



Programm zur wechselseitigen Transformation von RC-Partial- und RC-Kettenbruchsaltungen

Thermische Ersatzschaltbilder

Dipl.-Ing. Jürgen Schwarz

Die angegebenen Subroutinen-Programme in strukturiertem HP-Basic ermöglichen die Umrechnung zwischen dem praktisch verwendeten und dem physikalisch sinnvollen thermischen Ersatzschaltbild z. B. von elektronischen Bauelementen.

Partialbrüche in Kettenbrüche

Das thermische Verhalten von Halbleiterbauelementen und auch von anderen Einrichtungen kann durch Ersatzschaltbilder aus einer Vielzahl von thermischen Widerständen und thermi-

schen Kapazitäten in einer Kettenbruchsaltung nach Bild 1 beschrieben werden /1/. Mit diesem physikalisch sinnvollen Modell kann man nur schlecht arbeiten, so daß die Umrechnung in ein äquivalentes Ersatzschaltbild in Partialbruchsaltung nach Bild 2 zweckmäßig ist. Eine solche Umrechnung ist immer möglich, und beide Ersatzschaltbilder sind kanonisch, d. h., sie weisen beide eine minimale Zahl von Bauelementen auf /2/. Ein Algorithmus zur Umrechnung wurde in /3/ abgeleitet, allerdings ohne die Angabe eines Programms. Dies soll hier nachgeholt werden.

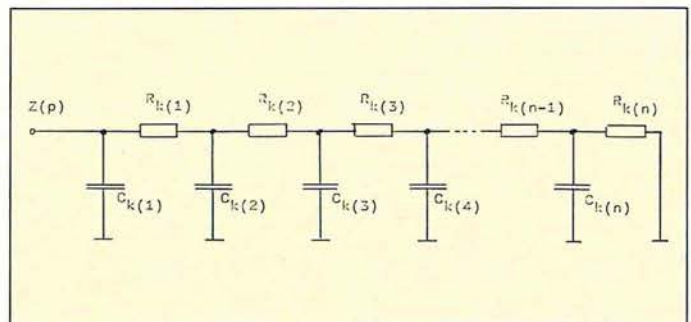


Bild 1: Thermisches Ersatzschaltbild in Kettenbruchsaltung nach /1/

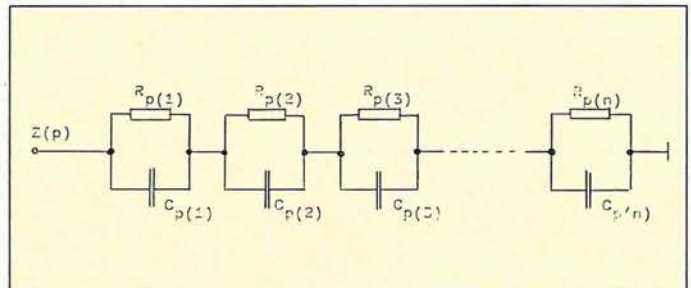


Bild 2: Thermisches Ersatzschaltbild in Partialbruchsaltung

Analyse der Partialbruchsaltung

Die Impedanz einer RC-Partialbruchsaltung setzt sich aus der Summe der Impedanzen der einzelnen Glieder (Partialbrüche)

$$Z(p) = \sum_{i=1}^n \frac{R_{p(i)}}{1 + p \cdot R_{p(i)} \cdot C_{p(i)}} \quad (1)$$

zusammen. Dabei kann das Produkt $R_{p(i)} \cdot C_{p(i)}$ zu $\tau_{p(i)}$, der Zeitkonstanten der einzelnen RC-Glieder zusammengefaßt werden. Die Gl. (1) läßt sich unter der Voraussetzung, daß alle Zeitkonstanten unterschiedlich sind, durch Erweitern in

$$Z(p) = \frac{\sum_{i=1}^n \left[R_{p(i)} \cdot \prod_{\substack{j=1 \\ j \neq i}}^n (1 + p \cdot \tau_{p(j)}) \right]}{\prod_{i=1}^n (1 + p \cdot \tau_{p(i)})} \quad (2)$$

umwandeln. Dies entspricht

$$Z(p) = \frac{R_{p(1)} \cdot (1 + p \cdot \tau_{p(2)}) (1 + p \cdot \tau_{p(3)}) \dots (1 + p \cdot \tau_{p(n)}) + \dots}{(1 + p \cdot \tau_{p(1)}) \cdot (1 + p \cdot \tau_{p(2)}) \cdot (1 + p \cdot \tau_{p(3)}) \cdot \dots}$$

