Jürgen Schwarz

Behandlung von Polynomen – Teil 5*

Stichwörter:

Polynom, HP-BASIC, Laplace-Transformation, Approximation, Ausgleichsrechnung

Hardware:

HP 9845 B mit SP ROM

Im ersten Teil dieser Serie wurden SUB-Programme für die rationalen Operationen mit Polynomen abgeleitet, der zweite Teil beinhaltete die Nullstellenbestimmung von Polynomen und im dritten Teil wurden Algorithmen zur Berechnung des größten gemeinsamen Teilers von zwei Polynomen und einige Bemerkungen zur Datensicherung angegeben. Der vierte Teil behandelte das HORNER'sche Schema und seine Implementationen. Während in den ersten drei Teilen dieser Serie einige Grundlagen zur Bearbeitung und Bestimmung der Eigenschaften von Polynomen abgeleitet wurden, soll nun weiter auf die Anwendung dieser SUB-Programme eingegangen werden. Wie bereits im Abschnitt 1 erwähnt, werden die Polynome in den hier zu besprechenden Anwendungsfällen als Approximations- und Ausgleichsfunktionen verwendet. Kommt dabei direkt eine Polynomapproximation zum Einsatz, so genügen i. allg. die bereits abgeleiteten SUB-Programme zur weiteren Verarbeitung. Anders sieht es aus, wenn die Polynome im Bildbereich der LAPLACE-Transformation verwendet werden.

9 Polynome und LAPLACE-Transformation

An dieser Stelle sollen nun nicht die Grundlagen der LAPLACE-Transformation niedergelegt werden. Darüber existiert viel Literatur, z. B. [13 – 15]. Hier soll auch eine Beschränkung auf die Fälle erfolgen, in denen im Bildbereich ganzrationale und insbesondere gebrochenrationale Funktionen entsprechend Gl. (43) auftreten. Trotz dieser Einschränkung wird aber damit bereits ein großer Teil der praktisch bedeutsamen Funktionen eingeschlossen, so alle ganzrationalen Polynome, die trigonometrischen, Hyperbel- und Exponentialfunktionen im Zeitbereich und auch bestimmte Kombinationen davon.

9.1 Transformation aus dem Zeit- in den Bildbereich

Die Transformation aus dem Zeitbereich in den Bildbereich der LAPLACE-Transformation bereitet i. allg. die wenigsten Schwierigkeiten. Mit Hilfe von Tafelwerken, die die Korrespondenzen $F(p)$ •—○ $f(t)$ enthalten, kann man die Transformation einfach durchführen. Dies sei an einem wichtigen Beispiel erläutert.

Untersucht man das thermische Verhalten von Halbleiterbauelementen bei einer Erregung durch einen Leistungssprung, so müssen meist Messungen über einen großen Zeitbereich durchgeführt werden (> 1 : 1000). Die dabei erhaltenen Meßergebnisse werden punktweise (zu definierten Zeitpunkten) gespeichert. Aus diesen Stützstellen werden dann geeignete Ausgleichskurven errechnet. Als Ansatz haben sich Exponentialsummen

$$f(t) = R_0 + \sum_{i=1}^{m} R_i(1 - e^{-t/\tau_i}) \qquad (77)$$

bewährt. Dieser Ansatz hat mehrere Vorteile. Zuerst verhält er sich „gutartig". Weil sich die einzelnen Terme streng monoton verhalten, sind zufällige Extremwerte, wie sie bei Polynomansätzen häufig auftreten, i. allg. auszuschließen. Ein weiterer Vorteil ist darin zu sehen, daß der Ansatz auch für große Zeitbereiche geeignet ist und ohne weiteres Extrapolationen über die Meßgrenzen hinaus gestattet. Für große Zeiten strebt er gegen einen festen Endwert. Außerdem entspricht der Ansatz auch der Theorie der Wärmeleitung. Wird die Wärmeleitungsgleichung in Ortsrichtung diskretisiert, dann sind die Lösungen Gleichungen der Art von Gl. (77) [16]. Auch bei der Anwendung des BEUKEN-Modells [17] trifft dies zu. Außer der bereits zitierten Arbeit [2] seien mit [18 – 19] noch zwei Arbeiten aus der deutschsprachigen Literatur zitiert, die sich mit der Exponentialsummenzerlegung nach Gl. (77) befassen.

Zur Transformation von Gl. (77) in den Bildbereich notieren wir zuerst drei erforderliche Korrespondenzen [15]

$$c_1 f_1(t) + c_2 f_2(t) \quad ○\!\!-\!\!\!-\!\!\bullet \quad c_1 F_1(p) + c_2 F_2(p) \qquad (78)$$

$$\epsilon(t) = \begin{cases} 0 \text{ für } t < 0 \\ 1 \text{ für } t > 0 \end{cases} \quad ○\!\!-\!\!\!-\!\!\bullet \quad \frac{1}{p} \qquad (79)$$

$$\exp(\pm \alpha t) \quad ○\!\!-\!\!\!-\!\!\bullet \quad \frac{1}{p \pm \alpha}\,. \qquad (80)$$

Nach Umschreiben von Gl. (77) in

$$f(t) = (R_0 + \sum_{i=1}^{m} R_i) - \sum_{i=1}^{m} R_i e^{-t/\tau_i} \qquad (81)$$

ergibt sich für F(p)

$$F(p) = (R_0 + \sum_{i=1}^{m} R_i)\frac{1}{p} - \sum_{i=1}^{m} \frac{R_i}{p - \frac{1}{\tau_i}} \qquad (82)$$

---

*Teil 4: s. CAL-Comp. Anw. Lab. **3** (1985) 224.

Jürgen Schwarz, Wiesbadener Straße 59 C, 1000 Berlin 33

Diese Gleichung kann durch Erweitern mit τ_i und Zusammenfassen der Elemente, die R_i enthalten, in

$$F(p) = \frac{1}{p}\left[R_0 + \sum_{i=1}^{m} \frac{R_i}{1+p\,\tau_i}\right] \quad (83)$$

umgewandelt werden.

Im allgemeinen ist es zweckmäßig, zur Berechnung der Ausgleichsfunktion nach Gl. (77) nicht direkt die gemessenen Temperaturen heranzuziehen. Günstiger ist es, den gemessenen Temperaturverlauf auf eine Bezugstemperatur T_A und den bereits erwähnten Leistungssprung P_J mit der Gleichung

$$T^*_{XA}(t) = \left.\frac{T_X(t) - T_A}{P_J}\right| \; T_A, P_J = \text{const.} \quad (84)$$

zu beziehen. T^*_{XA} hat dann die Dimension eines thermischen Widerstandes [K/W]. Auch die Elemente R_i von Gl. (77) haben dann diese Einheit. F(p) stellt demzufolge den bezogenen Temperaturverlauf bei einem Verlustleistungssprung P_J bei $t=0$ dar. Wird dieser Sprung bei $t=0$ auch aus Gl. (83) „herausgekürzt", so entspricht die verbleibende Summe

$$Z_{GA}(p) = \frac{z_{GA}(p)}{n_{GA}(p)} = R_0 + \sum_{i=1}^{m} \frac{R_i}{1+p\,\tau_i} \quad (85)$$

der thermischen Impedanz einer Ersatzschaltung nach Bild 4.

Zur Berechnung von $Z_{GA}(p)$ dient das SUB-Programm Rc_zp_pb (Listing 17). Zum Verständnis wird Gl. (85) ausmultipliziert

$$Z_{GA}(p) = \frac{R_0 \prod_{i=1}^{m}(1+p\,\tau_i) + \sum_{i=1}^{m} R_i \prod_{\substack{j=1\\ j\neq i}}^{m}(1+p\,\tau_j)}{\prod_{i=1}^{m}(1+p\,\tau_i)} \quad (86)$$

In den Zeilen 480 bis 530 erfolgt die Berechnung des Nennerpolynoms von Gl. (86) mit Hilfe von Gl. (6), hier speziell mit m = const. = 1, $b_0 = 1$ und $b_1 = \tau_i$, wobei insgesamt n-mal multipliziert wird. Die Berechnung des Zählerpolynoms erfolgt in den Zeilen 550 bis 680 konsequent nach Gl. (86).

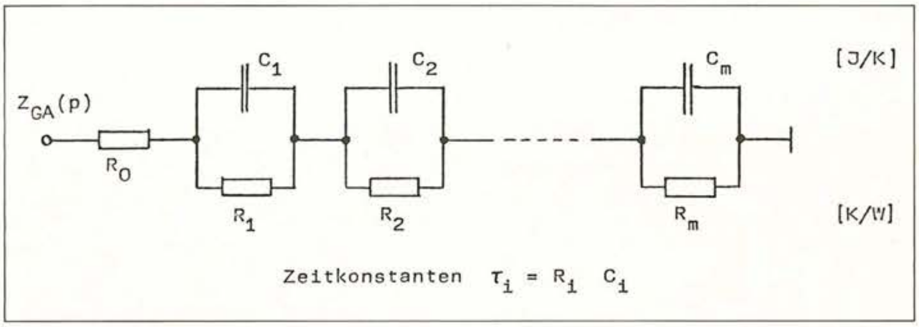

Bild 4: Elektrisches Ersatzschaltbild für die Ausgleichskurve eines gemessenen thermischen Verhaltens.

In das Programm wurde eine Fallunterscheidung eingearbeitet, die vorab prüft, ob R_0 vorhanden ist (Ch$ = „C") oder nicht (Ch$ = „H"), da dies entscheidend für den Grad des Zählerpolynoms und damit auch für die Rücktransformation ist. Auch wird vorab geprüft, ob die Ersatzschaltung nach Bild 4 kanonisch ist, d.h. ob nicht zufällig zwei (oder mehr) Zeitkonstanten gleich groß sind. Ohne diese Prüfung könnten sich für $z_{GA}(p)$ und $n_{GA}(p)$ gleiche Wurzeln ergeben, was entsprechend den Ausführungen von Abschnitt 6 nach Möglichkeit vermieden werden sollte.

9.2 Transformation aus dem Bild- in den Zeitbereich in einfachen Fällen

Gegeben sei eine gebrochen rationale Funktion

$$F(p) = \frac{Z(p)}{N(p)}, \; \text{gr}(Z) < \text{gr}(N) = n \quad (87)$$

im Bildbereich der LAPLACE-Transformation, die in den Zeitbereich transformiert werden soll. Allgemein soll der Grad des Zählerpolynoms kleiner als der des Nennerpolynoms sein, sonst ließe sich von F(p) ein ganzrationaler Anteil abspalten, der in den Zeitbereich transformiert, den DIRAC-Impuls und seine Ableitungen ergeben würde. Diese Fälle sollen hier ausgeschlossen sein, können aber mit dem Divisionsprogramm nach Listing 8 gelöst werden (siehe Abschnitt 9.5).

