002E-01	4.50846113E-01	-2.86201981E-04	-5.16132196E-06
P1T2	9.00000004E-01	5.48451029E-01	-4.44554674E-04	-9.75265843E-06
EC	1.00000000E+01	1.28031073-147	1.57079633E+00	6.40155365-148

Abbildung 3.11.: Optimierungsergebnisse für die Spannung VR3

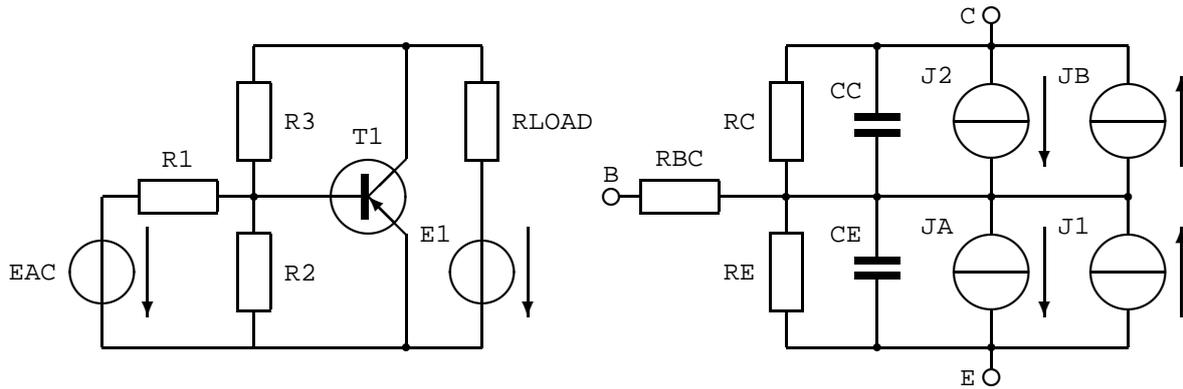


Abbildung 3.12.: Kleinsignalschaltung, Transistor-Ersatzschaltbild

$$Q1(A, B, C, D, E, F, G) = (A / (B - C)) * D + E * (F + G)$$

$$Q2(A, B, C) = (A + B * C)$$

OUTPUTS

VCC, VCE, NYQUIST, PLOT

CC, CE

JA, JB, DEGREES

J1, J2, RADIANS, PLOT

CIRCUIT DESCRIPTION

Example A11: small signal AC-Analysis

around a circuit's DC operating point

ELEMENTS

EAC, 1 - B = (0.2, 0.)

R1, B - 2 = 25E4

R2, 2 - 1 = 2000

R3, 2 - 3 = 11500

RLOAD, 3 - 4 = 100

E1, 1 - 4 = -20

T1, 2-3-1 = MODEL 101M

OUTPUTS

VR2, VR3, COMPLEX, PLOT

RUN CONTROLS

RUN INITIAL CONDITIONS

PRINT EIGENVALUES

PRINT EIGENVECTORS

RUN AC

INITIAL FREQUENCY = 1E5

FINAL FREQUENCY = 1E6

NUMBER FREQUENCY STEPS = 20

TYPE FREQUENCY RUN = LINEAR

END

Es wird eine kombinierte DC-/AC-Analyse durchgeführt. Dabei ist folgendes zu beachten: *bei der DC-Analyse werden alle AC-Quellen Null gesetzt. Umgekehrt werden bei der AC-Analyse alle DC-Quellen Null gesetzt.* Aus diesem Grund sollte man AC-Quellen von DC-Quellen mittels einer Kapazität oder eines großen Widerstandes (hier: R1) entkoppeln. Außerdem werden bei der AC-Analyse die Diodenquellen durch entsprechende Widerstände, die dem Arbeitspunkt des Transistors entsprechen, ersetzt.

3. SCEPTRE Beispiele

Teilausdruck der Simulationsergebnisse:

OUTPUTS BEFORE AC RUN

```
VCCT1      -1.3023927E+01
VCET1       3.6223429E-01
CCT1        2.3449600E-12
CET1        1.1243506E-09
JAT1        -1.8027600E-07
JBT1        6.5009336E-02
J1T1        6.5932389E-02
J2T1       -5.4300000E-07
VR2         -4.0836859E-01
VR3         1.2977793E+01
```

J9 SOURCE 'J1T1' IS REPLACED BY RESISTOR= 0.3949750

J9 SOURCE 'J2T1' IS REPLACED BY RESISTOR= 0.2400000E+13

LSNEW: COMPLEX EIGENVALUES

I	REAL	IMAGINARY	MAGNITUDE
1	-0.65292E+10	0.00000E+00	0.65292E+10
2	-0.33353E+08	0.00000E+00	0.33353E+08

ATRACE = -0.6562516388467601E+10

EIGENVALUE SUM = -0.6562516388467599E+10 0.0000000000000000E+00

WHEN COMPARING A*S WITH S*LAMBDA, THE WORST ERROR OBTAINED WAS
(-0.19073E-05, 0.00000E+00) WHEN MULTIPLYING MODAL MATRIX BY
INVERSE MODAL MATRIX

WORST ERROR IN DIAGONAL TERM WAS (0.00000E+00, 0.00000E+00)
WORST ERROR IN OFF-DIAGONAL TERM WAS (0.00000E+00, 0.00000E+00)

SINEW: MODAL MATRIX ... S

	1		2	
1	1.0000E+00	0.0000E+00	9.9999E-01	0.0000E+00
2	-2.0843E-03	0.0000E+00	3.9932E-03	0.0000E+00

SINEW: INVERSE MODAL MATRIX ...SI

	1		2	
1	6.5705E-01	0.0000E+00	-1.6454E+02	0.0000E+00
2	3.4296E-01	0.0000E+00	1.6454E+02	0.0000E+00

3.8. A13: Elektro-hydraulischer Stellantrieb

Die Schaltung nach Abb. 3.13 stellt einen elektro-hydraulischen Stellantrieb dar [6].

