

MASTER

Realisatiemogelijkheden met het Sallen- en Keyfilter

Backx, A.C.P.M.

Award date:
1977

[Link to publication](#)

Disclaimer

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

TECHNISCHE HOGESCHOOL EINDHOVEN
AFDELING DER ELEKTROTECHNIEK
VAKGROEP ET

Realisatiemogelijkheden met het
Sallen- en Keyfilter,
door
A.C.P.M. Backx.
ET-15-77



Verslag van een afstudeeronderzoek,
verricht in de vakgroep ET,
onder leiding van ir. Tirtoprodjo,
in de periode oktober 1976-oktober 1977.

306360

Samenvatting

Met dit onderzoek heb ik geprobeerd een inzicht te krijgen in de mogelijke poolrealisaties met het Sallen en Key actief laagdoorlaatfilter voor diverse ordes (tweede, derde en vierde orde) onder bepaalde voorwaarden.

Voor het tweede orde filter met gelijke capaciteiten bleek een versterkingsfactor twee voor de gebruikte versterker voldoende om elk stabiel filter te kunnen realiseren; elk filter is realiseerbaar vanaf een bepaalde ondergrens voor de versterkingsfactor, welke wordt bepaald door de poolligging. Voor versterkingsfactoren kleiner dan twee, waarvoor realisatie mogelijk is, bestaan twee of meer sets van oplossingen voor de weerstanden bij dezelfde versterkingsfactor. Dit geldt ook voor het derde en vierde orde filter.

Indien alle componenten vrij te kiezen zijn is een versterkingsfactor één toereikend om elk gewenst stabiel filter te realiseren; dit geldt ook zo voor het derde orde filter.

Derde en vierde orde filters met gelijke capaciteiten zijn voor versterkingsfactoren groter dan drie nagenoeg geen oplossingen mogelijk.

Voor het derde orde filter konden formules worden afgeleid, waarmee op eenvoudige wijze rechtstreeks uit de gewenste poolligging de componenten kunnen worden berekend (capaciteiten gelijk); er is slechts een kleine strook in het linkerhalfvlak langs de imaginaire as waar geen poolligging gerealiseerd kan worden.

Inhoudsopgave

| | | |
|-------|---|----|
| 1. | Inleiding | 1 |
| 2. | Tweede orde Sallen en Key filter | |
| 2.1 | Afleiding van de overdrachtsfunctie | 2 |
| 2.2 | Enige beschouwingen aan het filter | 3 |
| 2.3 | Grens van de realisatiemogelijkheden van het filter met gelijke capaciteiten als functie van de versterkingsfactor | 6 |
| 2.4 | Nomogrammen voor filters | 10 |
| | Derde orde Sallen en Key filter | |
| | Afleiding van de overdrachtsfunctie | 13 |
| 3.2 | Enige beschouwingen aan het filter | 15 |
| 3.3 | Zoeken naar een rechtstreeks verband tussen poolligging en weerstandswaarden van het filter met gelijke capaciteiten | 19 |
| 3.4 | Realisatiemogelijkheden met het filter als functie van de versterkingsfactor | 31 |
| 3.4.1 | Afleiding van een uiterste bovengrens van het realiseerbare gebied als functie van de versterkingsfactor | 31 |
| 4.2 | Berekening van de grenskrommen met de computer | 36 |
| 5 | Bepaling van nomogrammen voor enkele filters | 57 |
| 4. | Vierde orde Sallen en Key filter | |
| 4.1 | Afleiding van de overdrachtsfunctie | 69 |
| 4.2 | Enige beschouwingen aan het filter | 73 |
| 4.3 | Mogelijke realisaties door het filter met versterkingsfactor $k = 2$ en gelijke capaciteiten als functie van de ligging van twee vastgelegde, genormeerde polen | 77 |

| | | |
|------------|---|-----|
| 4.4 | Bepaling van nomogrammen voor diverse filters | 84 |
| 5. | Fysische beschouwingen van de diverse filters | 96 |
| | Conclusies en opmerkingen | 103 |
| | Literatuuropgave | 105 |
| Bijlage 1 | Programma voor de berekening van grenskrommen voor het derde orde filter | |
| Bijlage 2 | Programma voor de berekening van grenskrommen voor het vierde orde filter | |
| Bijlage 3 | Programma voor de berekening van nomogrammen voor het vierde orde filter | |
| Bijlage 4 | Enige numerieke beschouwingen | |
| Appendix 1 | Equal valued capacitor active RC network realisation of a 3rd order lowpass Butterworth characteristic | |
| Appendix 2 | Equal valued capacitor active RC network realisation of a fourth order lowpass Butterworth characteristic | |

1. Inleiding

In de literatuur is reeds vrij veel gepubliceerd over het derde orde Sallen en Key filter (zie lit. 1, 2, 3, 7) en er zijn enkele publicaties verschenen over het vierde orde filter (onder meer lit. 5).