Zunächst sei außerdem vorausgesetzt, daß das Nennerpolynom nur einfache reelle Nullstellen besitze, die mit p_i bezeichnet werden sollen. Bekanntlich läßt sich dann die Gl. (87) in Partialbrüche

$$F(p) = \frac{Z(p)}{N(p)} = \frac{R_1}{p-p_1} + \frac{R_2}{p-p_2} + \ldots + \frac{R_n}{p-p_n} = \sum_{i=1}^{n} \frac{R_i}{p-p_i} \quad (88)$$

zerlegen. Zur Berechnung der Konstanten R_i multipliziert man die Gl. (88) mit $(p-p_i)$ und erhält den Ausdruck

$$(p-p_i)\,F(p) = R_i + (p-p_i)\sum_{\substack{j=1\\ j\neq i}}^{n} \frac{R_j}{p-p_j} \;. \quad (89)$$

Bestimmt man jetzt den Wert dieses Ausdrucks für $p=p_i$, so wird der rechte Term null und es bleibt der unbestimmte Ausdruck

$$R_i = (p-p_i)\,F(p) = (p-p_i)\frac{Z(p)}{N(p)} \;. \quad (90)$$

Mit Hilfe der Regel von l'HOSPITAL läßt sich

$$\lim_{p\to p_i} (p-p_i)\frac{Z(p)}{N(p)} = \frac{\frac{d}{dp}\left[(p-p_i)\,Z(p)\right]}{\frac{d}{dp}N(p)} = \frac{Z(p_i)}{N'(p_i)} \quad (91)$$

ableiten, womit sich für die Koeffizienten R_i die Gleichung

$$R_i = \frac{Z(p_i)}{N'(p_i)} \quad (92)$$

aufschreiben läßt. Mit Hilfe der Korrespondenzbeziehungen nach den Gln. (78 – 80) folgt daraus der „Entwicklungssatz von HEAVISIDE"

$$F(p) = \frac{Z(p)}{N(p)} \bullet\!\!-\!\!\circ\; f(t) = \sum_{i=1}^{n} \frac{Z(p_i)}{N'(p_i)}\, e^{p_i t}, \; n = \text{gr}(N) \;. \quad (93)$$

```
10    SUB Rc_zp_pb(INTEGER N,N_stern,Kan,Ch$,REAL Rp(*),Taup(*),Z_ga(*),N_ga(*))
20    !
30    ! SUB-Programm Rc_zp_pb zur Berechnung der Impedanzfunktion
40    ! eines Zweipoles aus RC-Gliedern in Partialbruchschaltung
50    !
60    ! Eingabedaten:
70    !       n                ... Anzahl der RC-Glieder
80    !       Ch$              ... "C" für Modell "case-ambient"     [mit  Rp(0)]
90    !                            "H" für Modell "heatsink-ambient" [ohne Rp(0)]
100   !       Rp(*)            ... Widerstände der Partialbruchschaltung
110   !                            (0:n) für Ch$="C" bzw. (1:n) für Ch$="H"
120   !       taup(1:n)        ... Zeitkonstanten der Partialbruchschaltung
130   !
140   ! Ergebnis:
150   !       n_stern          ... charakteristischer Wert der Polynomgrößen der
160   !                            Impedanz: n-1 für Ch$="H" bzw. n für Ch$="C"
170   !       kan              ... 1 für eine kanonische Partialbruchschaltung,
180   !                            sonst 0 (und ohne Lösung)
190   !       Z_ga(0:n_stern)  ... Zählerpolynom des Impedanz
200   !       N_ga(0:n)        ... Nennerpolynom der Impedanz
210   !
220   ! (c) 1985 by Jürgen Schwarz          Sprache: HP-BASIC     Datum: 25.04.85
230   ! Speichermedium: Kassetten 57/58     File-Name: RC_ZPP     Version:    1.3
240   !
250     INTEGER I,J,K
260     REAL Taup_stern(1:N+1),Z_ga_stern(0:N)
270     !
280     IF (Ch$="c") OR (Ch$="C") THEN
290       REDIM Rp(0:N)
300       N_stern=N
310       IF N>0 THEN REDIM Taup(1:N)                 ! Für n=0 kein tau vorhanden
320     ELSE
330       REDIM Rp(1:N),Taup(1:N)
340       N_stern=N-1
350     END IF
360     MAT Z_ga=ZER
370     MAT N_ga=ZER
380     REDIM Z_ga(0:N_stern),N_ga(0:N)
390     !
400     MAT Taup_stern=Taup
410     MAT SORT Taup_stern
420     Kan=1
430     FOR I=2 TO N
440       Kan=Kan AND (Taup_stern(I-1)<>Taup_stern(I))
450     NEXT I
460     IF NOT Kan THEN SUBEXIT  ! die Partialbruchschaltung ist nicht kanonisch
470     !
480     N_ga(0)=1
490     FOR I=1 TO N
500       FOR J=I TO 1 STEP -1
510         N_ga(J)=N_ga(J)+N_ga(J-1)*Taup(I)
520       NEXT J                      ! Parameter des Nennerpolynoms der Impedanz
530     NEXT I
540     !
550     IF (Ch$="c") OR (Ch$="C") THEN MAT Z_ga=(Rp(0))*N_ga
560     MAT Taup_stern=Taup
570     FOR I=N TO 1 STEP -1
580       IF I<N THEN Taup_stern(I)=Taup(I+1)
590       MAT Z_ga_stern=ZER(0:N_stern)
600       Z_ga_stern(0)=1
610       FOR J=1 TO N-1
620         FOR K=J TO 1 STEP -1
630           Z_ga_stern(K)=Z_ga_stern(K)+Z_ga_stern(K-1)*Taup_stern(J)
640         NEXT K
650       NEXT J
660       MAT Z_ga_stern=(Rp(I))*Z_ga_stern
670       MAT Z_ga=Z_ga+Z_ga_stern
680     NEXT I                        ! Parameter des Zählerpolynoms der Impedanz
690   SUBEND
```

Listing 17: SUB-Programm zur Transformation gemessener Verläufe in den Bildbereich der Laplace-Transformation.

Bei der Durchführung der Transformation in den Bildbereich mit Hilfe der Gl. (85) wurde der Anteil $1/p$ aus formalen Gründen weggelassen. Er muß natürlich bei der Rücktransformation wieder einfließen. Liegt also eine Funktion

$$F(p) = \frac{Z(p)}{p\,N(p)} \qquad (94)$$

vor, die entsprechend zu behandeln ist, so ergibt Gl. (93) mit $N^*(p) = p\,N(p)$ und

$$\frac{d}{dp} N^*(p) = \frac{d}{dp} p\,N(p) = N(p) + p\,N'(p) \qquad (95)$$

sowie Abtrennen der Nullstelle $p_0 = 0$ die Lösung

$$F(p) = \frac{Z(p)}{p\,N(p)} \;\bullet\!\!-\!\!\circ\; f(t) = \frac{Z(0)}{N(0)} + \sum_{i=1}^{n} \frac{Z(p_i)}{p_i\,N'(p_i)}\, e^{p_i t}, \quad n = \mathrm{gr}(N)\,. \qquad (96)$$

Ist hier $\mathrm{gr}(Z) < \mathrm{gr}(N)$ so gilt wegen des Grenzwerttheorems [15]

$$\lim_{t \to 0} f(t) = \lim_{p \to \infty} p\,F(p) = 0 \qquad (97)$$

und damit

$$\frac{Z(0)}{N(0)} + \sum_{i=1}^{n} \frac{Z(p_i)}{p_i\,N'(p_i)} = 0\,. \qquad (98)$$

Unter dieser Bedingung läßt sich die Korrespondenz (96) auch in der Form

$$F(p) = \frac{Z(p)}{p\,N(p)} \;\bullet\!\!-\!\!\circ\; f(t) = \sum_{i=1}^{n} \frac{-Z(p_i)}{p_i\,N'(p_i)} \left(1 - e^{-t/\tau_i}\right), \qquad (99)$$

$$\mathrm{gr}(Z) < \mathrm{gr}(N) = n, \quad \tau_i = -\frac{1}{p_i}$$

aufschreiben.

Dieser Rücktransformationsalgorithmus ist im Listing 18 implementiert, weil er in vielen Fällen die Lösung liefert. Selbstverständlich kann man auf praktisch analoge Weise die Korrespondenzbeziehungen (93) und (96) implementieren. Die Berechnung der Nullstellen erfolgt zweckmäßig mit dem SUB-Programm Newton_mod analog zu Listing 5 [8]. In den Zeilen 400

```
10   SUB Rc_t0_pb(INTEGER N,REAL Z(*),N(*),N_0(*),R_p(*),Tau_p(*))
20   !
30   ! SUB-Programm Rc_t0_pb zur Durchführung einer Transformation
40   ! aus dem Bildbereich der Laplace-Transformation in den Zeitbereich
50   ! bei gegebenen Zähler- und Nennerpolynomen der zu berechnenden Größe
60   ! und den bereits vorab berechneten Nullstellen des Nenners und
70   ! Anregung durch einen Einheitssprung (1/p)
80   !
90   ! Anwendung des Entwicklungssatzes von Heaviside bei einfachen, rellen
100  ! Nullstellen auf der negativen reellen Achse
110  !
120  ! Eingabedaten:   n           ... Grad des Nennerpolynoms
130  !                 Z(0:*)      ... Zählerpolynom im Bildbereich (Max.: 0:n-1)
140  !                 N(0:n)      ... Nennerpolynom im Bildbereich
150  !                 N_0(1:n)    ... Nullstellen des Nennerpolynoms
160  ! Ergebnis:       R_p(1:n)    ... Parameter der Lösung im Zeitbereich
170  !                 tau_p(1:n)  ... Parameter der Lösung im Zeitbereich
180  !
190  !                                n
200  ! Lösung:         f(t) = Summe R_p * (1 - exp(-t/tau_p))
210  !                               i=1
220  !
230  ! (c) 1985 by Jürgen Schwarz           Sprache: HP-BASIC      Datum: 25.04.85
240  ! Speichermedium: Kassetten 57/58      File-Name: RC_th0      Version:    1.3
250  !
260    INTEGER I,J
270    REAL N_0,R_zaehler,R_nenner,Z_stern(0:N-1)
280    !
290    MAT Z_stern=Z
300    MAT R_p=ZER
310    MAT Tau_p=ZER
320    REDIM Z_stern(0:N-1),N(0:N),N_0(1:N),R_p(1:N),Tau_p(1:N)
330    !
340    ! Zerlegung mit dem "Entwicklungsatz von Heaviside"
350    !
360    FOR I=1 TO N
370      N_0=N_0(I)
380      Tau_p(I)=-1/N_0
390      R_nenner=R_zaehler=0
400      FOR J=N-1 TO 0 STEP -1
410        R_zaehler=Z_stern(J)+N_0*R_zaehler
420        R_nenner=(J+1)*N(J+1)+N_0*R_nenner
430      NEXT J
440      R_p(I)=-R_zaehler/(N_0*R_nenner)
450    NEXT I                          ! Bewertung der jeweiligen Zeitkonstante
460  SUBEND
```

Listing 18: Einfaches Rücktransformations-SUB-Programm für Nennerpolynome mit einfachen reellen Nullstellen.

bis 430 erfolgt die Funktionsberechnung des Zählerpolynoms und die der Ableitung des Nennerpolynoms $N'(p_i)$ mit dem einfachen HORNER'schen Schema. Sonst zeigt das SUB-Programm keine Besonderheiten.