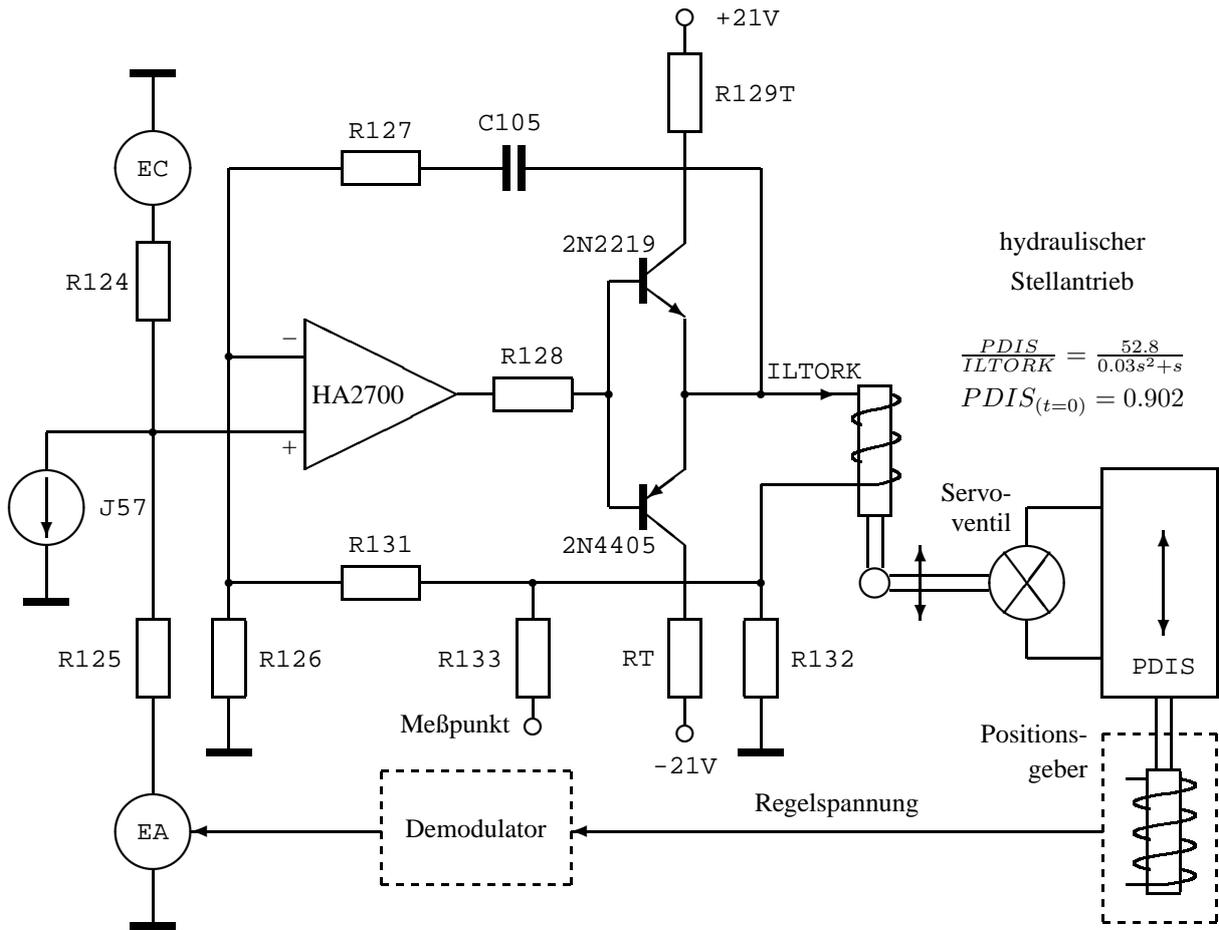


Abbildung 3.13.: Schaltung des Regelkreises

3. SCEPTRE Beispiele

Die Führungsgröße wird durch die Spannungsquelle EC=TABLE C (TIME) vorgegeben. Die Regeldifferenz wird mit Hilfe des Operationsverstärkers und der beiden Transistoren verstärkt und wirkt über einen Magnetanker auf ein Servoventil. Die Regelabweichung wird durch einen induktiven Geber in eine Regelspannung umgewandelt, demoduliert und dem Regler mit umgekehrten Vorzeichen mit Hilfe der gesteuerten Spannungsquelle EA wieder zugeführt.

Die Größe PDIS stellt die Regelgröße dar, deren Verlauf in Abbildung 3.15 zu sehen ist. Der Zusammenhang zwischen dem Spulenstrom ILTORK und PDIS wird durch eine Übertragungsfunktion beschrieben und kann in SCEPTRE direkt mit Hilfe der definierten Parameter und ihrer Ableitungen (PX1, PX2, DPX1, DPX2) berechnet werden.

Durch die Verwendung von Nullstromquellen lassen sich Spannungen zwischen beliebigen Knotenpunkten verlust- und rückwirkungsfrei messen. Mit der Nullstromquelle J57 erhält man z. B. das Differenzsignal VJ57 am Eingang des Operationsverstärkers.

In dem folgenden Listing wurde die Modellbeschreibung der Transistoren und des Operationsverstärkers aus Platzgründen weggelassen.

```
CIRCUIT DESCRIPTION
A13: torquer closed-loop step response test circuit parameters for
      a 2% step increase from 88% point in displacement schedule
ELEMENTS
  J57, 57-0 = 0,
  JT, 73-0 = 0,
  JDR, 72-0 = 0,
  JR1, 73-82 = 0,
  JR2, 83-0 = 0.,
  JOT, 71-0 = 0,
  O3, 57-70-71-0 = MODEL HA 2700,
  EC, 0-56 = TABLE C (TIME),
  R124, 56-57 = 23.7E3,
  R125, 57-11 = 20000,
  EA, 0-11 = XA (PDIS*1.843),
  R126, 70-0 = 10E3,
  C105, 73-81 = 4700E-12,
  R127, 81-70 = 800E3
  R128, 71-72 = 511
  R129T, 80-79 = 95.
  R131, 70-82 = 66.5E3
  R132, 82-0 = 10.
  R133, 83-82 = 10E3
  RTORK, 73-TI = 50
  LTORK, 82-TI = .2
  CTORK, 73-82 = .127E-6
  T4, 72-73-74 = MODEL 2N4405
  T3, 72-73-79 = MODEL 2N2219
  E13, 0-80 = 21
  E14, 78-0 = 21.
  RT, 74-78 = 60.
DEFINED PARAMETERS
  PX1 = .902, PX2 = 0.
  DPX1 = PX2, DPX2 = XXX ((-1./03)*PX2+1760.*ILTORK)
  PDIS = PX1
INITIAL CONDITIONS
  ILTORK = 0, VCCT3 = -2.1E1, VCCT4 = -2.09162E1, VCTORK = 0
OUTPUTS
  VJ57, EA, EC, VJOT, VJDR, VJT, ILTORK, PDIS, VJR2, VJR1
FUNCTIONS
  TABLE C = 0, -1.97, .01, -1.97, .01, -2.07, 1., -2.07
RUN CONTROLS
  LIST NODE MAP
  INTEGRATION ROUTINE = IMPLICIT
  STOP TIME = .45, MINIMUM STEPSIZE = 1E-20
END
```

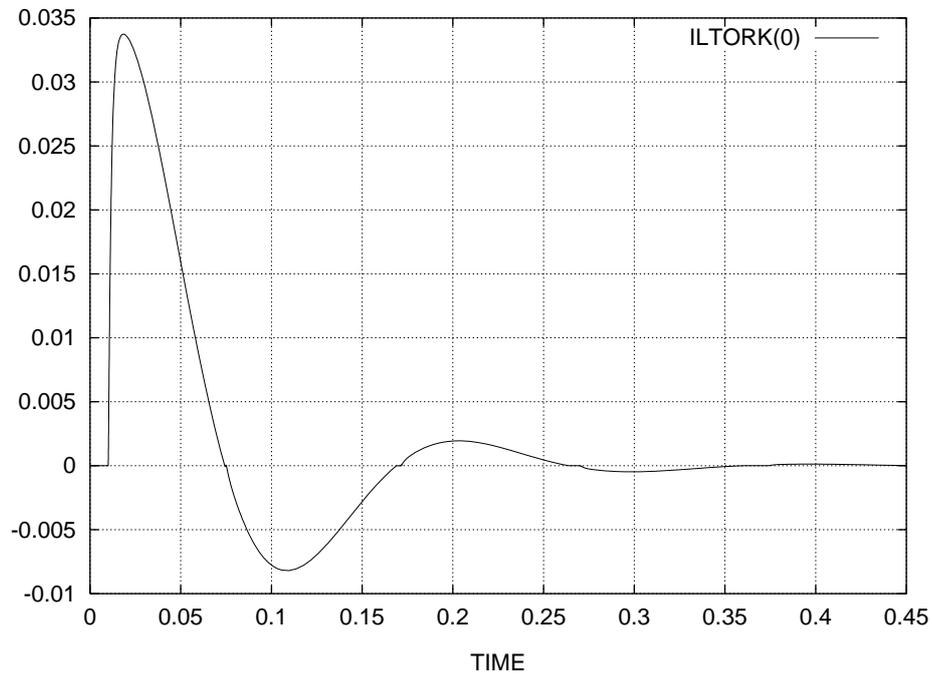


Abbildung 3.14.: Magnetstrom $ILTORK$ als Funktion der Zeit

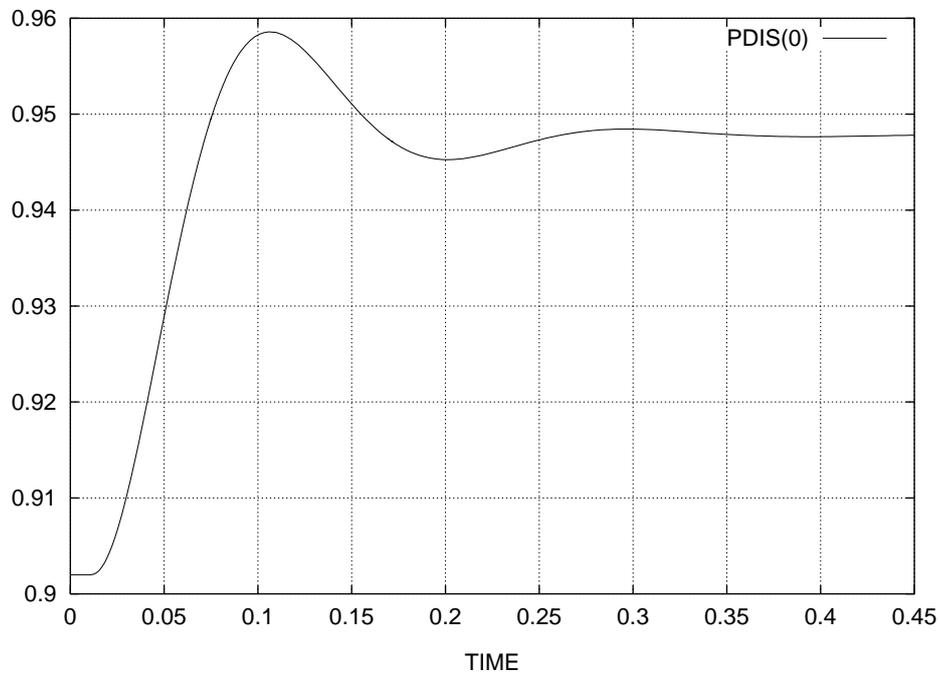


Abbildung 3.15.: Regelgröße $PDIS$ als Funktion der Zeit

4. Systeminformationen zur Benutzung von SCEPTRE

4.1. Programmorganisation

SCEPTRE läuft in mehreren Phasen ab (Abbildung 4.1):