De meeste publicaties hebben betrekking op realisaties van speciale filters zoals Butterworth, Tchebyshev, Bessel en anderen en geven geen inzicht in wat nu de realiseerbare poolliggingen zijn voor de diverse orde filters.

Tijdens dit onderzoek heb ik geprobeerd een algemeen inzicht te krijgen in wat theoretisch realiseerbaar is met een filter van een bepaalde orde (tot en met de vierde orde). Ik heb getracht formules af te leiden voor de diverse orde filters, waarmee, door eenvoudig de te realiseren poolligging in het complexe frequentievlak te vullen, de componentwaarden kunnen worden berekend. Tevens heb ik grenzen bepaald die aangeven welke poolliggingen realiseerbaar zijn voor een bepaald filter onder bepaalde voorwaarden.

De meeste berekeningen heb ik uitgevoerd aan filters, waarin alle capaciteiten gelijk gesteld zijn en genormeerd zijn op één Farad, omdat dit vanuit praktisch oogpunt erg interessante filters zijn (een filter waarin een spreiding in de capaciteitswaarden optreedt van een factor 10^4 is namelijk al vrij moeilijk realiseerbaar). Een gedeelte van de resultaten van dit onderzoek is inmiddels al behandeld op het twentieth Midwest Symposium on Circuits and Systems (lit. 8).

Mijn oprechte dank gaat uit naar mijn afstudeerhoogleraar Prof. dr. ing. H.J. Butterweck, naar mijn coach ir. S. Tirtoprodjo, met wie ik zeer vele gesprekken heb gehad, die de bron bleken van zeer vele vruchtbare ideeën, en naar alle medewerkers van de vakgroep ETB voor de gezellige tijd in hun midden.

2. Tweede Orde Sallen en Key Filter

2.1 Afleiding van de overdrachtsfunctie

Voor het tweede orde filter kunnen we als volgt de overdrachtsfunctie afleiden (zie ook fig. 1):

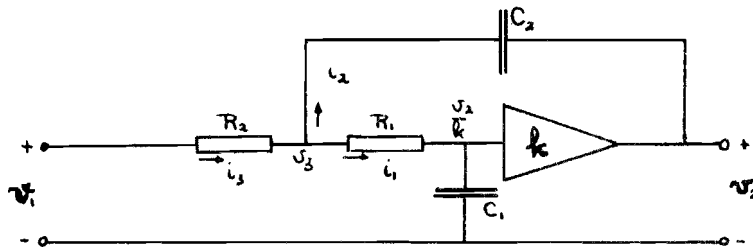


Fig. 1: tweede orde Sallen en Key laagdoorlaat filter.

$$i_1 = pC_1 \cdot \frac{U_2}{k} \quad (1)$$

$$U_3 = R_1 \cdot i_1 + \frac{U_2}{k} = (pC_1 \cdot R_1 + 1) \cdot \frac{U_2}{k} \quad (2)$$

$$i_2 = (U_3 - U_2) \cdot pC_2 \quad (3)$$

$$U_1 = (i_1 + i_2) \cdot R_2 + U_3 \quad (4)$$

$$= \left\{ p^2 C_1 R_1 C_2 R_2 + p(C_1 R_2 + [1 - k] C_2 R_2 + C_1 R_1) + 1 \right\} \cdot \frac{U_2}{k} \quad (5)$$

Voor de overdrachtsfunctie vinden we met (5):

$$\begin{aligned} H(p) = \frac{U_2}{U_1} &= \frac{k}{p^2 C_1 R_1 C_2 R_2 + p \{ C_1 R_2 + C_2 R_2 (1 - k) + C_1 R_1 \} + 1} \\ &= \frac{k}{a_2 p^2 + a_1 p + 1} \end{aligned} \quad (6)$$

2.2 Enige beschouwingen aan het filter

Indien we met dit filter een willekeurige ligging van de polen in het linkerhalfvlak willen realiseren zal het noemerpolynoom van (6) een Hurwitzpolynoom moeten zijn. De coëfficiënten moeten hiertoe aan de Hurwitzvoorwaarden voldoen welke voor een tweede orde polynoom luiden:

$$a_1 > 0 \quad \text{en} \quad a_2 > 0 \quad (7)$$

Verder zullen de R's en de C's positief reëel moeten zijn, opdat filter realiseerbaar is.