9.3 Rücktransformation beim Auftreten konjugiert komplexer Nullstellen des Nennerpolynoms

Die Korrespondenzbeziehung (96) gilt selbstverständlich auch beim Auftreten komplexer, einfacher Nullstellen des Nennerpolynoms. Die Nullstellen treten paarweise konjugiert

$$\begin{aligned} p_i &= \sigma + \mathrm{i}\,\omega \\ p_{i+1} &= \sigma - \mathrm{i}\,\omega \end{aligned} \tag{100}$$

auf und mit Hilfe der EULER'schen Formel [12]

$$e^{\mathrm{i}\,\omega t} = \cos\omega t + \mathrm{i}\sin\omega t \tag{101}$$

erhält man

$$\frac{Z(p_i)}{p_i\,N'(p_i)}\,e^{p_i t} + \frac{Z(p_{i+1})}{p_{i+1}\,N'(p_{i+1})}\,e^{p_{i+1} t} = e^{\sigma t}(A\cos\omega t + B\sin\omega t) \tag{102}$$

mit

$$\begin{array}{l|l} Z(\sigma + \mathrm{i}\,\omega) = r_Z + \mathrm{i}\,i_Z & Z(\sigma - \mathrm{i}\,\omega) = r_Z - \mathrm{i}\,i_Z \\ (\sigma + \mathrm{i}\,\omega)\,N'(\sigma + \mathrm{i}\,\omega) = r_N + \mathrm{i}\,i_N & (\sigma - \mathrm{i}\,\omega)\,N'(\sigma - \mathrm{i}\,\omega) = r_N - \mathrm{i}\,i_N \end{array} \tag{103}$$

und daraus unter Verwendung der Gl. (74)

$$\begin{aligned} A &= 2\,\frac{r_Z\,r_N + i_Z\,i_N}{r_N^2 + i_N^2} \\ B &= 2\,\frac{r_Z\,i_N - i_Z\,r_N}{r_N^2 + i_N^2}\,. \end{aligned} \tag{104}$$

Durch die Tatsache, daß die zwei Funktionswerte eines Polynoms mit zwei konjugiert komplexen Argumenten ebenfalls konjugiert komplex sind (Gl. (103)), was man durch Betrachten der Gl. (69) sofort einsieht, ergibt sich, daß die Lösung (102) reell ist. Diese läßt sich leicht in die gewünschte Form

$$R_i\,e^{-t/\tau_i}\cos(\omega_i + \varphi_i) \tag{105}$$

bringen, wobei hier

$$R_i = \sqrt{A^2 + B^2} \tag{106}$$

und

$$\varphi_i = -\arctan\frac{B}{A} \tag{107}$$

sind. Bei Gl. (107) sind zusätzlich die Beziehungen zu den einzelnen Quadranten zu beachten.

Listing 19 zeigt ein entsprechendes SUB-Programm. Die z. B. mit SILJAK (Listing 6) berechneten Nullstellen des Nennerpolynoms werden bei Aufruf des Programms mit angefordert. Diese Verfahrensweise ist zweckmäßig, wenn Berechnungen mit mehreren Zählerpolynomen und gleichem Nennerpolynom durchgeführt werden sollen (siehe Listing 22). Sonst kann der Aufruf von SILJAK auch innerhalb von Pb__he erfolgen.

Da SILJAK die Lösung in unregelmäßiger Folge liefert, werden diese in den Zeilen 600 bis 720 erst nach der Größe ihres Realteiles sortiert. In den Zeilen 740 bis 930 wird geprüft, ob die berechneten komplexen Wurzeln auch konjugiert auftreten. Die Wurzeln mit negativem Imaginärteil werden „gelöscht". Anschließend werden die Koeffizienten des Nennerpolynoms $p\,N'(p)$ berechnet. Dann erfolgt die Abarbeitung nach dem oben abgeleiteten Algorithmus. Zur Berechnung der Funktionswerte der Polynome werden die Programme FNHorner (Listing 11) und Komplex__horner (Listing 15) eingesetzt.

Gestattet der einzusetzende Rechner den Einsatz von Funktionsunterprogrammen und SUB-Programmen nicht, so kann man das in Listing 20 ausschnittsweise gezeigte äquivalente Programm ohne SUB-Programmaufruf einsetzen. Bei einer reellen Nullstelle arbeitet es wie das in Listing 18 vorgestellte Programm und bei einer rein imaginären Nullstelle wird die Tatsache berücksichtigt, daß die einzelnen Terme der Polynome rein reelle oder rein imaginäre Glieder liefern, je nachdem, ob die Potenz des Arguments gerade oder ungerade ist:

$\mathrm{i}^0 \Rightarrow$ reellen Teil
$\mathrm{i}^1 \Rightarrow$ imaginären Teil
$\mathrm{i}^2 \Rightarrow$ negativen reellen Teil
$\mathrm{i}^3 \Rightarrow$ negativen imaginären Teil
$\mathrm{i}^4 \Rightarrow$ reellen Teil
usw.

Bei einer konjugiert komplexen Nullstelle wird zur Berechnung das im Abschnitt 8.4 vorgestellte Verfahren verwendet. Die Wertzuweisung an die Matrizen erfolgt hier unter Verwendung von zwei Matrizen Real(∗) und Imag(∗), die jeweils die reelle Einheit 1 und die imaginäre Einheit i repräsentieren. Die Addition der mit den Real- und Imaginärteilen bewerteten Matrizen X(∗) und Y(∗) in den Zeilen 1440 bis 1460 führt schließlich zu der Repräsentation der komplexen Nullstelle. Die Division des Wertes des Zählerpolynoms durch den Wert des Nennerpolynoms wird hier durch Inversion nach Gl. (76) vorgenommen. Nach Ausgabe der Anteile der Kosinus- und der Sinusanteile erfolgt die Abarbeitung völlig analog zu Listing 19.

9.4 Rücktransformation bei mehrfachen (reellen und komplexen) Nullstellen

Beim Auftreten von mehrfachen Nennernullstellen in Gl. (87) versagen die bisher abgeleiteten Gleichungen, da dann die Ableitung des Nennerpolynoms an der Nullstelle ebenfalls null ist. Das Nennerpolynom kann nach Gl. (10) in der Form

$$\begin{aligned} N(p) &= \\ &(p-p_1)^{v_1}\,(p-p_2)^{v_2}\ldots(p-p_n)^{v_n} \end{aligned} \tag{108}$$

dargestellt werden. Die Partialbruchzerlegung von Gl. (87) hat hier folgende Form:

$$\begin{aligned} \frac{Z(p)}{N(p)} &= \\ &\frac{C_{11}}{(p-p_1)} + \frac{C_{12}}{(p-p_1)^2} + \ldots + \frac{C_{1v_1}}{(p-p_1)^{v_1}} + \ldots \\ &\ldots + \frac{C_{n1}}{(p-p_n)} + \ldots + \frac{C_{nv_n}}{(p-p_n)^{v_n}} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{v_i}\frac{C_{ij}}{(p-p_i)^j}\,. \end{aligned} \tag{109}$$

Zur Berechnung der unbekannten Konstanten C_{ij} multipliziert man die Gleichung mit $(p-p_i)^{v_i}$ und untersucht die gewonnene Beziehung an der Stelle $p = p_i$. Analog zu der Ableitung nach Gl. (90) bleibt nur der Term C_{iv_i} auf der rechten Seite stehen und eine entsprechende Grenzwertbetrachtung liefert schließlich

Listing 19: Rücktransformationsprogramm mit Berücksichtigung aller einfachen Nullstellen (reell und komplex).

```
10    SUB Pb_he(Z(*),N(*),R_0(*),I_0(*),R(*),Tau(*),Om(*),Phi(*),Ep,INTEGER N,M)
20    !
30    ! SUB-Programm Pb_he zur Durchführung der Laplace-Rücktransformation
40    ! aus dem Bildbereich in den Originalbereich
50    !
60    ! Gegeben: gebrochen rationale Bildfunktion F(p) = Z(p)/N(p),
70    !          die mit einem Sprung (1/p) angeregt wird
80    !
90    ! Verfahren:
100   ! Partialbruchzerlegung und Anwendung des Entwicklungssatzes von Heaviside
110   ! bei bereits berechneten, einfachen Nullstellen des Nenners
120   !
130   ! Eingangsgrößen:
140   !            n          ... Grad des Nennerpolynoms (ohne Sprung)
150   !            Z(0:*)     ... Koeffizienten des Zählerpolynoms (Max.: 0:n)
160   !            N(0:n)     ... Koeffizienten des Nennerpolynoms (ohne Sprung)
170   !            R_0(1:n)   ... Realteile      der Nullstellen des Nennerpolynoms
180   !            I_0(1:n)   ... Imaginärteile der Nullstellen des Nennerpolynoms
190   !            ep         ... Größe für die Genauigkeit der Nullstellen bei
200   !                           der Prüfung auf konjugiert komplexe Nullstellen
210   !
220   ! Form des Ergebnisses:
230   !
240   ! f(t) =  R_0 + Summe [ R_i * exp(-t/tau_i) ] + ...
250   !         ... + Summe [ R_i * exp(-t/tau_i) *  cos(om_i*t + phi_i) ]
260   !
270   ! Ausgangsgrößen:
280   !            n          ... Grad des Nennerpolynoms (ohne Sprung)
290   !            m          ... Anzahl der Elemente im Zeitbereich (m ≤ n)
300   !            R   (0:m)  ... Proportionalanteile              der Lösung
310   !            tau(1:m)   ... Zeit- bzw. Dämpfungskonstanten der Lösung
320   !            om  (1:m)  ... Kreisfrequenzen                  der Lösung
330   !            phi(1:m)   ... Phasenlagen                      der Lösung
340   !
350   ! (c) 1985 by Jürgen Schwarz         Sprache: HP-BASIC     Datum: 01.06.85
360   ! Speichermedium: Kassetten 73/74    File-Name: Heavis     Version:    1.1
370   !
380     INTEGER Boo_r,Boo_i,Dround,I,J
390     REAL A_cos,A_sin,R_0,I_0,R_zaehler,R_nenner,I_zaehler,I_nenner
400     REAL Z_st(0:N),N_st(0:N),R_0_st(1:N),I_0_st(1:N)
410     !
420     IF (N<=0) OR (Ep<=0) THEN PAUSE          ! widersprüchliche Eingabedaten
430     MAT Z_st=Z
440     REDIM Z_st(0:N),N(0:N),R_0(1:N),I_0(1:N)
450     MAT R=ZER
460     MAT Tau=ZER
470     MAT Om=ZER
480     MAT Phi=ZER
490     REDIM R(0:N),Tau(1:N),Om(1:N),Phi(1:N)
500     !
510     M=N
520     R(0)=Z_st(0)/N(0)
530     IF N=1 THEN
540       R(1)=Z_st(1)/N(1)-Z_st(0)/N(0)
550       Tau(1)=N(1)/N(0)
560       Om(1)=Phi(1)=0                 ! Bewertung so, daß cos(0*t+phi)=1 ist
570       SUBEXIT
580     END IF
590     !
600     MAT R_0_st=R_0
610     MAT I_0_st=ZER
620     MAT SORT R_0_st              ! Sortierung nach aufsteigenden Zeitkonstanten
630     MAT Tau=R_0
640     FOR I=1 TO N
650       J=0
660       REPEAT
670         J=J+1
680       UNTIL R_0_st(I)=Tau(J)
690       Tau(J)=9.99999999999E99
700       IF ABS(I_0(J))>=Ep*ABS(R_0_st(I)) THEN I_0_st(I)=I_0(J)
710       IF ABS(R_0_st(I))<Ep*ABS(I_0_st(I)) THEN R_0_st(I)=0
720     NEXT I
730     !
```