$$\frac{\dots + R_{p(2)} \cdot (1 + p \cdot \tau_{p(1)}) (1 + p \cdot \tau_{p(3)}) \dots (1 + p \cdot \tau_{p(n)}) + \dots}{\dots \cdot (1 + p \cdot \tau_{p(n-1)}) \cdot (1 + p \cdot \tau_{p(n)}}$$

$$\dots + R_{p(n)} \cdot (1 + p \cdot \tau_{p(1)}) \dots (1 + p \cdot \tau_{p(n-1)}) \quad (3)$$

einem Ausdruck, der ausmultipliziert die Form

$$Z(p) = \frac{A(p)}{B(p)} = \frac{a_0 + a_1 p + a_2 p^2 + \dots + a_{n-1} p^{n-1}}{b_0 + b_1 p + b_2 p^2 + \dots + b_{n-1} p^{n-1} + b_n p^n} \quad (4)$$

hat. Ist die oben genannte Voraussetzung ungleicher Zeitkonstanten nicht erfüllt, dann ist die Partialbruchsaltung nicht kanonisch und die Glieder mit gleichen Zeitkonstanten können zusammengefaßt werden.

Um die Koeffizienten der Gl. (4) zu berechnen, wird folgende Methode verwendet:

Gegeben sei ein Polynom

$$C(p) = c_0 + c_1 p + c_2 p^2 + \dots + c_n p^n, \quad (5)$$

das mit $(1 + p \cdot \tau)$ multipliziert werden soll. Dann ergibt sich das Produkt zu

$$(1 + p \cdot \tau) \cdot C(p) = c_0 + (c_1 + c_0 \tau) p + (c_2 + c_1 \tau) p^2 + \dots + (c_n + c_{n-1} \tau) p^n + c_n \cdot \tau \cdot p^{n+1}. \quad (6)$$



$$\begin{aligned}
 1 &= b_0 \\
 \downarrow \\
 R_{k(n)} &= \frac{R_{k(n)}}{b_0} = \frac{a_0}{b_0} \\
 \downarrow \\
 \frac{1}{R_{k(n)}} + p C_{k(n)} &= \frac{1 + p R_{k(n)} C_{k(n)}}{R_{k(n)}} = \frac{b_0 + p b_1}{a_0} \\
 \downarrow \\
 R_{k(n-1)} + \frac{R_{k(n)}}{1 + p R_{k(n)} C_{k(n)}} &= \frac{(R_{k(n)} + R_{k(n-1)}) + p(R_{k(n)} R_{k(n-1)} C_{k(n)})}{1 + p R_{k(n)} C_{k(n)}} \\
 &= \frac{a_0 + a_1 p}{b_0 + b_1 p} \\
 \downarrow \\
 \frac{b_0 + b_1 p}{a_0 + a_1 p} + p C_{k(n-1)} &= \frac{b_0 + b_1 p + b_2 p^2}{a_0 + a_1 p} \quad (11)
 \end{aligned}$$

Diese Verfahrensweise wird bis zum Ergebnis nach Gl. (4) fortgesetzt.

Synthese der Partialbruchschaltung

Zur Berechnung der Elemente der Partialbruchschaltung nach Bild 2 ist der rationale Bruch nach Gl. (4) in Partialbrüche zu zerlegen:

$$\begin{aligned}
 Z(p) = \frac{A(p)}{B(p)} &= \frac{A_1}{p - p_1} + \frac{A_2}{p - p_2} + \dots + \frac{A_n}{p - p_n} \\
 &= \sum_{i=1}^n \frac{A_i}{p - p_i} \quad (12)
 \end{aligned}$$

Dabei sind die p_1 bis p_n die Wurzeln des Nennerpolynoms $B(p) = 0$. (13)

Wie in /2/ bewiesen wurde, liegen diese für die hier vorliegenden Schaltungen sämtlich auf der negativen reellen Achse und sind auch alle voneinander verschieden (keine mehrfachen Nullstellen).