(6) volgt:

$$a_2 = R_1 C_1 R_2 C_2 \quad (8a)$$

$$a_1 = (R_1 + R_2) C_1 + (1-k) R_2 C_2 \quad (8b)$$

(8) zien we dat a_2 altijd positief reëel is voor positieve R's en C's. Om aan (7) te voldoen moeten we nog eisen:

$$a_1 = (R_1 + R_2) C_1 + (1-k) R_2 C_2 > 0 \quad (9)$$

$$\text{ofwel:} \quad k \leq \frac{R_1 C_1}{R_2 C_2} + \frac{C_1}{C_2} + 1 \quad (10)$$

We zien dat altijd aan (10) is voldaan door $k = 1$ te kiezen. Dat wil zeggen dat we voor $k = 1$ nooit polen in het rechterhalfvlak kunnen krijgen.

We kunnen nu nog nagaan of we met $k = 1$ alle polen in het linkerhalfvlak kunnen realiseren.

Uit (8) zien we dat geldt:

$$R_1 C_1 = \frac{a_2}{R_2 C_2} \quad (11)$$

Substitutie van (11) in de tweede vergelijking van (8) met $k = 1$ geeft:

$$a_2 - a_1 R_2 C_2 + R_2^2 C_1 = 0 \quad (12)$$

a_1 en a_2 willekeurig positief.

(12) vinden we als oplossing voor R_2 :

$$R_2 = \frac{a_1 C_2}{2 C_1 C_2} \pm \sqrt{\frac{a_1^2 C_2^2}{4 C_1^2 C_2^2} - \frac{a_2}{C_1 C_2}} \quad (13)$$

Hieruit zien we dat er altijd een positief reële oplossing voor R_2 te vinden is als geldt:

$$\frac{C_2}{C_1} \geq \frac{4 a_2}{a_1^2} \quad (14)$$

Omdat a_1 en a_2 positief reële constanten zijn en omdat aan C_1 en C_2 nog elke positief reële waarde kan worden toegekend, kunnen we C_1 en C_2 zo kiezen dat aan (14) wordt voldaan, waarmee we twee positief reële oplossingen voor R_2 vinden met vergelijking (13)

ook twee positief reële oplossingen voor R_1 met (11).

Dat dit geldt voor elke willekeurige positief reële a_1 en a_2 in alle poolliggingen in het linkerhalfvlak met de versterkingsfactor $k = 1$ realiseerbaar.

Uit vergelijking (8) zien we dat, indien we de weerstanden gelijk kiezen en normeren op $|\Omega|$, ook het gehele linkerhalfvlak realiseerbaar is met $k = 1$:

$$a_2 = C_1 C_2 \quad (15)$$

$$a_1 = 2C_1$$

dus:

$$C_1 = \frac{a_1}{2} \quad (F)$$

$$C_2 = \frac{a_2}{C_1} = \frac{2 a_2}{a_1} \quad (F) \quad (16)$$

tellen nu verder de condensatoren gelijk en normeren zo dat geldt:
 $C_2 = C = 1F$. Substitutie hiervan in (6) geeft voor de overdrachts-
 functie de volgende uitdrukking:

$$H(p) = \frac{k}{R_1 R_2 p^2 + \{R_1 + (2-k)R_2\}p + 1} = \frac{k}{a_2 p^2 + a_1 p + 1} \quad (17)$$

omdat we de polen realiseren met een negatief reëel deel moeten we
 de coëfficiënten van (17) weer voldoen aan de Hurwitzvoorwaarden (7).
 Omdat R_1 en R_2 positief reëel zijn moet dus gelden:

$$k \leq 2 + \frac{R_1}{R_2} \quad (18)$$

Voor $k = 2$ geldt nu met (17):