Listing 19, Fortsetzung

```
740     Dround=INT(LGT(Ep))
750     I=0
760     REPEAT
770       I=I+1
780       IF I_0_st(I)<>0 THEN
790         IF I+1>M THEN Un_konj
800         Boo_r=(DROUND(R_0_st(I),Dround)=DROUND(R_0_st(I+1),Dround))
810         Boo_i=(DROUND(I_0_st(I),Dround)=-DROUND(I_0_st(I+1),Dround))
820         IF NOT (Boo_r AND Boo_i) THEN Un_konj
830         M=M-1
840         R_0_st(I)=.5*(R_0_st(I)+R_0_st(I+1))
850         I_0_st(I)=ABS(.5*(I_0_st(I)-I_0_st(I+1)))
860         FOR J=I+1 TO M
870           R_0_st(J)=R_0_st(J+1)
880           I_0_st(J)=I_0_st(J+1)
890         NEXT J
900         REDIM R_0_st(1:M),I_0_st(1:M)
910       END IF
920     UNTIL I>=M
930     REDIM R(0:M),Tau(1:M),Om(1:M),Phi(1:M)
940     !
950     N_st(0)=0
960     FOR I=1 TO N
970       N_st(I)=I*N(I)                          ! Berechnung von p * d N(p)/d p
980     NEXT I
990     FOR I=1 TO M
1000      R_0=R_0_st(I)                     ! Realteil der Nullstelle
1010      Om(I)=I_0=I_0_st(I)               ! Imaginärteil der Nullstelle
1020      IF I_0=0 THEN                     ! rein reelle Nullstelle
1030        R(I)=FNHorner(N,R_0,Z_st(*))/FNHorner(N,R_0,N_st(*))
1040        Tau(I)=-1/R_0
1050        Om(I)=Phi(I)=0                    ! Bewertung so, daß cos(0*t+phi)=1 ist
1060      ELSE                              ! konjugierte komplexe Nullstelle
1070        Om(I)=I_0
1080        IF R_0=0 THEN                   ! rein imaginäre Nullstelle
1090          Tau(I)=9.99999999999E99       ! tau strebt gegen unendlich
1100        ELSE                            ! konjugiert komplexe Nullstelle
1110          Tau(I)=-1/R_0
1120        END IF
1130        CALL Komplex_horner(N,R_0,I_0,Z_st(*),R_zaehler,I_zaehler)
1140        CALL Komplex_horner(N,R_0,I_0,N_st(*),R_nenner,I_nenner)
1150        A_cos=2*(R_zaehler*R_nenner+I_zaehler*I_nenner)
1160        A_sin=2*(R_zaehler*I_nenner-I_zaehler*R_nenner)
1170        IF A_cos THEN
1180          IF A_cos>=0 THEN
1190            Phi(I)=ATN(-A_sin/A_cos)
1200            IF Phi(I)<0 THEN Phi(I)=Phi(I)+2*PI
1210          ELSE
1220            Phi(I)=ATN(-A_sin/A_cos)+PI
1230          END IF
1240        ELSE
1250          Phi(I)=PI*(1+.5*SGN(A_sin))
1260        END IF
1270        R(I)=SQR(A_cos*A_cos+A_sin*A_sin)
1280        R(I)=R(I)/(R_nenner*R_nenner+I_nenner*I_nenner)
1290      END IF
1300    NEXT I
1310    SUBEXIT
1320    !
1330 Un_konj: !
1340    PRINT "Fehler bei der Nullstellenbestimmung."
1350    PRINT "Es liegt eine unkonjugiert komplexe Nullstelle vor!"
1360    PAUSE
1370  SUBEND
```

$$C_{i\nu_i} = \left. \frac{(p-p_i)^{\nu_i} Z(p)}{N(p)} \right|_{p=p_i} = \nu_i! \, \frac{Z(p_i)}{N^{(\nu_i)}(p_i)} \qquad (110)$$

Zur Berechnung der anderen Konstanten wird Gl. (109) ebenfalls mit $(p-p_i)^{\nu_i}$ multipliziert und j-mal differenziert. Dann werden für $p = p_i$ alle Glieder der rechten Seite außer $j! \, C_{i,\nu_i - j}$ null. Es gilt also

$$C_{i,\nu_i - j} = \frac{1}{j!} \left. \frac{d^j}{dp^j} \frac{(p-p_i)^{\nu_i} Z(p)}{N(p)} \right|_{p=p_i} \qquad (111)$$

Eine Vereinfachung bringt die Einführung des mit der aktuellen Nullstelle gewonnenen Deflationspolynoms

$$N^*(p) = \frac{N(p)}{(p-p_i)^{v_i}} \tag{112}$$

in die obigen Gleichungen. Hier ergibt sich mit

$$C_{i,\,v_i-j} = \frac{1}{j\,!}\;\frac{d^j}{dp^j}\;\frac{Z(p_i)}{N^*(p_i)}\,,\quad j=0(1)v_i-1 \tag{113}$$

eine einheitliche Darstellung zur Berechnung der Konstanten C_{ij}.

Dabei ist zu berücksichtigen, daß $N^*(p)$ für komplexes p_i komplexe Koeffizienten hat. Jetzt muß noch eine geeignete Gleichung zur Berechnung der Ableitungen der rationalen Funktion bereitgestellt werden. Für die erste Ableitung gilt

$$\frac{d}{dp}\;\frac{Z(p)}{N^*(p)} = \frac{N^*(p)\,Z'(p) - Z(p)\,N^{*\prime}(p)}{N^{*2}(p)}\;. \tag{114}$$

Wird dieser Ausdruck ein zweites Mal nach p abgeleitet, dann erhält man mit

$$\frac{d^2}{dp^2}\;\frac{Z(p)}{N^*(p)} = \frac{N^{*2}\,Z'' - N^*(2\,Z'\,N^{*\prime} + Z\,N^{*\prime\prime}) + 2\,Z(N^{*\prime})^2}{N^{*3}(p)} \tag{115}$$

einen Ausdruck, dessen Nenner sich durch einfache Multiplikation von $N^*(p)$ aus dem Nenner der nächstniedrigeren Ableitung ergibt. Aus dieser Erkenntnis läßt sich die Rekursionsformel

$$\frac{d^j}{dp^j}\;\frac{Z(p)}{N^*(p)} = \frac{z_j(p)}{[N^*(p)]^{j+1}} \tag{116}$$

mit

$$z_{j+1}(p) = N^*(p)\,z_j'(p) - (j+1)\,N^{*\prime}(p)\,z_j(p) \tag{117}$$

und

$$z_0(p) = Z(p) \tag{118}$$

ableiten (Methode der vollständigen Induktion).

Abschließend werden mit

$$\frac{1}{p^n} \;\bullet\!\!-\!\!\circ\; \frac{1}{(n-1)\,!}\,t^{n-1} \tag{119}$$

$$\frac{1}{(p-p_i)^n} \;\bullet\!\!-\!\!\circ\; \frac{1}{(n-1)\,!}\,t^{n-1}\,e^{p_i t} \tag{120}$$

zwei Korrespondenzbeziehungen [15] notiert, mit deren Hilfe die einzelnen Partialbrüche nach Gl. (105) in den Zeitbereich transformiert werden können. Zusammenfassend gelten hier die Korrespondenzen

$$F(p) = \frac{Z(p)}{N(p)} \;\bullet\!\!-\!\!\circ\; f(t) = \sum_{i=1}^{n}\sum_{j=1}^{v_i} \frac{t^{v_i-j-1}\,e^{p_i t}}{j\,!\,(v_i-j-1)\,!}\;\frac{d^j}{dp^j}\;\frac{Z(p_i)}{N^*(p_i)} \tag{121}$$

$$F(p) = \frac{Z(p)}{p\,N(p)} \;\bullet\!\!-\!\!\circ\; f(t) = \frac{Z(0)}{N(0)} + \sum_{i=1}^{n}\sum_{j=1}^{v_i} \frac{t^{v_i-j-1}\,e^{p_i t}}{j\,!\,(v_i-j-1)\,!}\;\frac{d^j}{dp^j}\;\frac{Z(p_i)}{p_i\,N^*(p_i)}\;. \tag{122}$$

Listing 21 zeigt das SUB-Programm Hea, mit dessen Hilfe gebrochen rationale Bildfunktionen auch bei mehrfachen Nullstellen des Nennerpolynoms in den Zeitbereich transformiert werden können. Der Anfang ist analog zu dem Listing 19 aufgebaut. In den Zeilen 1140 bis 1420 wird das ursprüngliche Nennerpolynom $N(p)$ zu $N^*(p)$ abdividiert und anschließend wird es mit der „unterschlagenen" Nullstelle bei Null multipliziert und seine Ableitung bereitgestellt. In der folgenden LOOP-Schleife erfolgt zunächst die Berechnung der Koeffizienten nach Gl. (113) und danach die Berechnung der Ableitung mit Hilfe der Rekursionsformel nach den Gln. (116) und (117). Das in den Zeilen 1710 und 1880 aufgerufene Funktionsunterprogramm FNN_fak dient zur Berechnung von n !, ist in den Zeilen 2240 bis 2410 wiedergegeben und bedarf keiner weiteren Erklärung. Bei dem in den Zeilen 2030, 2040 und 2070 verwendeten SUB-Programm K_p handelt es sich um das in Listing 3 abgedruckte Programm Komplex_produkt, dessen Name gekürzt wurde, um mit 80 Zeichen pro Zeile auszukommen und die Variablenbezeichnungen möglichst anschaulich gestalten zu können.

9.5 Anwendungsbeispiel

Eine entscheidende Größe beim Einsatz von Halbleiterbauelementen ist die Temperatur der Sperrschicht. Leider ist der Aufwand zur Bestimmung der Sperrschichttemperatur sehr hoch [20], so daß solche Messungen im allgemeinen nur bei den Herstellern der Halbleiterbauelemente durchgeführt werden. Die Ergebnisse dieser Messungen werden den Anwendern dieser Bauelemente in Form von transienten Wärmewiderständen bekannt gemacht. Diese gelten aber nur in Zusammenhang mit einer bestimmten, bei den Messungen verwendeten Kühleinrichtung [21]. Sollen jetzt andere Kühler zum Einsatz kommen, dann kommt man an eigenen Messungen nicht vorbei. Zu diesem Zweck werden die Halbleiterbauelemente, wie beim späteren Einsatz vorgesehen, auf den Kühlern montiert. Am Übergang Halbleiterbauelement – Kühleroberfläche wird ein Temperaturfühler (i. allg. ein Thermoelement) befestigt und es wird der Verlauf der Temperatur an diesem Punkt bei einem Sprung der Verlustleistung in der Sperrschicht gemessen. Aus diesem Temperaturverlauf soll nun der Verlauf der Sperrschichttemperatur und der Verlauf der Leistung über den Übergang berechnet werden.

Das thermische Verhalten der Halbleiterbauelemente kann mit der Gleichung

$$\begin{pmatrix} T_J(p) \\ P_G(p) \end{pmatrix} = \begin{pmatrix} H_{11}(p) & H_{12}(p) \\ H_{21}(p) & H_{22}(p) \end{pmatrix} \cdot \begin{pmatrix} P_J(p) \\ T_G(p) \end{pmatrix} \tag{123}$$

vollständig beschrieben werden [22]. Dabei ist $T_J(p)$ die Sperrschichttemperatur, $P_G(p)$ die über den Übergang Halbleiterbauelement—Kühler fließende Leistung, $P_J(p)$ der Verlustleistungsverlauf in der Sperrschicht und $T_G(p)$ der am Übergang gemessene Temperaturverlauf, jeweils dargestellt im Bildbereich der LAPLACE-Transformation. Die Matrix mit den Hybridparametern charakterisiert die thermischen Eigenschaften des Halbleiterbauelementes.