1. Mit dem Programm EXEC1 werden die Eingabedaten verarbeitet. Dabei können permanente Modelle in die Modellbibliothek abgespeichert oder aus der gleichen Bibliothek in die zu untersuchende Schaltung eingefügt werden. Sind die Eingabedaten fehlerfrei, werden die Dateien `acsave` und `datasave` für die Ausführungsphase und einige FORTRAN-Unterprogramme (`SIMUL8`, `SIMTR`, `LOUTR`, ...) für die Datei `pgmsave` generiert.
2. Die Unterprogramme in der Datei `pgmsave` werden, evtl. mit eigenen FORTRAN Unterprogrammen (`user`), übersetzt.
3. Der Linker bindet die Objektmodule des FORTRAN Übersetzers mit den vorübersetzten Programmen aus den Bibliotheken `usrlib`, `x2lib` und `x3lib` zu einem ausführbaren Programm EXEC2 zusammen. Die Bibliotheken `x2lib` und `x3lib` gehören zu SCEPTRE, `usrlib` dient zur Aufnahme von vorübersetzten Benutzerunterprogrammen.
4. Das Programm EXEC2 wird ausgeführt. Dazu werden die Dateien `datasave` und `acsave` aus der 1. Phase benötigt. Neben der Standarddruckausgabe `liste2` werden die Ergebnisse mit dem Unterprogramm `POST` (siehe Kapitel 4.4) zwecks Nachverarbeitung auf eine direkte Zugriffsdatei `dirac` geschrieben.

Diese etwas umständlich erscheinende Programmorganisation hat einen entscheidenden Vorteil gegenüber direkt ausführbaren Programmen: das generierte Programm EXEC2 ist auf das zu untersuchende Problem zugeschnitten. Alle Vektoren und Arbeitsbereiche sind nur so groß dimensioniert, wie sie für diese Simulation benötigt werden. Ebenso sind nur die Unterprogramme vorhanden, die durch die Analyseart bestimmt werden. Weiterhin ist es sehr einfach, eigene Programme beliebig in eine SCEPTRE Simulation einzubinden und damit die Universalität von SCEPTRE zu erhöhen.

4.2. Aufruf unter Unix

SCEPTRE wird mit dem Kommando

```
sceptre dateiname
```

gestartet. Die SCEPTRE-Eingabedaten müssen in der Datei `dateiname.d` stehen.

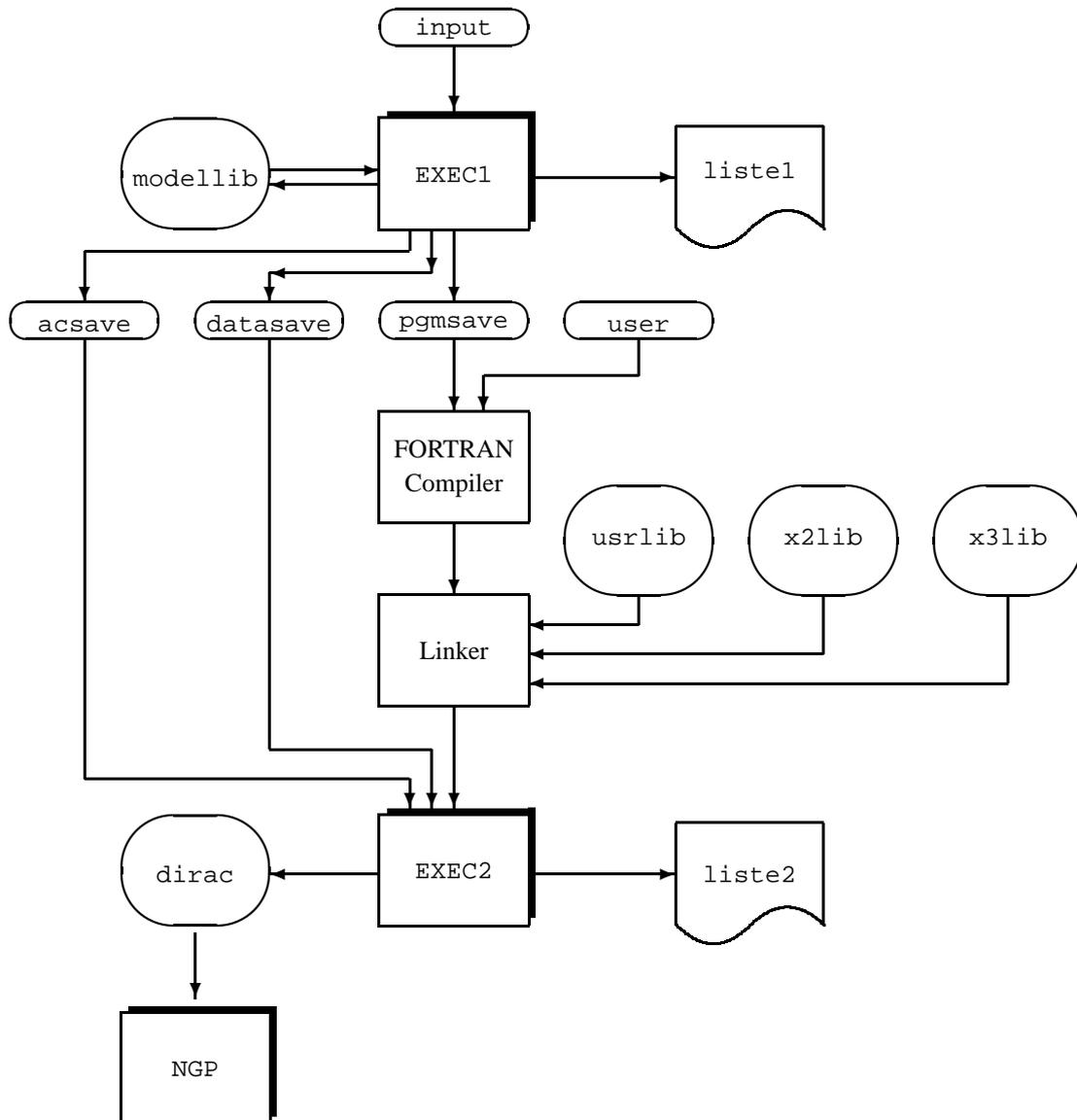


Abbildung 4.1.: Programm- und Datenfluß

4. Systeminformationen zur Benutzung von SCEPTRE

Nach Verarbeitung der Eingabedaten wird das Unix-Kommando `less` benutzt, mit dem man sich die Druckausgabe der ersten Phase (`dateiname.ls1`) anschauen kann. `less` wird mit der Taste `Q` beendet. Sind die Eingabedaten fehlerfrei, wird automatisch zur zweiten Phase übergegangen.

Standardmäßig werden bei der AC- und TR-Analyse die *laufenden* Ergebnisse als „Digitalanzeige“ mittels des ANSI-Treibers ausgegeben (siehe Kapitel 4.3.2). Falls der Ausgabebildschirm nicht dazu geeignet ist oder die Ausgabe unterdrückt werden soll¹, muß in den SCEPTRE Eingabedaten unter der Überschrift `RUN CONTROLS` die Anweisung `X PLOT DIMENSION = 0` eingefügt werden.

Am Ende jeder Simulation und vor jedem neuen Wiederholungslauf muß die Simulation mit dem Befehl `go` fortgeführt werden (Bildschirmnachrichten des Systems beachten).