Um in Gl. (12) eine Konstante A_i zu bestimmen, multipliziert man die Gleichung mit $p - p_i$:

$$\begin{aligned}
 \frac{A(p)}{B(p)} (p - p_i) &= \frac{p - p_i}{p - p_1} A_1 + \frac{p - p_i}{p - p_2} A_2 + \dots + A_i + \dots + \frac{p - p_i}{p - p_n} \\
 &= A_i + (p - p_i) \cdot \sum_{\substack{j=1 \\ j \neq i}}^n \frac{A_j}{p - p_j} \quad (14)
 \end{aligned}$$

Wenn man nun den Grenzwert für $p \rightarrow p_i$ untersucht, dann geht Gl. (14) in

$$A_i = \lim_{p \rightarrow p_i} \frac{A(p) \cdot (p - p_i)}{B(p)} \quad (15)$$

über. Da p_i eine Wurzel von $B(p)$ ist, wird mit $p \rightarrow p_i$ die Funktion $B(p) \rightarrow B(p_i) = 0$, und der Grenzübergang Gl. (15) ergibt einen unbestimmten Ausdruck. Nach der Regel von l'Hospital ergibt sich durch Bildung der ersten Ableitung

$$\begin{aligned}
 A_i &= \lim_{p \rightarrow p_i} \frac{\frac{d}{dp} [(p - p_i) \cdot A(p)]}{\frac{d}{dp} B(p)} = \lim_{p \rightarrow p_i} \frac{A(p) + (p - p_i) \cdot A'(p)}{B'(p)} \\
 &= \frac{A(p_i)}{B'(p_i)} \quad (16)
 \end{aligned}$$

ein Ausdruck für die einzelnen Koeffizienten. Die erste Ableitung von $B(p)$ ist bekanntlich

$$B'(p) = \sum_{i=1}^n i \cdot b_i \cdot p^{i-1} \quad (17)$$

Vergleicht man Gl. (12) mit Gl. (1), kann für die einzelnen Elemente der RC-Partialbruchschaltung

$$R_{p(i)} = - \frac{A(p_i)}{p_i \cdot B'(p_i)} \quad (18)$$

$$C_{p(i)} = \frac{B'(p_i)}{A(p_i)} \quad (19)$$

angegeben werden. Die Herleitung entspricht dem Satz von der Aufspaltung.

Umsetzung in ein Rechenprogramm

Listing 3 zeigt ein Subprogramm zur Durchführung der Transformation und Listing 2 ein dazu notwendiges Nullstellenberechnungsprogramm, welches in Zeile 410 des ersten Programms aufgerufen wird. Das Subprogramm Rc_tr_kp enthält keine Besonderheiten und arbeitet streng nach den oben abgeleiteten Algorithmen.

Zur Nullstellenbestimmung wurde ein nach dem Newtonverfahren arbeitendes Programm verwendet /4/. Dies ist hier möglich, da keine konjugiert komplexen und auch keine mehrfachen Nullstellen auftreten können. Die Berechnung der Zahlenwerte erfolgt hier mit dem vollständigen Horner Schema.

Literatur:

- /1/ Beuken, C. L.: Wärmeverluste bei periodisch betriebenen elektrischen Öfen. Dissertation Sächsische Bergakademie Freiberg 1936.
- /2/ Cauer, W.: Die Verwirklichung von Wechselstromwiderständen vorgeschriebener Frequenzabhängigkeit. Archiv für Elektrotechnik Bd. 17 (1926) H. 4, S. 355-388.
- /3/ Müller, H.: Beziehungen zwischen dem praktisch verwendeten und dem physikalisch sinnvollen Wärmeersatzschaltbild von Dioden und Thyristoren. Archiv für Elektrotechnik Bd. 54 (1971) H. 3, S. 170-176.
- /4/ Zurmühl, R.: Praktische Mathematik für Ingenieure und Physiker. Berlin, Heidelberg, New York: Springer-Verlag 1965.



```

10  SUB Rc_tr_pk<INTEGER N,Kanonisch,REAL Rp(*),Cp(*),Rk(*),Ck(*)>
20  !
30  ! SUB-Programm Rc_tr_pk zur Transformation einer aus Widerständen und
40  ! Kondensatoren bestehenden kanonischen Partialbruchsaltung in eine
50  ! aus gleichartigen Bauelementen bestehende äquivalente, kanonische
60  ! Kettenbruchsaltung
70  !
80  ! Eingabedaten: n      ... Anzahl der RC-Glieder
90  !                  Rp(1:n) ... Widerstände der Partialbruchsaltung
100 !                  Cp(1:n) ... Kapazitäten der Partialbruchsaltung
110 ! Ergebnis:   kanonisch ... 1 für eine kanonische Partialbruchsaltung,
120 !                  sonst 0 (und ohne Lösung)
130 !                  Rk(1:n) ... Widerstände der Kettenbruchsaltung
140 !                  Ck(1:n) ... Kapazitäten der Kettenbruchsaltung
150 !
160 !                  Programmierer:  Jüngen Schwarz
170 !                  Programm-Name:   RCTRpk
180 !                  Datum/Variante:  14.02.85 / 02
190 !                  Speichermedium:  Kassetten 57/58
200 !
210  INTEGER I,J,K
220  REAL A(0:N-1),A_stern(0:N-1),B(0:N),Taup(1:N),Taup_stern(1:N)
230  !
240  MAT Rk=ZER
250  MAT Ck=ZER
260  REDIM Rp(1:N),Cp(1:N),Rk(1:N),Ck(1:N)
270  MAT Taup=Rp.Cp      ! Zeitkonstanten der einzelnen RC-Glieder
280  MAT Taup_stern=Taup
290  MAT SORT Taup_stern      ! Sortierung nach aufsteigender Größe
300  Kanonisch=1
310  FOR I=2 TO N
320    Kanonisch=Kanonisch AND (Taup_stern(I-1)<>Taup_stern(I))
330  NEXT I
340  IF NOT Kanonisch THEN
350    PRINT "Die vorgegebene Partialbruchsaltung ist nicht kanonisch!"
360    SUBEXIT
370  END IF
380  !
390  B(0)=1
400  FOR I=1 TO N
410    FOR J=I TO 1 STEP -1
420      B(J)=B(J)+B(J-1)*Taup(I) ! Parameter des Nennerpolynoms der Impedanz
430    NEXT J
440  NEXT I
450  !
460  MAT Taup_stern=Taup
470  FOR I=N TO 1 STEP -1
480    IF I<N THEN Taup_stern(I)=Taup(I+1)
490    MAT A_stern=ZER
500    A_stern(0)=1
510    FOR J=1 TO N-1
520      FOR K=J TO 1 STEP -1
530        A_stern(K)=A_stern(K)+A_stern(K-1)*Taup_stern(J)
540      NEXT K
550    NEXT J
560    MAT A_stern=(Rp(I))*A_stern
570    MAT A=A+A_stern      ! Parameter des Zählerpolynoms der Impedanz
580  NEXT I
590  !
600  FOR I=1 TO N
610    Ck(I)=B(N+1-I)/A(N-I)      ! Element der Kettenbruchsaltung
620    FOR J=1 TO N+1-I
630      B(J)=B(J)-Ck(I)*A(J-1)
640    NEXT J
650    Rk(I)=A(N-I)/B(N-I)      ! Element der Kettenbruchsaltung
660    FOR J=0 TO N-I
670      A(J)=A(J)-Rk(I)*B(J)
680    NEXT J
690  NEXT I
700  SUBEND