Alle diese Parameter sind gebrochene rationale Funktionen. Die Hybridmatrix hat folgenden Aufbau

$$\|H\| = \frac{1}{\ell(p)} \begin{pmatrix} h_{11}(p) & 1 \\ 1 & h_{22}(p) \end{pmatrix}, \tag{124}$$

kann also mit drei charakteristischen Polynomen $h_{11}(p)$, $h_{22}(p)$ und $\ell(p)$ beschrieben werden. Wird der zur Messung erzeugte Leistungssprung

$$P_J(p) = \frac{P_J}{p} \tag{125}$$

```
10     SUB Hea_e(Z(*),N(*),R_0(*),I_0(*),R(*),Tau(*),Om(*),Phi(*),Ep,INTEGER N,M)

360    ! (c) 1984 by Jürgen Schwarz          Sprache: HP-BASIC      Datum: 08.06.84
370    ! Speichermedium: Kassetten 73/74     File-Name: Heavis      Version:    1.0

430      REAL Zaehler(1:2,1:2),Nenner(1:2,1:2),Real(1:2,1:2),Imag(1:2,1:2)
440      REAL Null_st(1:2,1:2),X(1:2,1:2),Y(1:2,1:2)

990      DATA 1,0,0,1,    0,-1,1,0     ! Repräsentation der rellen und der
1000     MAT READ Real,Imag            ! imaginären Einheit (1 bzw. i)
1010     FOR I=1 TO M
1020       R_0=R_0_st(I)               ! Realteil der Nullstelle
1030       I_0=I_0_st(I)               ! Imaginärteil der Nullstelle
1040       IF I_0=0 THEN               ! rein reelle Nullstelle
1050         Tau(I)=-1/R_0
1060         Om(I)=Phi(I)=0                 ! Bewertung so, daß cos(0*t+phi)=1 ist

1140       ELSE                           ! konjugierte komplexe Nullstelle
1150         Om(I)=I_0
1160         IF NOT R_0 THEN              ! rein imaginäre Nullstelle
1170           Tau(I)=9.99999999999E99    ! tau strebt gegen unendlich
1180           R_zaehler=I_zaehler=R_nenner=I_nenner=0
1190           FOR J=N TO 0 STEP -1
1200             SELECT J MOD 4
1210             CASE 0                              ! J = 0, 4, 8, 12, ...
1220               R_zaehler=Z_st(J)+R_zaehler*I_0*I_0
1230               IF J<=N-1 THEN I_nenner=(J+1)*N(J+1)+I_nenner*I_0*I_0
1240             CASE 1                              ! J = 1, 5, 9, 13, ...
1250               I_zaehler=Z_st(J)+I_zaehler*I_0*I_0
1260               IF J<=N-1 THEN R_nenner=(J+1)*N(J+1)+R_nenner*I_0*I_0
1270             CASE 2                              ! J = 2, 6, 10, 14, ...
1280               R_zaehler=-Z_st(J)+R_zaehler*I_0*I_0
1290               IF J<=N-1 THEN I_nenner=-(J+1)*N(J+1)+I_nenner*I_0*I_0
1300             CASE 3                              ! J = 3, 7, 11, 15, ...
1310               I_zaehler=-Z_st(J)+I_zaehler*I_0*I_0
1320               IF J<=N-1 THEN R_nenner=-(J+1)*N(J+1)+R_nenner*I_0*I_0
1330             END SELECT
1340           NEXT J
1350           I_zaehler=I_zaehler*I_0
1360           R_nenner=-R_nenner*I_0*I_0
1370           I_nenner=I_nenner*I_0
1380           A_cos=2*(R_zaehler*R_nenner+I_zaehler*I_nenner)
1390           A_cos=A_cos/(R_nenner*R_nenner+I_nenner*I_nenner)
1400           A_sin=2*(R_zaehler*I_nenner-I_zaehler*R_nenner)
1410           A_sin=A_sin/(R_nenner*R_nenner+I_nenner*I_nenner)
1420         ELSE                      ! R_0 AND I_0: konjugiert komplexe Nullstelle
1430           Tau(I)=-1/R_0
1440           MAT X=(R_0)*Real        ! Realteil       der komplexen Nullstelle
1450           MAT Y=(I_0)*Imag        ! Imaginärteil   der komplexen Nullstelle
1460           MAT Null_st=X+Y         ! Repräsentation der komplexen Nullstelle
1470           !
1480           MAT Zaehler=(Z_st(N))*Real ! Repr. der reellen Zählerkoeffizienten
1490           MAT Nenner=ZER
1500           FOR J=N-1 TO 0 STEP -1
1510             MAT X=(Z_st(J))*Real      ! Repr. der reellen Zählerkoeffizienten
1520             MAT Y=Zaehler*Null_st
1530             MAT Zaehler=X+Y
1540             MAT X=((J+1)*N(J+1))*Real ! Repr. der Ableitung der Nennerkoeff.
1550             MAT Y=Nenner*Null_st
1560             MAT Nenner=X+Y
1570           NEXT J
1580           MAT Y=Nenner*Null_st     ! = N_0*Nenner
1590           MAT Nenner=INV(Y)        ! = 1/(N_0*Nenner)
1600           MAT Y=Zaehler*Nenner     ! = Zähler/(N_0*Nenner]
1610           A_cos=2*Y(1,1)           ! = 2*Re[Zähler/(N_0*Nenner)]
1620           A_sin=2*Y(1,2)           ! = -2*Im[Zähler/(N_0*Nenner)]
1630         END IF
1640         IF A_cos THEN
1650           IF A_cos>=0 THEN
```

Listing 20: Ausschnitt aus einem SUB-Programm mit denselben Aufgaben wie Listing 19 ohne SUB-Programmaufruf.

Listing 21: SUB-Programm zur Rücktransformation aus dem Bildbereich der Laplace-Transformation beim Auftreten auch mehrfacher Nullstellen des Nennerpolynoms.

```
10    SUB Hea(Z(*),N(*),R(*),I(*),C(*),Tau(*),Om(*),Phi(*),Ep,INTEGER N,M,V(*))
20    !
30    ! SUB-Programm Hea zur Durchführung der Laplace-Rücktransformation
40    ! aus dem Bildbereich in den Originalbereich
50    !
60    ! Gegeben: gebrochen rationale Bildfunktion F(p) = Z(p)/N(p),
70    !          die mit einem Sprung (1/p) angeregt wird
80    !
90    ! Verfahren:
100   ! Partialbruchzerlegung und Anwendung des Entwicklungssatzes von Heaviside
110   ! bei bereits berechneten Nullstellen des Nenners R(*) und I(*). Mehr-
120   ! fache Nullstellen werden in R(*) und I(*) nicht mehrfach wiedergegeben.
130   !
140   ! Eingangsgrößen:
150   !  n                  ... Anzahl der Nullstellen des Nennerpolynoms
160   !                         (ohne Sprung und mehrfache Nullstellen)
170   !  Z(0:*)             ... Koeffizienten des Zählerpolynoms (Max.: 0:SUM(v))
180   !  N(0:SUM(v))        ... Koeffizienten des Nennerpolynoms (ohne Sprung)
190   !  R(1:n)             ... Realteile     der Nullstellen des Nennerpolynoms
200   !  I(1:n)             ... Imaginärteile der Nullstellen des Nennerpolynoms
210   !  v(1:n)             ... Vielfachheit der Nullstellen des Nennerpolynoms
220   !  ep                 ... Größe für die Genauigkeit der Nullstellen bei
230   !                         der Prüfung auf konjugiert komplexe Nullstellen
240   !
250   ! Form des Ergebnisses:
260   !
270   !                   m    v_i-1
280   ! f(t) =  C_00 + Summe Summe [ C_ij * t^j * EXP(-t/tau_i) ] * ...
290   !                 i=1   j=0             ... * COS(om_i*t + phi_ij) ]
300   !
310   ! Ausgangsgrößen:
320   !  m                    ... Anzahl der Elemente im Zeitbereich (m ≤ n)
330   !  C(0:m,0:MAX(v)-1)    ... Proportionalanteile               der Lösung
340   !  tau(1:m)             ... Zeit- bzw. Dämpfungskonstanten der Lösung
350   !  om (1:m)             ... Kreisfrequenzen                   der Lösung
360   !  phi(1:m,0:MAX(v)-1) ... Phasenlagen                       der Lösung
370   !  v(1:m)               ... Anzahl der Elemente mit gleichem tau(i)
380   !
390   ! (c) 1985 by Jürgen Schwarz        Sprache: HP-BASIC     Datum: 01.06.85
400   ! Speichermedium: Kassetten 73/74   File-Name: Heavis     Version:    1.1
410   !
420     INTEGER Boo_m,Boo_r,Boo_i,Dround,D,I,J,K,V_0,N_poly,N_nenner,N_zaehler
430     INTEGER N_max,N_stern,V_st(1:N)
440     !
450     REAL C_re,C_im,R_0,I_0,R_zaehler,R_nenner,U,V,X,Y
460     REAL R_st(1:N),I_st(1:N)
470     !
480     REAL Z_r(0:200),Z_i(0:200),Z_r_strich(0:199),Z_i_strich(0:199)
490     REAL N_r(0:200),N_i(0:200),P_r(0:200),P_i(0:200)       ! Feld: (0:n_max)
500     REAL M_r(0:SUM(V)),M_i(0:SUM(V))
510     REAL M_r_strich(0:SUM(V)-1),M_i_strich(0:SUM(V)-1)
520     !
530     IF (N<=0) OR (SUM(V)<N) OR (Ep<=0) THEN PAUSE   ! widersprüchliche Daten
540     M=N
550     N_poly=SUM(V)                                   ! Grad des Nennerpolynoms
560     MAT C=ZER
570     MAT Tau=ZER
580     MAT Om=ZER
590     MAT Phi=ZER
600     REDIM Z(0:ROW(Z)-1),N(0:N_poly),R(1:N),I(1:N),V(1:N)
610     IF SUM(V)<>N_poly THEN PAUSE                    ! widersprüchliche Daten
620     !
630     MAT R_st=R
640     MAT SORT R_st
650     MAT Tau=R
660     MAT I_st=ZER
670     FOR I=1 TO N
680       J=0
690       REPEAT
700         J=J+1
710       UNTIL R_st(I)=Tau(J)
720       Tau(J)=9.99999999999E99
730       V_st(I)=V(J)
740       IF ABS(I(J))>=Ep*ABS(R_st(I)) THEN I_st(I)=I(J)
750       IF ABS(R_st(I))<Ep*ABS(I_st(I)) THEN R_st(I)=0
```