Anschließend wird wieder mit `less` die Druckausgabe der zweiten Phase (`dateiname.ls2`) angezeigt (Taste `Q` zum Beenden).

Zum Schluß werden alle Dateien `dateiname.*` aufgelistet.

Der Postprozessor NGP (siehe Kapitel 4.5) wird mit dem Kommando

```
ngp dateiname
```

unter X11 aufgerufen.

4.3. Grafikschnittstellen

Standardmäßig liefert SCEPTRE Druckerplots, und zwar nur am Ende eines jeden Simulationslaufes. Das Programm bietet jedoch zwei Schnittstellen, um externe Grafiksoftware benutzen und zusätzlich *während* der Simulation die Ergebnisse grafisch oder tabellarisch („Digitalanzeige“) darstellen zu können. Dazu müssen die beiden Platzhalterunterprogramme `POST` und `INTPLT` durch eigene Routinen ersetzt werden, zweckmäßigerweise in der Benutzerbibliothek `usrlib`, da diese Routinen *vor* den Unterprogrammibliotheken `x2lib` und `x3lib` in der Phase 2 gelinkt werden müssen.

4.3.1. Das Unterprogramm INTPLT

Das Unterprogramm wird zu jedem Integrations- bzw. Frequenzschritt aufgerufen und beeinflußt entscheidend die Rechenzeit der Simulation. Aus diesem Grund werden die auszugebenden Daten nicht als Parameter im Aufruf übergeben, sondern es wird auf die internen COMMON-Blöcke zugegriffen. Aufruf:

```
subroutine INTPLT (nstep)
```

Die Variable `nstep` enthält die Nummer des aktuellen Integrations- bzw. Frequenzschrittes. Der Schrittzähler wird bei jedem Wiederholungslauf zurückgesetzt und kann damit zur Steuerung der Ausgabe benutzt werden. Es gilt:

`nstep = 1`: Beginn einer neuen Simulation, die Ergebnisse des ersten Schrittes liegen vor;

`nstep > 1`: die Ergebnisse des `nstep`. Schrittes liegen vor;

`nstep = 0`: Ende der Simulation; es liegen keine Ergebnisse vor.

¹verringert die Simulationszeit

Die Variable `nstep` darf nicht verändert werden!

Folgende COMMON-Bereiche werden benötigt:

```
COMMON /CNTRLS/  CNTRL
REAL*8          CNTRL(160)
```

Der Vektor CNTRL enthält allgemeine Steuerinformationen:

```
CNTRL(1)  (START) TIME (TR-Analyse)
CNTRL(2)  STOP TIME (TR-Analyse)
CNTRL(24) Anzahl der Ausgabevariablen NORQST
CNTRL(32) RUN/RERUN Nummer - 1.D0
CNTRL(71) 0.D0 = TR-Analyse, 1.D0 = AC-Analyse
CNTRL(72) INITIAL FREQUENCY (AC-Analyse)
CNTRL(73) FINAL FREQUENCY (AC-Analyse)
CNTRL(74) TYPE FREQUENCY RUN, ≤ 1.D0 = LINEAR, > 1.D0 = LOG
CNTRL(75) NUMBER FREQUENCY STEPS
CNTRL(20) X PLOT DIMENSION
CNTRL(21) Y PLOT DIMENSION
```

Die Namen der Ausgabevariablen erhält man von

```
COMMON /OUTPUT/ TPINDX
CHARACTER*8      TPINDX (NORQST)
```

Für `nstep > 0` stehen bei jedem Unterprogrammaufruf NORQST Werte zur Verfügung, und zwar in

```
COMMON /OUTDTA/ OUTBFR
REAL*8          OUTBFR (NORQST)
COMPLEX*16      OUTPAC (NORQST)
EQUIVALENCE     (OUTPAC(1), OUTBFR(1))
```

Der Name der unabhängigen Variablen TIME/FREQ steht unter TPINDX(1), die aktuellen Werte in OUTBFR(1) bzw. REAL(OUTPAC(1)).

Externe Dateien müssen über den COMMON-Block /TAPES/ und den installationsspezifischen Unterprogrammen QQQOPN und QQQCLS verwaltet werden.

Die Maximal- und Minimalwerte der unabhängigen Variablen TIME bzw. FREQ erhält man aus den Kontrollvariablen CNTRL(1), CNTRL(2) bzw. CNTRL(72), CNTRL(73). Entsprechende Werte für die abhängigen Variablen liegen vor der Simulation nicht vor.

4. Systeminformationen zur Benutzung von SCEPTRE

4.3.2. Beispiel zur Ausgabe mittels ANSI Treibers

Das folgende Programmbeispiel zeigt die Ausgabe der Ergebnisse mittels direkter Cursorsteuerung durch eine ANSI Escape-Zeichenfolge. Die Routine berücksichtigt die unterschiedlichen Formate bei der TR- und AC-Analyse. Ausgeführt wird diese "Digitalausgabe" nur dann, wenn die Kontrollvariable `cntrl(20) = X PLOT DIMENSION` ungleich 0 ist.

```
subroutine intplt (nstep)

COMMON /CNTRLS/  CNTRL
REAL*8          CNTRL(160)
COMMON /OUTPUT/ TPINDX
CHARACTER*8     TPINDX (2)
COMMON /OUTDTA/ OUTBFR
REAL*8         OUTBFR (2)
COMPLEX*16     OUTPAC (2)
EQUIVALENCE   (OUTPAC(1),OUTBFR(1))
character      wpos*20,wclear*4
character*12   wnumbr
logical        lfirst,lac
data lfirst    /.true./

if (nint(cntrl(20)).eq.0) go to 999

if (lfirst) then
  lfirst=.false.
  norqst=nint(cntrl(24))
  lac=nint(cntrl(71)).ne.0
  wclear(1:1)=char(27)
  wclear(2:4)='[2J'
  wpos(1:2)=wclear(1:2)
  print *, wclear
end if

if (nstep.eq.0) then
  pause
  print *, wclear
  go to 999
end if

if (nstep.eq.1) then
  do 10 i=1,norqst
    write (wpos(3:),'(i2.2,',';1H'',a)') i+3,tpindx(i)
    print *, wpos
10  continue
  end if

C nstep .ge. 1:

do 20 i=1,norqst
  if (lac) then
    write (wnumbr,'(1p,e12.4)') aimag(outpac(i))
    write(wpos(3:),'(i2.2,',';21H'',a)') i+3,wnumbr
    print *, wpos
    write (wnumbr,'(1p,e12.4)') real(outpac(i))
    write (wpos(3:),'(i2.2,',';9H'',a)') i+3,wnumbr
  else
```

```

        write (wnumbr,'(1p,e12.4)') outbfr(i)
        write (wpos(3:),'(i2.2,',';9H','a)') i+3,wnumbr
    end if
    print *, wpos
20 continue

999 continue
    return
end

```

4.3.3. Das Unterprogramm POST

Das Unterprogramm POST wird *nach jedem* Simulations-/Wiederholungslauf aufgerufen und wertet eine interne, unformatierte Datei INOUT aus. In ihr sind *alle* berechneten Ergebnisse enthalten. Standardmäßig kopiert POST diese Ergebnisse auf eine direkte Zugriffsdatei *dirac* (siehe Kapitel 4.4). Will man eigene Grafikprogramme an dieser Stelle ausführen, muß dieses Unterprogramm durch eine Routine gleichen Namens ersetzt werden, welches folgende Struktur aufweisen muß:

```

SUBROUTINE POST (lpost)

COMMON /TAPES/    NOUTP,INTAPE,LIBTP,INOUTP
COMMON /CNTRLS/  CNTRL(160)
DOUBLE PRECISION CNTRL,XRUNNO
    EQUIVALENCE (CNTRL(32),XRUNNO)

CHARACTER*8      HEADNG(9,11)
CHARACTER*8      OUTREQ(100)
DOUBLE PRECISION TEMP,XTYPE
REAL             BUFFER(3000)
    COMPLEX      CBUF(1500)
    EQUIVALENCE (CBUF(1),BUFFER(1))
CHARACTER*1      CK(100)
LOGICAL          LAC,lpost

    REWIND (INOUTP)
C nrunno=0: master run, > 0: rerun number
    NRUNNO=NINT(XRUNNO)-1
C AC or TR run ?
    READ (INOUTP) XTYPE
    LAC=.FALSE.
    IF (NINT(XTYPE).EQ.1) LAC=.TRUE.
C headings:
    READ (INOUTP) TEMP
    LENHDG=NINT(TEMP)
    IF (LENHDG.GT.0) THEN
        DO 10 N=1,LENHDG
            READ (INOUTP) (HEADNG(M,N),M=1,9)
10    CONTINUE
    END IF
C no. of output requests:
    READ (INOUTP) TEMP
    LNINDX=NINT(TEMP)
    DO 20 N=1,LNINDX
        READ (INOUTP) OUTREQ(N)
20 CONTINUE