$$H(p) = \frac{k}{R_1 R_2 p^2 + R_1 p + 1} \quad (19)$$

Uit vergelijking (19) zien we nu dat bij gelijke condensatoren
 (1F) en voor de versterkingsfactor $k = 2$ alle willekeurige positief
 reële waarden voor de coëfficiënten a_1 en a_2 realiseerbaar zijn en
 hiermee elke willekeurige ligging van de polen in het linkerhalf-
 vlak:

$$R_1 = a_1 \quad (20)$$

$$R_2 = \frac{a_2}{R_1} = \frac{a_2}{a_1}$$

2.3 Grens van de realisatiemogelijkheden van het filter met gelijke capaciteiten als functie van de versterkingsfactor

Voor het berekenen van de grens van de realiseerbare polen in het complexe frequentievlak als functie van de versterkingsfactor zullen we twee gebieden onderscheiden, namelijk als eerste de reeks grenskrommen voor de versterkingsfactor $k \leq 2$ en vervolgens die voor $k > 2$.

We bekijken nu eerst de situatie voor een versterkingsfactor $k \leq 2$. Uit (17) zien we dat in dit geval zeker aan de Hurwitzvoorwaarden is voldaan zodat we altijd polen in het linkerhalfvlak hebben. De polen geldt met (17):

$$p_{1,2} = -\frac{a_1}{2a_2} \pm \frac{1}{2a_2} \sqrt{a_1^2 - 4a_2} \quad (21)$$

We zien nu dat we complex toegevoegde polen hebben als de uitdrukking onder het wortelteken van (21) kleiner dan nul is:

$$a_1^2 - 4a_2 \leq 0 \quad (22)$$

met:

$$\begin{aligned} a_2 &= R_1 R_2 \\ a_1 &= R_1 + (2-k)R_2 \end{aligned} \quad (23)$$

Stel de polen complex toegevoegd, dus:

$$p_{1,2} = x \pm jy \quad (24)$$

Met

$$\frac{(p-p_1)(p-p_2)}{x^2 + y^2} = \frac{1}{x^2 + y^2} p^2 - \frac{2x}{x^2 + y^2} p + 1 \quad (25)$$

vinden we:

$$a_2 = \frac{1}{x^2 + y^2} \quad (26)$$

$$a_1 = -\frac{2x}{x^2 + y^2}$$

Uit (23) zien we:

$$a_1 = \frac{a_2}{R_2} + (2-k)R_2 \quad (27)$$

ofwel:

$$k) R_2^2 - a_1 R_2 + a_2 = 0 \quad (28)$$

Oplossing voor R_2 vinden we hiermee:

$$R_2 = \frac{a_1 \pm \sqrt{a_1^2 - 4(2-k)a_2}}{2(2-k)} \quad (29)$$

$$R_1 = \frac{a_2}{R_2} \quad (30)$$

Opdat R_1 en R_2 realiseerbaar zijn zien we dat moet gelden:

$$a_1^2 \geq 4(2-k)a_2 \quad \text{en} \quad a_1 \geq 0 \quad (31)$$

De grens van de realiseerbare polen als functie van de versterkingsfactor vinden we door in vergelijking (31) de uitdrukkingen (26) te substitueren en het gelijkteken te nemen:

$$x^2 = (2-k)(x^2 + y^2) \quad (32)$$

$$x^2 = \frac{2-k}{k-1} y^2 \quad (33)$$

De vergelijkingen voor de grenskrommen zijn met (33):

$$x = \pm \sqrt{\frac{2-k}{k-1}} y \quad \text{en } x \leq 0 \quad (34)$$

Dit zijn rechten door de oorsprong (fig 2).