Listing 21, Fortsetzung

```
760     NEXT I
770     !
780     I=D=0
790     Dround=INT(LGT(Ep))
800     REPEAT
810       I=I+1
820       D=MAX(D,V_st(I))               ! maximale Vielfachheit einer Nullstelle
830       IF I_st(I)<>0 THEN
840         IF I+1>M THEN Un_konj
850         Boo_m=(V_st(I)=V_st(I+1))
860         Boo_r=(DROUND(R_st(I),Dround)=DROUND(R_st(I+1),Dround))
870         Boo_i=(DROUND(I_st(I),Dround)=-DROUND(I_st(I+1),Dround))
880         IF NOT (Boo_m AND Boo_r AND Boo_i) THEN Un_konj
890         M=M-1
900         R_st(I)=.5*(R_st(I)+R_st(I+1))
910         I_st(I)=ABS(.5*(I_st(I)-I_st(I+1)))
920         FOR J=I+1 TO M
930           V_st(J)=V_st(J+1)
940           R_st(J)=R_st(J+1)
950           I_st(J)=I_st(J+1)
960         NEXT J
970         REDIM V_st(1:M),R_st(1:M),I_st(1:M)
980       END IF
990     UNTIL I>=M
1000    REDIM C(0:M,0:D-1),Tau(1:M),Om(1:M),Phi(1:M,0:D-1)
1010    N_max=MAX(D*(N_poly-D+1),N_poly+(D-1)*(N_poly-D))      ! Grad des größten
1020    !                            auftretenden Polynoms (muß kleiner als 201 sein)
1030    C(0,0)=Z(0)/N(0)
1040    FOR I=1 TO M
1050      V_0=V_st(I)                               ! Vielfachheit der Nullstelle
1060      R_0=R_st(I)                               ! Realteil      der Nullstelle
1070      I_0=Om(I)=I_st(I)                         ! Imaginärteil der Nullstelle
1080      IF R_0=0 THEN                             ! rein imaginäre   Nullstelle
1090        Tau(I)=9.99999999999E99                 !  tau strebt gegen unendlich
1100      ELSE
1110        Tau(I)=-1/R_0
1120      END IF
1130      !
1140      MAT N_r=ZER(0:N_max)
1150      MAT N_i=ZER(0:N_max)
1160      MAT M_r=N
1170      MAT M_i=ZER(0:N_poly)
1180      IF I_0=0 THEN                                     ! rein reelle Nullstelle
1190        FOR J=1 TO V_0
1200          U=M_r(N_poly-J+1)
1210          M_r(N_poly-J+1)=0
1220          FOR K=N_poly-J TO 0 STEP -1
1230            X=M_r(K)
1240            M_r(K)=U+R_0*M_r(K+1)
1250            U=X
1260          NEXT K
1270        NEXT J  ! v-fache Division des Nenners durch die aktuelle Nullstelle
1280      ELSE                                                 ! komplexe Nullstelle
1290        FOR J=1 TO V_0
1300          U=M_r(N_poly-J+1)
1310          V=M_i(N_poly-J+1)
1320          M_r(N_poly-J+1)=M_i(N_poly-J+1)=0
1330          FOR K=N_poly-J TO 0 STEP -1
1340            X=M_r(K)
1350            Y=M_i(K)
1360            M_r(K)=U+R_0*M_r(K+1)-I_0*M_i(K+1)
1370            M_i(K)=V+R_0*M_i(K+1)+I_0*M_r(K+1)
1380            U=X
1390            V=Y
1400          NEXT K
1410        NEXT J  ! v-fache Division des Nenners durch die aktuelle Nullstelle
1420      END IF
1430      N_stern=N_nenner=N_poly-V_0+1      ! Grad des abdividierten und mit 1/p
1440      MAT M_r_strich=ZER(0:N_poly-2)     ! multiplizierten Nennerpolynoms
1450      MAT M_i_strich=ZER(0:N_poly-2)
1460      REDIM M_r_strich(0:N_stern-1),M_i_strich(0:N_stern-1)
1470      FOR K=N_stern TO 1 STEP -1
1480        M_r(K)=M_r(K-1)            ! Multiplikation mit der Sprungfunktion 1/p
1490        M_r_strich(K-1)=K*M_r(K)                  ! Ableitung des Nennerpolynoms
1500        M_i(K)=M_i(K-1)            ! Multiplikation mit der Sprungfunktion 1/p
```

Listing 21, Fortsetzung

```
1510        M_i_strich(K-1)=K*M_i(K)              ! Ableitung des Nennerpolynoms
1520      NEXT K                   ! Multiplikation mit der Sprungfunktion 1/p
1530      M_r(0)=M_i(0)=0
1540      REDIM M_r(0:N_nenner),M_i(0:N_nenner)
1550      MAT N_r=M_r
1560      MAT N_i=M_i
1570      !
1580      MAT Z_r=ZER(0:N_max)
1590      MAT Z_i=ZER(0:N_max)
1600      MAT Z_r_strich=ZER(0:N_max-1)
1610      MAT Z_i_strich=ZER(0:N_max-1)
1620      MAT Z_r=Z
1630      N_zaehler=N_poly
1640      REDIM Z_r(0:N_poly),Z_i(0:N_poly)
1650      !
1660      J=0
1670      LOOP
1680        IF I_0=0 THEN                                 ! rein reelle Nullstelle
1690          R_zaehler=FNHorner(N_zaehler,R_0,Z_r(*))
1700          R_nenner=FNHorner(N_nenner,R_0,N_r(*))
1710          C(I,V_0-J-1)=R_zaehler/(FNN_fak(J)*FNN_fak(V_0-J-1)*R_nenner)
1720        ELSE                                           ! komplexe Nullstelle
1730          CALL Komplex_polynom(N_zaehler,R_0,I_0,Z_r(*),Z_i(*),U,V)
1740          CALL Komplex_polynom(N_nenner,R_0,I_0,N_r(*),N_i(*),X,Y)
1750          C_re=(U*X+V*Y)/(X*X+Y*Y)     ! komplexe Division +Re<(u+jv)/(x+jy)>
1760          C_im=-(X*V-U*Y)/(X*X+Y*Y)    ! komplexe Division -Im<(u+jv)/(x+jy)>
1770          IF C_re<>0 THEN
1780            IF C_re>=0 THEN
1790              Phi(I,V_0-J-1)=ATN(-C_im/C_re)
1800              IF Phi(I,V_0-J-1)<0 THEN Phi(I,V_0-J-1)=Phi(I,V_0-J-1)+2*PI
1810            ELSE
1820              Phi(I,V_0-J-1)=ATN(-C_im/C_re)+PI
1830            END IF
1840          ELSE
1850            Phi(I,V_0-J-1)=PI*(1+.5*SGN(C_im))
1860          END IF
1870          C(I,V_0-J-1)=2*SQR(C_re*C_re+C_im*C_im)
1880          C(I,V_0-J-1)=C(I,V_0-J-1)/(FNN_fak(J)*FNN_fak(V_0-J-1))
1890        END IF
1900      EXIT IF J=V_0-1
1910        J=J+1                              ! Berechnung der j-ten Ableitung
1920        REDIM Z_r_strich(0:N_zaehler-1),Z_i_strich(0:N_zaehler-1)
1930        FOR K=1 TO N_zaehler
1940          Z_r_strich(K-1)=K*Z_r(K)              ! Ableitung des Zählerpolynoms
1950          Z_i_strich(K-1)=K*Z_i(K)              ! Ableitung des Zählerpolynoms
1960        NEXT K
1970        IF I_0=0 THEN                                 ! rein reelle Nullstelle
1980          CALL Produkt(N_stern,N_zaehler-1,M_r(*),Z_r_strich(*),P_r(*))
1990          CALL Produkt(N_stern-1,N_zaehler,M_r_strich(*),Z_r(*),Z_r(*))
2000          CALL Addition(1,P_r(*),-J,Z_r(*),Z_r(*))
2010          CALL Produkt(N_nenner,N_stern,N_r(*),M_r(*),N_r(*))
2020        ELSE                                           ! komplexe Nullstelle
2030          CALL K_p(M_r(*),M_i(*),Z_r_strich(*),Z_i_strich(*),P_r(*),P_i(*))
2040          CALL K_p(M_r_strich(*),M_i_strich(*),Z_r(*),Z_i(*),Z_r(*),Z_i(*))
2050          CALL Addition(1,P_r(*),-J,Z_r(*),Z_r(*))
2060          CALL Addition(1,P_i(*),-J,Z_i(*),Z_i(*))
2070          CALL K_p(M_r(*),M_i(*),N_r(*),N_i(*),N_r(*),N_i(*))
2080        END IF
2090        N_zaehler=N_zaehler+N_stern-1
2100        N_nenner=N_nenner+N_stern
2110        REDIM Z_r(0:N_zaehler),Z_i(0:N_zaehler)
2120        REDIM N_r(0:N_nenner),N_i(0:N_nenner)
2130      END LOOP
2140    NEXT I
2150    MAT V=ZER
2160    MAT V=V_st
2170    SUBEXIT
2180    !
2190 Un_konj: !
2200    PRINT "Fehler bei der Nullstellenberechnung."
2210    PRINT "Es liegt eine unkonjugiert komplexe Nullstelle vor!"
2220    PAUSE
2230  SUBEND
2240  DEF FNN_fak(INTEGER N)
2250  !
```

Listing 21, Fortsetzung

```
2260  ! Funktions-Unterprogramm zur Berechnung von n-Fakultät
2270  !
2280  ! (c) 1985 by Jürgen Schwarz        Sprache: HP-BASIC     Datum: 01.06.85
2290  ! Speichermedium: Kassetten 73/74    File-Name: Heavis     Version:    1.0
2300  !
2310    INTEGER I
2320    REAL N_fak
2330    !
2340    IF N<0 THEN RETURN 0                            ! nicht definiertes Ergebnis
2350    IF (N=0) OR (N=1) THEN RETURN 1
2360    N_fak=1
2370    FOR I=2 TO N
2380      N_fak=I*N_fak
2390    NEXT I
2400    RETURN N_fak
2410  FNEND
```

Listing 22: Ausschnitt aus einem Programm zur Berechnung des Sperrschichttemperaturverlaufes von Halbleiterbauelementen aus dem Gehäusetemperaturverlauf.

```
10    ! ************************************************************************
20    ! *                                                                      *
30    ! * Programm ZHe_eK zur Berechnung des transienten Wärmewiderstandes und *
40    ! *   damit des Temperaturverlaufes in der virtuellen Sperrschicht in    *
50    ! *       einem einseitig gekühlten Halbleiterbauelement gegebenen       *
60    ! *    Temperaturverlauf am Case des Halbleiterbauelementes bei einem    *
70    ! *                Leistungssprung in der Sperrschicht                   *
80    ! *                                                                      *
90    ! *                 Anwendung der Zweitortheorie und                     *
100   ! *               des Heavisideschen Entwicklungssatzes                  *
110   ! *                                                                      *
120   ! ************************************************************************
130   ! *                                                                      *
140   ! * (c) 1985 by Jürgen Schwarz       Sprache: HP-BASIC  Datum: 01.07.85  *
150   ! * Speichermedium: Kassetten 73/74  File-Name: ZHe_eK  Version:    1.0  *
160   ! *                                                                      *
170   ! ************************************************************************

1490  ! ************************************************************************
1500  ! * Beginn der Berechnungen                                               *
1510  ! *                                                                      *
1520  ! * 1. Berechnung der Vierpolparameter des Thyristor-Ersatzschaltbildes  *
1530  ! *    und Transformation der Meßwerte der Temperatur am Übergang        *
1540  ! *    in den Bildbereich                                                *
1550  ! ************************************************************************
1560  !
1570  CALL Rc_vp_h(N_th,N_th_stern,Ch$,Rk_th(*),Ck_th(*),H_11(*),H_22(*),L(*))
1580  H_12(0)=1                                                       ! = H_21
1590  CALL Rc_zp_pb(M_kd,M_kd_stern,Kan,"H",R_kd(*),Tau_kd(*),Z_ga(*),N_ga(*))
1600  !
1610  ! ************************************************************************
1620  ! * 2. Berechnung der Sprungantworten (H-Parameter)                      *
1630  ! ************************************************************************
1640  !
1650  DISP "2.1  Berechnung des transienten Wärmewiderstandes [H_11(t)]"
1660  N_th_h=N_th_stern
1670  CALL Newton_mod(N_th_h,L(*),N_0(*),5E-8)                ! Pole der H-Matrix
1680  CALL Rc_t0_pb(N_th_h,H_11(*),L(*),N_0(*),H_th_11(*),Gamma_th(*))

1770  DISP "2.2  Berechnung der Sprungantwort H_12(t) [= H_21(t)]"
1780  CALL Rc_t0_pb(N_th_h,H_12(*),L(*),N_0(*),H_th_12(*),Gamma_th(*))

1860  DISP "2.3  Berechnung der Sprungantwort H_22(t)"
1870  CALL Rest_division(N_th,N_th_stern,I,H_22(*),L(*),Hf1(*),R_22(*),1E-6)
1880  REDIM Hf1(0:1)
1890  D_th_22=Hf1(1)
1900  S_th_22=Hf1(0)
1910  CALL Rc_t0_pb(N_th_h,R_22(*),L(*),N_0(*),H_th_22(*),Gamma_th(*))
```