```

4. Systeminformationen zur Benutzung von SCEPTRE

```

C AC formats (i.e. DEGREES, RADIANS, COMPLEX) :
  IF (LAC) READ (INOUP) (CK(I),I=1,LNINDX)
  KOUNT=0

C fill output buffer:
  30 READ (INOUP) NGRPS
  IF (NGRPS.EQ.0) GO TO 99
  KOUNT=KOUNT+NGRPS
  LEND=NGRPS*LNINDX
  IF (LAC) THEN
    READ (INOUP) (CBUF(K),K=1,LEND)
  ELSE
    READ (INOUP) (BUFFER(K),K=1,LEND)
  END IF
C CBUF/BUFFER now contains the results of the LNINDX output variables
C for NGRPS integration steps (NGRPS*LNINDX)
  GO TO 30

  999 CONTINUE
  WRITE (NOUP,608) KOUNT, NRUNNO
  608 FORMAT ('OPOST - NO. OF SOLUTION POINTS: ',I6,10X,' RUN: ',I3)
  REWIND (INOUP)
  9999 continue
  lpost=.false.
  RETURN

  END

```

Man beachte, daß bei Wiederholungsläufen die Datei INOUT mit den neuen Ergebnissen überschrieben wird. Sollen die Ergebnisse von verschiedenen Simulationsläufen in einer Grafik miteinander verglichen werden, muß der Benutzer entsprechende Vorkehrungen treffen.

Alle Ergebnisse werden mit einfacher Genauigkeit abgespeichert.

Bei der AC-Analyse enthält die Datei INOUP zusätzlich den Vektor CK. Pro Ausgabevariable beschreibt ein Zeichen die Art der AC Ergebnisse:

CK(i)	REAL(CBUF(i))	AIMAG(CBUF(i))
C	Realteil	Imaginärteil
R	Amplitude	Winkel [rad]
D	Amplitude	Winkel [degr]

4.4. Auswertung der Ergebnisse (“Postprocessing”)

Standardmäßig erzeugt SCEPTRE nach jeder Simulation eine direkte Zugriffsdatei (DA) `dirac`, die alle Ergebnisvektoren der TR- und AC-Analyse enthält, die der Benutzer unter der Eingabeüberschrift `OUTPUTS` angefordert hat. Der Inhalt dieser DA-Datei wird im Ausdruck `liste2` protokolliert:

```

directory of direct access file

no. symbol_name type length address last access

  1 DIRECTOR 0 0 1 01/06/92 18:50:48

```

```

2  HEADING      9      72      2 01/06/92 18:50:48
3  TIME        _0     4    3038    3 01/06/92 18:50:49
4  VCE         _0     4    3038    9 01/06/92 18:50:50
5  VCC         _0     4    3038   15 01/06/92 18:50:52
6  VC1         _0     4    3038   21 01/06/92 18:50:53
7  IR3         _0     4    3038   27 01/06/92 18:50:54
8  J1          _0     4    3038   33 01/06/92 18:50:55
9  TIME        _1     4    2870   39 01/06/92 18:51:14
10 VCE         _1     4    2870   45 01/06/92 18:51:15
11 VCC         _1     4    2870   51 01/06/92 18:51:16
12 VC1         _1     4    2870   57 01/06/92 18:51:17
13 IR3         _1     4    2870   63 01/06/92 18:51:19
14 J1          _1     4    2870   69 01/06/92 18:51:20
15 TIME        _2     4    2962   75 01/06/92 18:51:39
16 VCE         _2     4    2962   81 01/06/92 18:51:40
17 VCC         _2     4    2962   87 01/06/92 18:51:41
18 VC1         _2     4    2962   93 01/06/92 18:51:42
19 IR3         _2     4    2962   99 01/06/92 18:51:44
20 J1          _2     4    2962  105 01/06/92 18:51:45
21 ZZZZZZZZ    0      0    111 01/06/92 18:51:45

```

Die Namen der Ausgabevariablen in der direkten Zugriffsdatei enthalten zusätzlich zwei Informationen: den Typ der Ausgabevariablen und die Nummer des Wiederholungslaufes, aus dem diese Variable stammt. Der Name ist wie folgt aufgebaut:

variable t n

n enthält die Laufnummer (0 entspricht dem *master run*), das Zeichen *t* beschreibt den Typ der Ausgabevariablen entsprechend folgender Tabelle:

<i>t</i>	Typ der Ausgabevariablen
—	Variable stammt aus einer TR-Analyse
I	AC Analyse, Imaginärteil
D	AC Analyse, Phasenwinkel in Grad
R	AC Analyse, Phasenwinkel im Bogenmaß
A	AC Analyse, Amplitude bzw. Realteil

Am einfachsten lassen sich die Ergebnisse grafisch auswerten mit dem Programm NGP (siehe eigene Beschreibung). Für eigene Anwendungen bestehen die Möglichkeiten, entweder die internen Routinen INTPLT und POST durch eigene Unterprogramme zu ersetzen (siehe Kapitel 4.3) oder die DA-Datei selber zu verarbeiten.

4.5. Das Grafikprogramm NGP

Das Programm NGP dient zur grafischen Darstellung von Vektordaten aus der direkten Zugriffsdatei (DA) *dirac*. Es können maximal 6 Kurven in ein Diagramm gezeichnet werden. Die Auswahl der zu zeichnenden Ergebnisvektoren geschieht mittels Indizes statt symbolischer Namen. Die Darstellung der Kurven erfolgt in einem selbstskalierenden Koordinatensystem. Werden mehrere Kurven gleichzeitig dargestellt, richtet sich die Skalierung automatisch nach den Maxima und Minima aller Kurven. Enthält die DA-Datei Ergebnisvektoren aus Wiederholungsläufen (siehe 4.4), können diese Ergebnisse ebenfalls gleichzeitig dargestellt werden. Das Programm zeichnet automatisch jede *y*-Variable mit der dazugehörigen *x*-Variablen.

4. Systeminformationen zur Benutzung von SCEPTRE

4.5.1. Übersicht der Kommandos

Zu Beginn und nach jedem NGP-Kommando werden alle Vektoren mit ihrem Index, Minimum, Maximum, Erstellungsdatum und -uhrzeit aufgelistet. Der Index wird bei den Kommandos X, Y und S benötigt.

```
X n0
```

Als x-Werte werden die Werte der n0. Variable genommen. Es kann nur eine Variable angegeben werden.

```
Y n1 n2 ...
```

In einer Zeichnung werden gleichzeitig die Werte der n1., n2... Variable als y-Werte gezeichnet.

```
S n0 n1 n2 ...
```

Die n0. Variable wird als x-Wert, die n1., n2... werden als y-Werte benutzt. Dies ist die schnellste Art, eine Zeichnung zu generieren.

```
L
```

Listet die aktuellen Parameter auf.

```
D        oder        P
```

Erstellt eine vollständige Zeichnung entsprechend der aktuellen Parameter.

```
E        oder        Q
```

beendet das Programm.

Alle darzustellenden Größen können auf den Bereich $\pm 100\%$ normiert werden mit

```
scale on        und        scale off
```

Der Skalierungsfaktor wird beim `L`-Befehl ausgegeben.

4.5.2. NGP und GNUPLOT

Das NGP-Programm schreibt die zu zeichnenden Vektoren in ASCII-Dateien und erzeugt gleichzeitig eine Steuerdatei `temp.GNU`. Anschließend wird aus `NGP_GNU` heraus² das Grafikpaket `GNUPLOT` mit dem Kommando `gnuplot temp.GNU` gestartet und damit die ausgewählten Vektoren automatisch gezeichnet.

Diese Version läuft nur unter UNIX-ähnlichen Betriebssystemen.