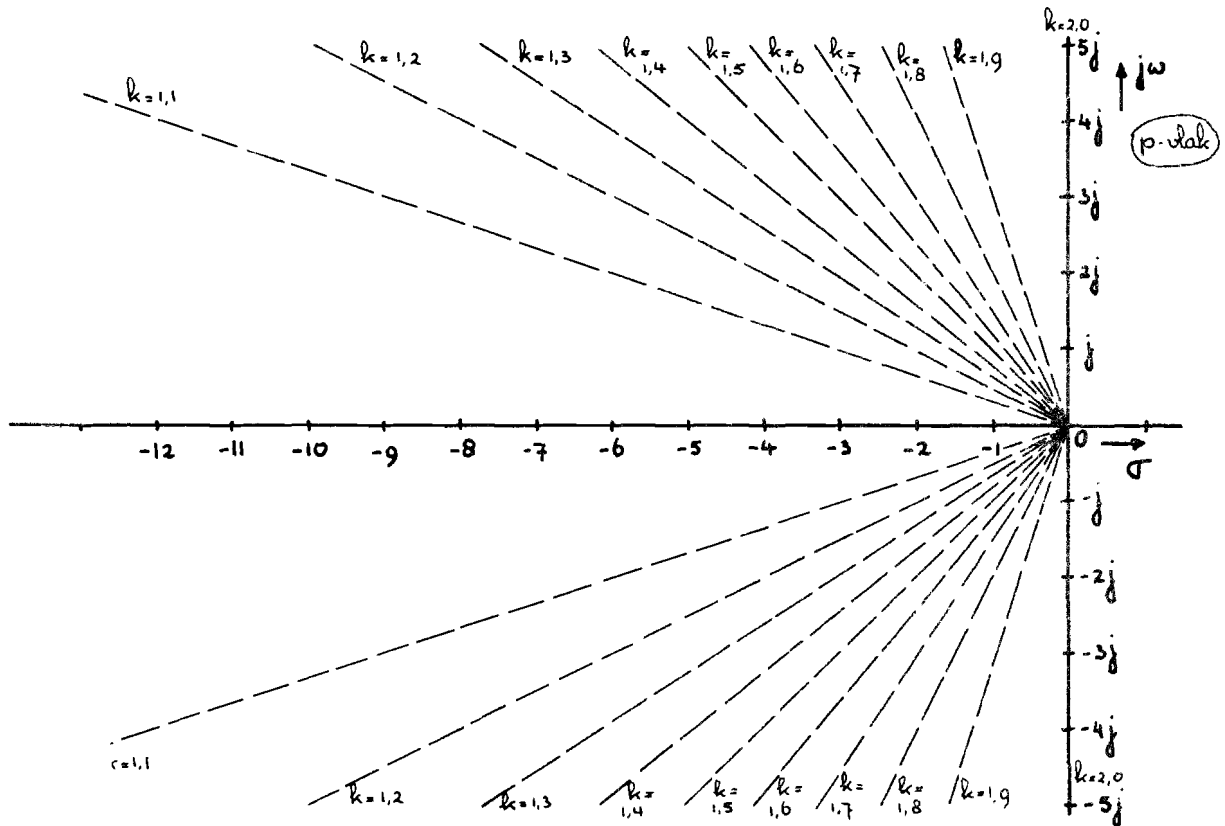


fig. 2: "grenskrommen" voor het tweede orde Sallen en Key Filter voor $1 \leq k \leq 2$

Voor de versterkingsfactor $k \geq 2$ gaan we op dezelfde wijze te werk. Uit de vergelijking (29) zien we dat slechts het min teken voldoet voor een positief reële waarde voor R_2 . Substitutie van de uitdrukkingen (26) in (29) en (30) geeft:

$$R_2 = \frac{-2x}{x^2 + y^2} \frac{-\sqrt{\frac{4x^2}{(x^2 + y^2)^2} + 4(k-2) \frac{1}{x^2 + y^2}}}{2(2-k)} \geq 0 \quad (35)$$

en

$$R_1 = \frac{\frac{2(2-k)}{x^2+y^2}}{-\frac{2x}{x^2+y^2} - \sqrt{\frac{4x^2}{(x^2+y^2)^2} + 4(k-2)\frac{1}{x^2+y^2}}} \geq 0 \quad (36)$$

ofwel

$$= \frac{x + \sqrt{(k-1)x^2 + (k-2)y^2}}{(k-2)(x^2+y^2)} \geq 0 \quad (37)$$

..

$$= \frac{k-2}{x + \sqrt{x^2 + (k-2)(x^2+y^2)}} \geq 0 \quad (38)$$

zien nu dat nu dat voor alle waarden van X en Y aan (37) en (38) is voldaan zodat alle poelligingen door het hele complexe frequentievlak met versterkingsfactoren $k \geq 2$ realiseerbaar zijn; er is dus nu geen sprake meer van een "grenskromme".