```

Listing 1: Transformation einer Partialbruchsaltung in eine Kettenbruchsaltung



```

10 SUB Newton_mod(INTEGER N_polynom,REAL R_koeff(*),R_wurzel(*),Epsilon)
20 !
30 ! SUB-Programm zur Berechnung der Nullstellen (Wurzeln) eines Polynoms
40 ! f(x) vom Grad n mit reellen Koeffizienten und nur reellen Nullstellen
50 ! (modifiziert, hier nur für Nullstellen auf der negativen reellen Achse)
60 !
70 ! Eingabedaten:
80 !     n_polynom           ... Grad des Polynoms
90 !     r_koeff(0:n_polynom) ... Koeffizienten des Polynoms
100 !     epsilon            ... Genauigkeitsschranke für die Nullstellen
110 ! Ergebnis:
120 !     r_wurzel(1:n_polynom) ... Nullstellen (Wurzeln) des Polynoms
130 !
140 !     Programmierer:   Jürgen Schwarz
150 !     Programm-Name:   Neut_m
160 !     Datum/Variante:  14.02.85 / 02
170 !     Speichermedium:  Kassetten 57/58
180 !
190 INTEGER J,K,N
200 REAL Delta_x,X
210 REAL A(0:N_polynom),B(0:N_polynom),C(1:N_polynom)
220 !
230 ! Datensicherung
240 !
250 N=N_polynom
260 REDIM R_koeff(0:N_polynom)
270 MAT A=R_koeff
280 MAT R_wurzel=ZER
290 REDIM R_wurzel(1:N_polynom)
300 !
310 ! Beginn der Lösung
320 !
330 X=0
340 WHILE N>1
350     C(N)=B(N)=A(N)
360     FOR K=0 TO 2             ! zweimalige Nachiteration
370         REPEAT
380             FOR J=N-1 TO 1 STEP -1
390                 B(J)=A(J)+X*B(J+1)
400                 C(J)=B(J)+X*C(J+1)
410             NEXT J
420             B(0)=A(0)+X*B(1)
430             IF C(1)/A(N)<=0 THEN
440                 PRINT "Fehler im SUB-Programm Newton_mod!"
450                 PRINT "Ergebnis konvergiert nicht!"
460                 PAUSE
470             END IF
480             Delta_x=-B(0)/C(1)
490             X=X+Delta_x
500             DISP "N =";N;"   Delta_x =";Delta_x;"   Epsilon =";Epsilon
510             UNTIL ABS(Delta_x)<Epsilon
520         NEXT K
530         R_wurzel(N)=X           ! Lösung gefunden
540         FOR J=0 TO N-1
550             A(J)=B(J+1)       ! Reduktion der Ausgangsgleichung
560         NEXT J
570         N=N-1
580     END WHILE
590     X=-A(0)/A(1)             ! Lösung für den letzten Schritt bzw. n = 1
600     R_wurzel(1)=X
610 SUBEND

```

Listing 2: Nullstellenberechnung eines Polynoms $f(x)$



```

10  SUB Rc_tr_kp<INTEGER N,REAL Rk(*),Ck(*),Rp(*),Cp(*)>
20  !
30  ! SUB-Programm Rc_tr_kp zur Transformation einer aus Widerständen und
40  ! Kondensatoren bestehenden (kanonischen) Kettenbruchschialtung in eine
50  ! aus gleichartigen Bauelementen bestehende äquivalente, kanonische
60  ! Partialbruchschialtung
70  !
80  ! Eingabedaten:  n      ... Anzahl der RC-Glieder
90  !                Rk(1:n) ... Widerstände der Kettenbruchschialtung
100 !                Ck(1:n) ... Kapazitäten der Kettenbruchschialtung
110 ! Ergebnis:      Rp(1:n) ... Widerstände der Partialbruchschialtung
120 !                Cp(1:n) ... Kapazitäten der Partialbruchschialtung
130 !
140 !                Programmierer:  Jürgen Schwarz
150 !                Programm-Name:   RCTRkp
160 !                Datum/Variante:  14.02.85 / 02
170 !                Speichermedium:  Kassetten 57/58
180 !
190  INTEGER I,J
200  REAL B_0,R_zaeher,R_nenner
210  REAL A(0:N-1),B(0:N),B_0(1:N)
220  !
230  MAT Rp=ZER
240  MAT Cp=ZER
250  REDIM Rk(1:N),Ck(1:N),Rp(1:N),Cp(1:N)
260  !
270  ! Berechnung der Koeffizienten der Impedanz
280  !
290  B(0)=1
300  FOR I=N TO 1 STEP -1
310    FOR J=0 TO N-I
320      A(J)=A(J)+Rk(I)*B(J)          ! Parameter des Zählerpolynoms
330    NEXT J
340    FOR J=1 TO N+1-I
350      B(J)=B(J)+Ck(I)*A(J-1)      ! Parameter des Nennerpolynoms
360    NEXT J
370  NEXT I
380  !
390  ! Nullstellenbestimmung und Zerlegung mit dem "Satz von der Aufspaltung"
400  !
410  CALL Newton_mod(N,B(*),B_0(*),1E-9)
420  FOR I=1 TO N
430    B_0=B_0(I)
440    R_nenner=R_zaeher=0
450    FOR J=N-1 TO 0 STEP -1
460      R_zaeher=A(J)+B_0*R_zaeher
470      R_nenner=(J+1)*B(J+1)+B_0*R_nenner
480    NEXT J
490    Rp(I)=-R_zaeher/(B_0*R_nenner) ! Element der Partialbruchschialtung
500    Cp(I)=R_nenner/R_zaeher       ! Element der Partialbruchschialtung
510  NEXT I
520  SUBEND

```

Listing 3: Transformation einer Kettenbruchschialtung in eine Partialbruchschialtung