Die Dateinamen für die ASCII-Dateien, die die Wertepaare der zu zeichnenden Vektoren enthalten, werden nach folgendem Schema gebildet:

```
yyyyyy_n.xxxxxx.GNU
```

Es bedeuten:

²mit dem Unterprogramm `SYSTEM`

yyyyyy SCEPTRE-Output-Variable für die Ordinate

_n Nummer des Laufes

xxxxxxx SCEPTRE-Output-Variable für die Abszisse.

Zur Auswahl des Ausgabegerätes gibt es einen neuen Befehl:

```
TERM 'cmd'
```

Für cmd muß ein gültiger GNUPLOT-Befehl angegeben werden, z. B. 'set term x11' für das X Window System, 'set term kc_tek40xx' für die farbige Ausgabe mittels der Kermit-Emulation (MS-DOS). Dieses Kommando überschreibt die Befehle, die in der Konfigurationsdatei `ngp.rc` (s. u.) unter der Kennung `term:` eingetragen sind.

Eine Papierkopie (*hardcopy*) zu der zuvor auf dem Bildschirm dargestellten Grafik läßt sich mit dem Befehl

```
hc
```

erstellen. Es werden die GNUPLOT-Befehle ausgeführt, die in der Konfigurationsdatei `ngp.rc` (s. u.) unter der Kennung `hc:` eingetragen sind.

Zusätzliche GNUPLOT-Befehle, die nur temporär wirken, können mit dem Kommando

```
cmd 'kommandos'
```

ausgeführt werden. Die unter 'kommandos' angegebenen Befehle, evtl. durch `;` getrennt, überschreiben die in der Konfigurationsdatei `ngp.rc` unter der Kennung `cmd:` eingetragenen Befehle.

4.5.3. Konfigurationsdatei `ngp.rc`

Beim Aufruf von NGP wird als Terminal und für die Hardcopy-Ausgabe der Terminaltyp `dumb` (ASCII-Terminal) angenommen. Die Handhabung von NGP läßt sich durch eine Konfigurationsdatei vereinfachen. Diese Datei enthält GNUPLOT-Kommandos zur Voreinstellung und muß im Verzeichnis `$HOME/etc` unter dem Namen `ngp.rc` angelegt werden. Die Datei enthält eine Kennung und anschließend einen oder mehrere GNUPLOT-Befehle (durch `;` getrennt). Kennung und Befehle dürfen 72 Zeichen nicht überschreiten. Die Reihenfolge der Kennungen ist beliebig. Folgende Kennungen sind vorgesehen:

`term:` setzt den Terminaltyp und evtl. die Ausgabedatei

`hc:` setzt den Terminaltyp und die Ausgabedatei für die Hardcopy-Ausgabe

`cmd:` GNUPLOT Befehle, die bei jeder Zeichnung ausgeführt werden.

Beispiel:

```
hc: set term post land; set out "|lpr"
term: set term x11
cmd: set grid; set time
```

Für die Papierkopie wird PostScript im Querformat ausgewählt und die erzeugte Datei direkt an das Druckkommando weitergegeben. Die Grafik wird auf einem X11-Terminal ausgegeben. Jede Grafik wird mit Gitterlinien und Datum/Uhrzeit versehen.

5. Fortsetzungsläufe (CONTINUE)

Es ist möglich, eine TR-Analyse fortzusetzen (siehe Abschnitt 1.14, Seite 28). Dazu muß man eine Datei anlegen, die mindestens folgende Anweisungen enthalten muß:

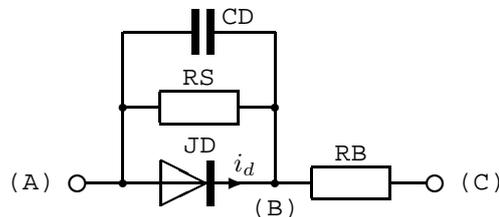
```
CONTINUE
RUN CONTROLS
    STOP TIME = ....
END
```

Für ... ist die neue Simulationszeit einzusetzen, die größer sein muß als die des vorangegangenen Laufes. Gestartet wird der Fortsetzungslauf, in dem man SCEPTRE mit dieser Datei aufruft. Weiterhin ist darauf zu achten, daß die Dateien `pgmsave.for` und `datasave.dat` von dem fortzusetzenden Lauf vorhanden sein müssen. Da diese Dateien durch jeden SCEPTRE-Lauf überschrieben werden, muß der Fortsetzungslauf unmittelbar der fortzusetzenden TR-Analyse folgen¹.

¹oder man erzeugt eine Kopie beider Dateien

A. Dioden- und Transistor-Ersatzschaltbilder

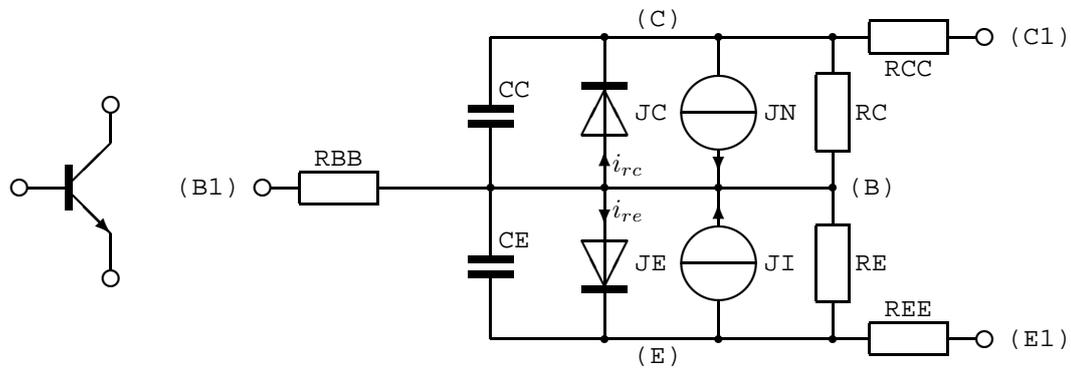
Zur Simulation von Halbleiterelementen gibt es verschiedene Modelle. In den nachfolgenden Abbildungen A.1, A.2 und A.3 werden die Ersatzschaltungen einer Diode und bipolaren pnp/npn-Transistoren sowie ihre Darstellung in SCEPTRE beschrieben. Die Ersatzschaltungen beruhen auf dem Ebers-Moll Modell. Die in den SCEPTRE-Anweisungen kursiv gedruckten Parameter müssen aus den Datenblätter der entsprechenden Halbleiter entnommen werden. Dabei ist auf ein konsistentes Maßeinheitensystem zu achten.



Parameter:	$i_s, \Theta, R_B, R_S, C_0, V_0, n, K$	MODEL <i>name</i> (A-C) ELEMENTS
	$i_d = i_s(e^{\Theta V_{JD}} - 1)$	JD, A-B = DIODE EQUATION (<i>i_s</i> , Θ)
	$C_D = \frac{C_0}{(V_0 - V_{JD})^n} + K(i_d + i_s)$	CD, A-B = Q1 (<i>C₀</i> , <i>V₀</i> , <i>VJD</i> , <i>n</i> , <i>K</i> , JD , <i>i_s</i>)
		RB, B-C = <i>R_B</i> RS, A-B = <i>R_S</i>
		FUNCTIONS Q1 (A, B, C, D, E, F, G) = (A / (B-C) ** D + E * (F+G))

Abbildung A.1.: SCEPTRE Diodenersatzschaltbild

A. Dioden- und Transistor-Ersatzschaltbilder



Parameter:	$i_{es}, \Theta_e, \alpha_i, i_{cs}, \Theta_c, \alpha_n,$ $C_{0e}, V_{0e}, K_e, n_e, C_{0c}, V_{0c}, K_c, n_c,$ $R_{BB}, R_{EE}, R_{CC}, R_E, R_C$	MODEL name (B1-E1-C1) ELEMENTS
$i_{re} = i_{es}(e^{\Theta_e V_{BE}} - 1)$		JE, B-E = DIODE EQUATION (i_{es}, Θ_e)
$i_{rc} = i_{cs}(e^{\Theta_c V_{BC}} - 1)$		JC, B-C = DIODE EQUATION (i_{cs}, Θ_c)
$i_i = \alpha_i i_{rc}$		JI, E-B = $\alpha_i * JC$
$i_n = \alpha_n i_{re}$		JN, C-B = $\alpha_n * JE$
$C_e = \frac{C_{0e}}{(V_{0e} - V_{BE})^{n_e}} + K_e(i_{re} + i_{es})$		CE, B-E = Q1 ($C_{0e}, V_{0e}, V_{CE}, n_e, K_e, JE, i_{es}$)
$C_c = \frac{C_{0c}}{(V_{0c} - V_{BC})^{n_c}} + K_c(i_{rc} + i_{cs})$		CC, B-C = Q1 ($C_{0c}, V_{0c}, V_{CC}, n_c, K_c, JC, i_{cs}$)
		RBB, B1-B = R_{BB} REE, E1-E = R_{EE} RCC, C1-C = R_{CC} RE, E-B = R_E RC, C-B = R_C
		FUNCTIONS Q1 (A, B, C, D, E, F, G) = (A / (B-C) ** D + E * (F+G))

Abbildung A.2.: SCEPTRE npn Transistor-Ersatzschaltbild

B. Programmkapazität

Tabelle B.1 enthält die Programmgrenzen von SCEPTRE. Wird eine Grenze überschritten, wird der Lauf mit einer entsprechenden Fehlernachricht abgebrochen.