Listing 22, Fortsetzung

```
2310   ! ********************************************************************
2320   ! * 4. Rechnung mit dem speziellen Temperaturverlauf                  *
2330   ! ********************************************************************
2340   !
2350   DISP "4.1  Berechnung des Verhalten des Systems"
2360   !
2370   N_sy=N_th_h+M_kd              ! vorläufige Anzahl der Systemzeitkonstanten
2380   REDIM R_ja(1:N_sy,0:1),K_pg(1:N_sy,0:1),Tau_sy(1:N_sy),N_0(1:N_sy)
2390   REDIM Gamma_th(1:N_th_h+1),Tau_kd(1:M_kd+1)
2400   MAT V_sy=CON(1:N_sy)
2410   Gamma_th(N_th_h+1)=Tau_kd(M_kd+1)=1E99
2420   I=0
2430   I_th=I_kd=1
2440   REPEAT
2450     I=I+1
2460     Tau_sy(I)=MIN(Gamma_th(I_th),Tau_kd(I_kd))
2470     N_0(I)=-1/Tau_sy(I)
2480     !
2490     Boolean_th=(DROUND(Tau_sy(I),6)=DROUND(Gamma_th(I_th),6))
2500     Boolean_kd=(DROUND(Tau_sy(I),6)=DROUND(Tau_kd(I_kd),6))
2510     IF Boolean_th THEN I_th=I_th+1
2520     IF Boolean_kd THEN I_kd=I_kd+1
2530     !
2540     IF Boolean_th AND Boolean_kd THEN
2550       V_sy(I)=2
2560       N_sy=N_sy-1                             ! zwei Zeitkonstanten sind gleich
2570       REDIM R_ja(1:N_sy,0:1),K_pg(1:N_sy,0:1)
2580       REDIM Tau_sy(1:N_sy),N_0(1:N_sy),V_sy(1:N_sy)
2590     END IF
2600   UNTIL I=N_sy
2610   MAT I_0=ZER(1:N_sy)
2620   REDIM Om(1:N_sy),P(1:N_sy,0:1)
2630   REDIM Gamma_th(1:N_th_h),Tau_kd(1:M_kd)
2640   !
2650   CALL Produkt(0,0,H_11(*),N_ga(*),Hf1(*))        ! Zählerpolynom der
2660   CALL Addition(1,Hf1(*),1,Z_ga(*),Z_ja(*))       ! Sperrschichttemperatur
2670   CALL Produkt(0,0,H_22(*),Z_ga(*),Hf1(*))        ! Zählerpolynom des
2680   CALL Addition(1,Hf1(*),1,N_ga(*),Z_pg(*))       ! Leistungsverlaufes
2690   CALL Produkt(0,0,L(*),N_ga(*),N_sy(*))          ! Nennerpolynom des Systems
2700   !
2710   CALL Hea(Z_ja(*),N_sy(*),N_0(*),I_0(*),R_ja(*),Tau(*),Om(*),P(*),1E-6,N_sy
,M_sy,V_sy(*))
2720   CALL Hea(Z_pg(*),N_sy(*),N_0(*),I_0(*),K_pg(*),Tau(*),Om(*),P(*),1E-6,N_sy
,M_sy,V_sy(*))



3440 Z_th: ! Unterprogramm zur Berechnung des transienten Widerstandes
3450   Z_th=C(0,0)
3460   FOR P=1 TO N
3470     Z_th=Z_th+C(P,0)*EXP(-T/Tau(P))
3480     IF V_sy(P)=2 THEN Z_th=Z_th+C(P,1)*T*EXP(-T/Tau(P))
3490   NEXT P
3500   RETURN
```

entsprechend den Voraussetzungen von Gl. (85) „herausgekürzt", so bleibt von ihm nur noch 1 und mit Gl. (85) für den bezogenen Temperaturverlauf am Übergang ergibt sich für die bezogene Sperrschichttemperatur über

$$T^*_{JA}(p) \;=\; H_{11}(p) \;+\; H_{12}(p)\, Z_{GA}(p) \qquad (126)$$

die Lösung bei Verwendung der Einzelpolynome aus Gl. (124) zu

$$T^*_{JA}(p) \;=\; \frac{h_{11}(p)\, n_{GA}(p) \;+\; h_{12}(p)\, z_{GA}(p)}{\ell(p)\, n_{GA}(p)} \qquad (127)$$

mit $h_{12}(p) = 1$. Für den bezogenen Verlauf der Leistung am Übergang ergibt sich entsprechend

$$P^*_{G}(p) \;=\; \frac{h_{21}(p)\, n_{GA}(p) \;+\; h_{22}(p)\, z_{GA}(p)}{\ell(p)\, n_{GA}(p)}$$

mit $h_{21}(p) = 1$. (128)

Die mit den Gln. (127) und (128) gewonnenen Ausdrücke müssen nun zurück in den Zeitbereich transformiert werden. Da sich nicht ausschließen läßt, daß einzelne Eigenzeitkonstanten des Halbleiterbauelementes (Nullstellen von $\ell(p)$) gleich mit einzelnen Zeitkonstanten der Exponentialsummenzerlegung des gemessenen Verlaufes sind, muß zur Rücktransformation das Programm nach Listing 21 zur Anwendung kommen, da nur dieses SUB-Programm mehrfache (hier höchstens doppelte) Nennernullstellen verarbeiten kann. Konjugiert komplexe Nullstellen sind hier allerdings auszuschließen.

Listing 22 zeigt Ausschnitte aus einem Rechenprogramm zur Abarbeitung der abgeleiteten Algorithmen. In den Zeilen 1570 und 1580 erfolgt die Berechnung der Hybridparameter des Halbleiterbauelementes aus den Daten des thermi-

Berlin, den 02. Juli 1985

Berechnung des Temperaturverlaufes in der Sperrschicht
von einseitig gekühlten Halbleiterbauelementen

Berechnung mit dem Heaviside'schen Entwicklungssatzes bei Vorgabe des
Temperaturverlaufes am Übergang zur Kühleinrichtung

ZHe_eK - Kassetten 73/74 - Version 1.0 - 01. Juli 1985 - Sz

Bauelement: Thyristor T 270 N 2000 ... 2600 (AEG AG)
Modell: Modell "junction-heatsink" [J-H]
Meßstelle: Anlageflächen des Kühlelementes (heatsink)

Daten der Ersatzschaltung des Halbleiterbauelementes in Kettenbruchschaltung:

R_k_TH (01)	=	4070.515 μK/W	C_k_TH (01)	=	557.913 mJ/K
R_k_TH (02)	=	7855.402 μK/W	C_k_TH (02)	=	606.904 mJ/K
R_k_TH (03)	=	20.908 mK/W	C_k_TH (03)	=	2437.699 mJ/K
R_k_TH (04)	=	47.166 mK/W	C_k_TH (04)	=	17.615 J/K
R_k_TH (05)	=	18.000 mK/W	C_k_TH (05)	=	76.365 J/K

Summe: R_k_TH = 98.000 mK/W

Parameter der Sprungantwort H 11(t):

H_11_JH (01)	=	721.732 μK/W	γ_11_JH (01)	=	1035.274 μs
H_11_JH (02)	=	5064.302 μK/W	γ_11_JH (02)	=	6744.398 μs
H_11_JH (03)	=	18.720 mK/W	γ_11_JH (03)	=	64.671 ms
H_11_JH (04)	=	19.332 mK/W	γ_11_JH (04)	=	661.397 ms
H_11_JH (05)	=	54.162 mK/W	γ_11_JH (05)	=	2110.162 ms

Summe: H_11_JH = 98.000 mK/W

Parameter der Sprungantwort H 12(t) [= H 21(t)]:

H_12_JH (01)	=	2270.215 $_{10}$-12	γ_12_JH (01)	=	1035.274 μs
H_12_JH (02)	=	-4543.482 $_{10}$-9	γ_12_JH (02)	=	6744.398 μs
H_12_JH (03)	=	3887.638 $_{10}$-6	γ_12_JH (03)	=	64.671 ms
H_12_JH (04)	=	-512.016 $_{10}$-3	γ_12_JH (04)	=	661.397 ms
H_12_JH (05)	=	1508.133 $_{10}$-3	γ_12_JH (05)	=	2110.162 ms

Summe: H_12_JH = 1000.000 $_{10}$-3

Parameter der Sprungantwort H 22(t):

S_22_JH	=	-55.556 ε(t) * W/K			
H_22_JH (01)	=	-268.678 pW/K	γ_22_JH (01)	=	1035.274 μs
H_22_JH (02)	=	4591.579 pW/K	γ_22_JH (02)	=	6744.398 μs
H_22_JH (03)	=	807.354 μW/K	γ_22_JH (03)	=	64.671 ms
H_22_JH (04)	=	13.561 W/K	γ_22_JH (04)	=	661.397 ms
H_22_JH (05)	=	41.994 W/K	γ_22_JH (05)	=	2110.162 ms

Summe: H_22_JH = 55.556 W/K

Bild 5: Ersatzschaltbild eines Thyristors und seine Hybridparameter im Zeitbereich.