2.4 Nomogrammen voor Filters

We kunnen uit de overdrachtsfunctie voor het filter (formule(6)) rechtstreeks een nomogram berekenen, waarin de weerstandswaarden worden gegeven als functie van de versterkingsfactor voor elk gewenst filter, door de bij het filter behorende waarden voor de coëfficiënten a_1 en a_2 te substitueren.

We vinden op deze wijze:

$$= R_1 R_2 \quad (39a)$$

$$a_1 = R_2(2-k) + R_1 \quad (39b)$$

ruit volgt door R_2 te elimineren:

$$a_1 = (2-k) \frac{a_2}{R_1} + R_1 \quad (40)$$

ofwel:

$$R_1^2 - a_1 R_1 + (2-k) a_2 = 0$$

$$R_1 = \frac{a_1 \pm \sqrt{a_1^2 - 4a_2(2-k)}}{2} \quad (41)$$

We zien dat er voor $k < 2$ twee oplossingen mogelijk zijn mits is voldaan aan de volgende voorwaarde:

$$k \geq 2 - \frac{a_1^2}{4a_2} \quad (42)$$

Voor $k \geq 2$ is steeds één oplossing mogelijk.

Uit de vergelijkingen (39a), (41) en (42) kunnen we nu alle nomogrammen berekenen.

Voor het Butterworth filter verkrijgen we op deze wijze het volgende nomogram: (zie ook lit. 9)

$$a_2 = 1 \quad (43a)$$

$$a_{1,2} = \sqrt{2} \quad (43b)$$

met (41) en (39a):

$$R_1 = \frac{\sqrt{2} \pm \sqrt{4k-6}}{2} \quad (44)$$

$$R_2 = \frac{2}{\sqrt{2} \pm \sqrt{4k-6}} \quad (45)$$

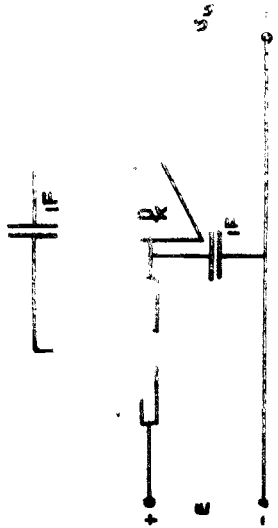
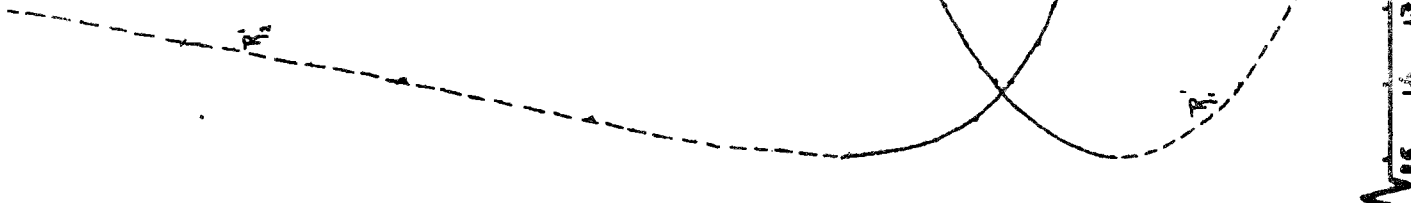
Er zijn twee oplossingen mogelijk voor (42):

$$1,5 \leq k < 2 \quad (46)$$

: nomogram ziet U afgebeeld in fig. 3.

impedans (ohm)

3,2
3,0
2,8
2,6
2,4
2,2
2,0
1,8
1,6
1,4
1,2
1,0
0,8
0,6
0,4
0,2
0



$$H(p) = \frac{k}{p^2 + \sqrt{2}p + 1}$$

R₁

fig. 3: nomogram voor het tweede orde Butterworth filter

R₂

3,2 3,0 2,8 2,6 2,4 2,2 2,0 1,8 1,6 1,4 1,2 1,0 0,8 0,6 0,4 0,2 0 0,2 0,4 0,6 0,8 1,0 1,2 1,4 1,6 1,8 2,0 2,2 2,4 2,6 2,8 3,0 3,2