Programmgröße	Maximum
Netzwerkelemente	300
Knotenpunkte	301
Ableitungen von Quellen DE, DJ	50
Definierte Parameter P	200
Definierte Parameter W	50
Definierte Parameter DP	200
Gegeninduktivitäten M	50
Argumente in EQUATION-Anweisungen	50
Tabellenänderungen in Modellen	15
EQUATION-Änderungen in Modellen	15
Ausgabeunterdrückungen in Modellen	10
Ausgabevariablen	100
Anfangsbedingungen	100
EQUATION-Anweisungen	insgesamt ca. 5000 Zeichen
Folgezeilen für EQUATION-Anweisung	20
Tabellen	insgesamt 800 Wertepaare
Abbruchbedingungen TERMINATE, STOP	10
Permanent gespeicherte Modelle	250
Externe Modellknotenpunkte	25
Interne Modellknotenpunkte	301
AC-Quellen 1. plus 2. Ordnung	50
Unabhängige AC-Strom- und Spannungsquellen	50
Kommentarkarten	11

Tabelle B.1.: Programmkapazität

C. Übersicht aller Anweisungen und internen Variablen

Tabelle C.1 enthält eine Übersicht aller SCEPTRE-Anweisungen und internen Variablen, die einen Bezug zu diesen Anweisungen haben.

SCEPTRE-Anweisungen	Variable	Index
START TIME	TIME	1
STOP TIME	XSTOPT	2
MAXIMUM STEP SIZE	XXMISS	7
INTEGRATION ROUTINE	XIR	4
STARTING STEP SIZE	XTISSS	5
MINIMUM STEP SIZE	XMNISS	6
MINIMUM ABSOLUTE ERROR	XMNAIE	8
MAXIMUM ABSOLUTE ERROR	XXMAIE	9
RUN INITIAL CONDITIONS	XIC	10
COMPUTER TIME LIMIT	XXMERT	3
RUN INITIAL CONDITIONS ONLY	XTRANS	34
MINIMUM RELATIVE ERROR	XXMRIE	13
MAXIMUM RELATIVE ERROR	XXMRIE	14
RELATIVE CONVERGENCE	XICRER	17
ABSOLUTE CONVERGENCE	XICAER	18
X PLOT DIMENSION	XXDIMS	20
Y PLOT DIMENSION	XXYDIMS	21
MAXIMUM PRINT POINTS	XXMXPRT	19
NO ELEMENT SORT		
MAXIMUM INTEGRATION PASSES	XXMPAS	12
NEWTON-RAPHSON PASS LIMIT	XXMXP	16
PUNCH BINARY CARDS		
FULIST		
PUNCH PROGRAM		
WRITE SIMUL8 DATA		
COMPUTER SAVE INTERVAL	XSAVE	35
VECTOR EQUATIONS		
WRITE DEBUG		
PRINT B MATRIX		
SOLUTION TIME LIMIT	XXMXT	37
PLOT INTERVAL	XPLTI	38
PRINT INTERVAL	XPRTI	39
LIST NODE MAP		

Fortsetzung nächste Seite

C. Übersicht aller Anweisungen und internen Variablen

SCEPTRE-Anweisungen	Variable	Index
IC FOR RERUNS	XSAVIC	53
RUN IC VIA IMPLICIT	XTRNIC	55
USE DIFFERENCED JACOBIAN	XDIFF	61
USE SYMBOLIC JACOBIAN	XSMBL	62
EXECUTE SETUP PHASE ONLY		
RUN AC	XAC	71
INITIAL FREQUENCY	XINFRQ	72
FINAL FREQUENCY	XFNFRQ	73
TYPE FREQUENCY	XTYPFQ	74
NUMBER FREQUENCY STEPS	XNFRQS	75
PRINT A MATRIX		
PRINT EIGENVALUES		
PRINT EIGENVECTORS		
RUN MONTE CARLO	XNMCPS	121
RUN OPTIMIZATION	XNOPPS	131
RUN WORST CASE	XWCLHN	151
FREQUENCY	FREQ	79
INITIAL RANDOM NUMBER	XRANNO	122
DISTRIBUTION	XDISTR	125
LIST MONTE CARLO DETAILS		
USE FIXED AC MATRIX IN RERUNS	XACRE	85
IMPULSE RESPONSE BUFFER	XHAVE	87
INPUT FUNCTION BUFFER	XIAVE	88
COMPRESSION CRITERION	XCMPCR	89
COMPRESSION COUNT	XCMPCT	90
LIST OPTIMIZATION DETAILS		
PUNCH OPTIMIZATION RESULTS		
OPTIMIZATION CRITERION	XOCPR	134
OPTIMIZATION RANDOM STEPS	XNOPRS	135
RANDOM STEP SIZE CONTROL	XRNSSC	136
INITIAL H MATRIX FACTOR	XHMFAC	137
RUN SENSITIVITY	XSENST	141
MINIMUM FUNCTION ESTIMATE	XMNFES	138

Tabelle C.1.: SCEPTRE-Anweisungen und interne Variablen

Zugriff auf die internen Variablen erhält man über den COMMON-Bereich /CNTRLS/. Man kann die Variable sowohl mit ihrem Namen als auch über den Vektor CTROLS und dem entsprechenden Index ansprechen.

```

IMPLICIT REAL*8(A-J,L-M,O-Z),INTEGER*4(K,N)
COMMON /CNTRLS/
*   TIME ,XSTOPT,XXMERT,XIR ,XTISSS,XMNISS,XXMISS,
*   XMNAIE,XXMAIE,XIC ,XRERNO,XXXPAS,XXMNRIE,XXMXRIE,
*   XMNISS,XXMIXP,XICRER,XICAER,XXMOTP,XXDIMS,XYDIMS,
*   XICPAS,XXNOPRQ,XXNTNDX,XXNHEAD,XXNDFEQ,XERT ,XXNRERN,
*   XCNTNU,XXSTPNO,XXPASNO,XXRUNNO,XXSTPSZ,XXTRANS,XXSAVE ,
*   XTMON ,XXMSLT,XPLTI ,XPRTI ,XXMCPUT,SP1(3),XXWRTSM,
*   SP2(8),XXSAVIC,XXGER ,XXTRNIC,SP3(5),XXDIFF ,XXSYMBL,
*   SP4(8),XXAC ,XXINFRQ,XXFNFRQ,XXTYPFQ,XXNFRQS,SP5(3),
*   FREQ ,XXNWDBM,SP6(3),XXDERIV,XXACRE ,XXNCONV,XXHAVE ,

```

```

*   XIAVE ,XCMPCR,XCMPCT,XNEY ,XNE7DT,XNBC ,XNBR ,
*   XNBL ,XNLC ,XNLR ,XNLL ,XNJX ,XNJ8DT,XNJ0 ,
*   XNJ9 ,XNJ8AC,XNE7AC,XNM33 ,XNM36 ,XNM66 ,XNDP ,
*   XNDPD ,XNELTS,XLSW ,XLSWP ,SP7(2),XICPRT,SP8(5),
*   XNMCPS,XRANNO(2),SP9,XDISTR,SPA(5),XNOPPS,SPB(2),
*   XOPCR ,XNOPRS,XRNSSC,XHMFAC,XMNFES,SPC(2),XSENST,
*   XADJNT,SPD(8),XWCLHN,SPE(9)
DIMENSION CTROLS(160)
EQUIVALENCE (CTROLS,TIME)

```

Die Werte der Kontrollvariablen dürfen nicht verändert werden.