Berlin, den 02. Juli 1985

Berechnung des Temperaturverlaufes in der Sperrschicht
von einseitig gekühlten Halbleiterbauelementen

Berechnung mit dem Heaviside'schen Entwicklungssatzes bei Vorgabe des
Temperaturverlaufes am Übergang zur Kühleinrichtung

ZHe_eK - Kassetten 73/74 - Version 1.0 - 01. Juli 1985 - Sz

Bauelement: Thyristor T 270 N 2000 ... 2600 (AEG AG)
Modell: Modell "junction-heatsink" [J-H]

Meßstelle: Anlageflächen des Kühlelementes (heatsink)
Kühlung: Aluminium-Kühlkörper bei verstärkter Luftkühlung
Kühlmittel: Luft mit einem Volumenstrom von 150 m^3/h

Daten des gemessenen Verlaufs der (bezogenen) Temperatur
am Übergang Halbleiterbauelement - Kühleinrichtung:

Verlustleistungssprung	P_J	=	300.0 W
Kühlmittelzuflußtemperatur	ϑ_A	=	40.0 °C

Z_11_HA^{x} (01)	=	-6254.525 μK/W	τ_11_HA^{x} (01)	=	661.397 ms
Z_11_HA^{x} (02)	=	55.425 mK/W	τ_11_HA^{x} (02)	=	7928.398 ms
Z_11_HA^{x} (03)	=	23.007 mK/W	τ_11_HA^{x} (03)	=	29.649 s
Z_11_HA^{x} (04)	=	77.823 mK/W	τ_11_HA^{x} (04)	=	130.488 s

Summe: Z_11_HA^{x} = 150.000 mK/W

Parameter der Antwort des Systems an der Sperrschicht:

C_th_JA (00,0)	=	248.000 mK/W			
C_th_JA (01,0)	=	-721.732 μK/W	τ_th_JA (01)	=	1035.274 μs
C_th_JA (02,0)	=	-5064.302 μK/W	τ_th_JA (02)	=	6744.398 μs
C_th_JA (03,0)	=	-18.721 mK/W	τ_th_JA (03)	=	64.671 ms
C_th_JA (04,0)	=	-29.868 mK/W	τ_th_JA (04)	=	661.397 ms
C_th_JA (04,1)	=	-4841.881 μK/Ws			
C_th_JA (05,0)	=	-5519.167 μK/W	τ_th_JA (05)	=	2110.162 ms
C_th_JA (06,0)	=	-83.160 mK/W	τ_th_JA (06)	=	7928.398 ms
C_th_JA (07,0)	=	-25.397 mK/W	τ_th_JA (07)	=	29.649 s
C_th_JA (08,0)	=	-79.549 mK/W	τ_th_JA (08)	=	130.488 s

Parameter des Leistungsverlaufes am Übergang [H]:

C_pg_H (00,0)	=	1000.000 $_{10}$-3			
C_pg_H (01,0)	=	-2270.214 $_{10}$-12	τ_pg_H (01)	=	1035.274 μs
C_pg_H (02,0)	=	4543.482 $_{10}$-9	τ_pg_H (02)	=	6744.398 μs
C_pg_H (03,0)	=	-3887.746 $_{10}$-6	τ_pg_H (03)	=	64.671 ms
C_pg_H (04,0)	=	210.357 $_{10}$-3	τ_pg_H (04)	=	661.397 ms
C_pg_H (04,1)	=	128.238 $_{10}$-3/s			
C_pg_H (05,0)	=	-153.681 $_{10}$-3	τ_pg_H (05)	=	2110.162 ms
C_pg_H (06,0)	=	-912.551 $_{10}$-3	τ_pg_H (06)	=	7928.398 ms
C_pg_H (07,0)	=	-81.148 $_{10}$-3	τ_pg_H (07)	=	29.649 s
C_pg_H (08,0)	=	-59.095 $_{10}$-3	τ_pg_H (08)	=	130.488 s

Bild 6: Ergebnis der Exponentialsummenzerlegung des Temperaturverlaufes am Übergang und numerische Darstellung der Lösung.

schen Ersatzschaltbildes nach den Ausführungen in [21] und [22]. Anschließend wird das Ergebnis der Exponentialsummenzerlegung (nach [18] und [19]) in den Bildbereich transformiert. In Zeile 1670 werden die Pole der H-Matrix (Nullstellen von $\ell(p)$) errechnet und anschließend werden die H-Parameter zur Demonstration in den Zeitbereich transformiert. Da $H_{22}(p)$ eine unechtgebrochene rationale Funktion ist, wird der ganzrationale Anteil mit Hilfe des SUB-Programmes Rest_division abgespalten. Der Wert von Hf1(1) entspricht dabei dem DIRAC-Impuls $\delta(t)$ und der von Hf1(0) dem Einheitssprung $\epsilon(t)$. In den Zeilen 2370 bis 2630 erfolgt eine Prüfung auf das Auftreten gleicher Nullstellen in den Polynomen $\ell(p)$ und $n_{GA}(p)$. Dabei ist vorausgesetzt, daß die Zeitkonstanten in den Vektoren Gamma_th(∗) und Tau_kd(∗) aufsteigend sortiert vorliegen. Die eigentliche Berechnung erfolgt in den Zeilen 2650 und 2720 konsequent nach den Gln. (127) und (128). Anschließend ist noch ein Unterprogramm zur konkreten Berechnung der Ergebnisse im Zeitbereich dargestellt. Die in dem Listing 22 fehlenden Zeilen dienen der Ein- und Ausgabe der Ergebnisse und zur Steuerung der Graphik.

Die Bilder 5 bis 7 zeigen konkrete Rechenergebnisse. Im Bild 5 sind die thermischen Eigenschaften des betrachteten Halbleiterbauelementes dargestellt. Die gemessenen und ausgewerteten Ausgangsdaten sowie die Lösung in ihrer numerischen Form sind im Bild 6 abgedruckt, während Bild 7 die entsprechenden graphischen Darstellungen zeigt.

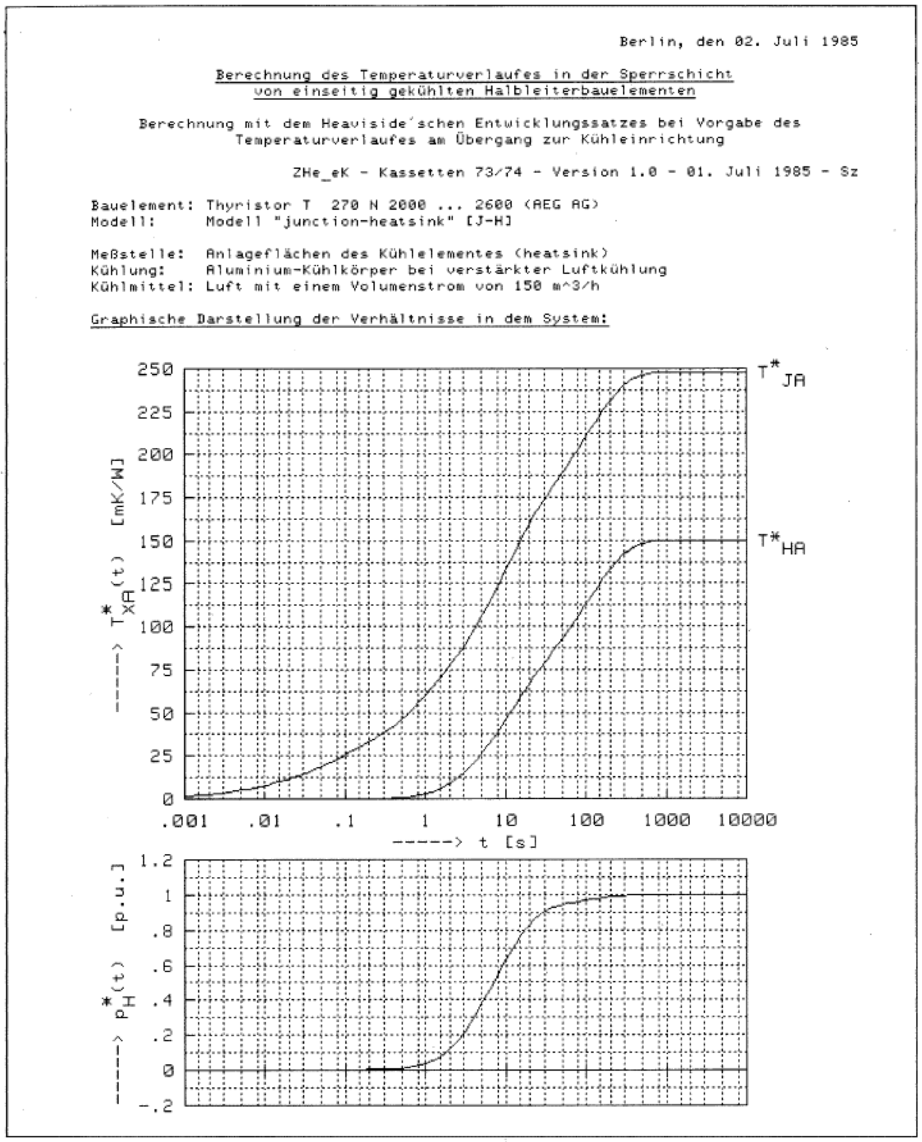

Bild 7: Graphische Darstellung der berechneten Verläufe der Temperatur in der Sperrschicht und des Leistungsverlaufes am Übergang und des gemessenen Temperaturverlaufes am Übergang.

10 Zusammenfassung

In der nun vollständig vorliegenden Arbeit wurden einige Möglichkeiten zur Anwendung von Polynomen aufgezeigt. Dabei wurde weniger Wert auf mathematische Exaktheit und mehr Wert auf praktische Brauchbarkeit und leichtes Verständnis der Abhandlungen gelegt. Die meisten skizzierten Anwendungsfälle wurden mit praktisch erprobten BASIC-Programmen in strukturiertem HP-BASIC, dessen Sprachumfang mit normalem BASIC kaum vergleichbar ist und mehr an ALGOL oder PASCAL erinnert, unterstützt.

Die Programme sind für die konkrete Arbeit im Labor entstanden und sind deshalb in bezug auf Rundungseigenschaften und Fehlerbehandlung nur soweit optimiert worden, wie es für die Anwendungsfälle nötig war. Es sind sicher Aufgabenstellungen denkbar, bei denen Verbesserungen erforderlich sind. Aber diese Serie sollte ja in erster Linie Anregungen zur Weiterarbeit und keine Rezepte liefern. Der Autor hofft, daß die Arbeit auch so verstanden und benutzt wird.

Literatur

[13] G. DOETSCH: „Handbuch der Laplace-Transformation", Bd. I, II, III. Birkhäuser Verlag, Basel, Stuttgart (1950 – 56).

[14] G. DOETSCH: „Anleitung zum praktischen Gebrauch der Laplace-Transformation und der Z-Transformation". Oldenbourg-Verlag München, Wien (1967).

[15] K. SIMONYI: „Theoretische Elektrotechnik". Dtsch. Vlg. Wissenschaften, Berlin (1977).

[16] U. GRIGULL und H. SANDNER: „Wärmeleitung". Springer-Verlag Berlin, Heidelberg, New York (1979).

[17] C. L. BEUKEN: „Wärmeverluste bei periodisch betriebenen elektrischen Öfen." Dissertation Sächsische Bergakademie Freiberg (1936).

[18] W. KÖCHLI: „Identifikation des thermischen Verhaltens einer Hochleistungsdiode". Dissertation ETH Zürich (1969).

[19] W. BÜTTNER: „Ein numerisches Verfahren zur Exponentialapproximation von transienten Wärmewiderständen". Archiv für Elektrotechnik 59 (1977) 351 – 359.

[20] H. MÜLLER: „Berechnung des transienten thermischen Verhaltens von Halbleiterventilen im Bereich kurzer Zeiten". Dissertation RWTH Aachen (1972).

[21] J. SCHWARZ: „Kühlung von Leistungshalbleitern". Elektronik Journal 20 (1985) 13/14, 38 – 43

[22] J. SCHWARZ: „Junction- und Case-Temperatur". Elektronik Journal 20 (1985) 17, 60 – 66

Abstract

Jürgen Schwarz

298 Behandlung von Polynomen – Teil 5

In diesem letzten Teil der Serie geht es um konkrete Anwendungen der bisher betrachteten SUB-Programme.