Literaturverzeichnis

- [1] D. Becker. *Extended SCEPTRE Vol. 1, User's Manual*.
AFWL-TR-73-75¹ (NTIS²: ADA 009594)
- [2] D. Becker. *Extended SCEPTRE Vol. 2, Mathematical Formulation*.
AFWL-TR-73-75 (NTIS: ADA 009595)
- [3] J.C. Bowers, S.R. Sedore. *SCEPTRE: A Computer Program for Circuit and System Analysis*.
Englewood Cliffs, N.J., Prentice-Hall, Inc., 1971
- [4] R. W. Jensen, M. D. Lieberman. *IBM Electronic Circuit Analysis Program Techniques and Applications*.
Englewood Cliffs, N.J., Prentice-Hall, Inc., 1968
- [5] *Advanced Statistical Analysis Program (ASTAP) General Information Manual*.
IBM GH20-1271-0
- [6] R. W. Jensen, L. P. McNamee. *Handbook of Circuit Analysis Languages and Techniques*.
Englewood Cliffs, N.J., Prentice-Hall, Inc., 1976
- [7] James C. Bowers, et al. *A survey of Computer-Aided-Design & Analysis Programs*
AFAPL-TR-76-33 (NTIS: AD-A 026567)
- [8] Soo Young Shin. *A Survey of Computer-Aided Electronic Circuit Analysis Programs*
NTIS: AD-A 009185
- [9] H. Spiro. *Simulation integrierter Schaltungen durch universelle Rechnerprogramme: Verfahren und Praxis der rechnergestützten Simulation nichtlinearer Schaltungen*.
Verlag Oldenbourg, München , 1985
- [10] W. Kruckow, T. Vetter. *Digitale Nachbildung des dynamischen Verhaltens einer Asynchronmaschine mit Kurzschlußläufer*.
EW-Bericht Nr. 74/84, Institut für Elektrische Energiewandlung, FB 17, Technische Hochschule Darmstadt, D-64283 Darmstadt
- [11] W. Kruckow. *Käfigläufermotor und Frequenzumrichter mit Stromzwischenkreis für höhere Drehzahlen*.
Fortschrittberichte VDI, Reihe 21: Elektrotechnik, Nr. 27, VDI Verlag
- [12] A. Schindler. *Betriebsverhalten und Verluste von Asynchron-Käfigläufermotoren beim Betrieb am Transistor-Pulswechselrichter*.
Dissertation am Fachbereich Elektrische Energietechnik der Technischen Hochschule Darmstadt, Juni 1985
- [13] U. Tenne. *Auswirkungen einer welligen Zwischenkreisspannung auf das Betriebsverhalten umrichter gespeister Asynchronmaschinen*.
Dissertation am Fachbereich Elektrische Energietechnik der Technischen Hochschule Darmstadt, April 1986

¹Air Force Weapons Laboratory, Kirtland, AFB, NM 87117, USA

²National Technical Information Service, Springfield, VA 22151, USA

- [14] W. Merz. *Untersuchung transienter Beanspruchung der Hochspannungskabel für die Versorgung der HERA HF-Sender.*
DESY³ M-88-16
- [15] J. I. Lubell. *Transmission Line Modeling for Use with Circuit/System Analysis Programs.*
AFWL-TR-73-128 (NTIS: AD 913-800)
- [16] P. Krehl, W.-R. Novender. *A Graphical and Analytical Method to Determine the Transient Response for an Ideal Transmission Line, Loaded by a Time-Varying Impedance.*
IEEE Transactions on Plasma Science, Vol. PS-13, No. 2, April 1985
- [17] E. N. Lorenz. *Deterministic Nonperiodic Flow.*
J. Atmos. Sci. Vol. 20, p. 130, 1963
- [18] R. Hirota, K. Suzuki. *Theoretical and Experimental Studies of Lattice Solitons in Nonlinear Lumped Networks.*
Proceedings of the IEEE, Vol. 61, No. 10, October 1973
- [19] H. Mais, W.-R. Novender. *Solitons in Nonlinear Networks - Simulations using SCEPTRE.*
8-mm Film, DESY, 1987
- [20] W. Popp. *Verfahren für den Entwurf kopplungsfreier Dämpfungsentzerrer.*
Dissertation am Fachbereich 19, TH Darmstadt, 1978
- [21] F.-J. Decker. *Zur experimentellen Bestimmung der Dichteverteilung von Elektronenringen am Wake Field Transformator Experiment bei DESY.*
DESY M-87-11
- [22] C. B. Frye, Jr., M. J. Apfelbaum. *Mixed Domain Transient Analysis of Large Non-Linear Networks.*
Tenth Annual Allerton Conference on Circuit and System Theory, October 1972
- [23] W. Kaplan. *Ordinary Differential Equations.*
Addison Wesley, Reading, Mass., 1958, Section 10-2, pp. 400–401.
- [24] eb.da, Section 10-4, “Heun’s Method”, pp. 402–403
- [25] K. G. Ashar, H. N. Ghosh, A. W. Aldridge, L. J. Patterson. *Transient Analysis and Device Characterization of ACP Circuits.*
IBM Journal of Research and Development, Vol. 7, p. 218, 1963
- [26] P. E. Chase. *Stability Properties of Predictor-Corrector Methods for Ordinary Differential Equations.*
Journal A.C.M 9, October 1962, pp. 457–468
- [27] R. J. Kuhler. *Application of Generator Analysis Methods.*
AFAPL-TR-77-31 (NTIS: AD A042071)
- [28] Wm. C. Davidon. *Variable Metric Method for Minimization.*
AEC Research and Development Report ANL-5990 (REV) 1959
- [29] M. J. Box. *A Comparison of Several Current Optimization Methods, and the Use of Transformation in Constrained Problems.*
The Computer Journal 9, pp. 67-77, 1966
- [30] M. J. Box, D. Davies, W. G. Swann. *Non-Linear Optimization Techniques.*
Imperial Chemical Industries, Ltd. Monograph No. 5, 1969

³Deutsches Elektronen Synchrotron, 2000 Hamburg 52

Literaturverzeichnis

- [31] *Variable Metric Minimization*.
SHARE Routine No. 1117, AN Z013, A3. SHARE Inc., Suite 750, 25 Broadway New York, NY 10004
- [32] J. C. Bowers, J. E. O'Reilly, G. A. Shaw. *SUPER-SCEPTRE – A Program for the Analysis of Electrical, Mechanical, Digital and Control Systems*. University of South Florida, Tampa, Florida 33620, May 1975 (NTIS: AD A011348)
- [33] Otto Justus. *Dynamisches Verhalten elektrischer Maschinen — Eine Einführung in die numerische Modellierung mit PSPICE*.
Verlag Vieweg, 1993
- [34] W. A. Cordwell. *Transistor and Diode Model Handbook*.
AFWL-TR-69-44 (NTIS: AD 862556)
- [35] Y. C. Liang, V. J. Gosbell. *A Versatile Switch Model for Power Electronics SPICE2 Simulations*.
IEEE Transactions on Industrial Electronics, vol. 36, no. 1, February 1989, p. 86
- [36] Lawrence J. Giacoletto. *Simple SCR and TRIAC PSPICE Computer Models*.
IEEE Transactions on Industrial Electronics, vol. 36, no. 3, August 1989, p. 451
- [37] F. Javier Gracia, Fernando Arizti, F. Javier Aranceta. *A Nonideal Macromodel of Thyristor for Transient Analysis in Power Electronic Systems*.
IEEE Transactions on Industrial Electronics, vol. 37, no. 6, December 1990, p. 514
- [38] Vineeta Agrawal, Anant K. Agarwal, Krishna Kant. *A Study of Single-Phase to Three-Phase Cycloconverter using PSPICE*.
IEEE Transactions on Industrial Electronics, vol. 39, no. 2, April 1992, p. 141
- [39] H. A. Nienhaus, J. C. Bowers, M. S Ziemacki. *A Computer Model for a High Power SCR*.
AFAPL-TR-75-106⁴ (NTIS: ADA 022375)

⁴Air Force Aero Propulsion Laboratory