3. Derde Orde Sallen en Key Filter

3.1 Afleiding van de overdrachtsfunctie

Het bepalen van de overdrachtsfunctie van het derde orde systeem gaat op dezelfde wijze als bij het tweede orde systeem. Het eerste gedeelte van de afleiding is precies het zelfde als de afleiding van paragraaf 2.1. We kunnen da afleiding dan ook beginnen met de formules (2-1), (2-3) en (2-5). (zie ook fig. 1)

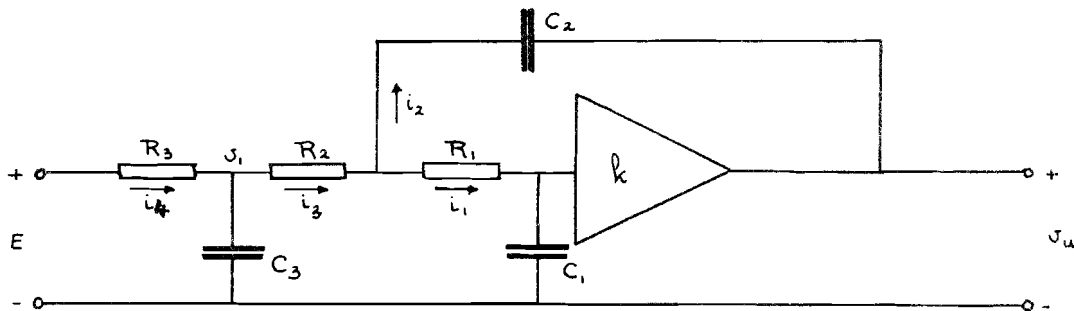


fig. 1: Derde orde Sallen en Key laagdoorlaat filter.

Met (2-1) en (2-3):

$$i_3 = \{ p^2 C_1 C_2 R_1 + p C_1 + (1-k) p C_2 \} \frac{v_u}{k} \quad (1)$$

Met (2-5):

$$v_1 = \{ p^2 C_1 R_1 C_2 R_2 + p [C_1 R_2 + C_1 R_1 + (1-k) C_2 R_2] + 1 \} \frac{v_u}{k} \quad (2)$$

$$i_4 = p C_3 v_1 + i_3 \quad (3)$$

$$\begin{aligned}
 E = & [p^3 C_1 R_1 C_2 R_2 C_3 R_3 + p^2 \{ C_1 R_2 C_3 R_3 + C_1 R_1 C_3 R_3 + (1-k) C_2 R_2 C_3 R_3 \\
 & + C_1 R_1 C_2 R_3 + C_1 R_1 C_2 R_2 \} + p \{ C_3 R_3 + C_1 R_3 + (1-k) C_2 R_3 \\
 & + C_1 R_2 + C_1 R_1 + (1-k) C_2 R_2 \} + 1 \quad (4)
 \end{aligned}$$

De overdrachtsfunctie luidt:

$$\begin{aligned}
 Y) = \frac{U_u}{E} &= \frac{k}{p^3 C_1 R_1 C_2 R_2 C_3 R_3 + p^2 \{ C_1 R_2 C_3 R_3 + C_1 R_1 C_3 R_3 + (1-k) C_2 R_2 C_3 R_3 \\
 & + C_1 R_2 C_3 R_3 + C_1 R_1 C_2 R_2 \} + p \{ C_3 R_3 + C_1 R_3 + \\
 & (1-k) C_2 R_3 + C_1 R_2 + C_1 R_1 + (1-k) C_2 R_2 \} + 1} \\
 &= \frac{k}{a_3 p^3 + a_2 p^2 + a_1 p + 1} \quad (5)
 \end{aligned}$$

3.2 Enige beschouwingen aan het filter

Indien we met dit derde orde filter slechts polen in het linkerhalfvlak willen realiseren dan zullen de coëfficiënten a_3 , a_2 en a_1 aan de Hurwitzvoorwaarden moeten voldoen. Voor een derde orde polynoom luiden deze voorwaarden:

$$a_1 \geq 0, \quad a_2 \geq 0, \quad a_3 \geq 0 \quad (6)$$

en

$$a_1 \cdot a_2 \geq a_3 \quad (7)$$

Met (5) en (6) moet gelden:

$$a_3 = C_1 R_1 C_2 R_2 C_3 R_3 \geq 0 \quad (8a)$$

$$a_2 = C_1 R_1 C_2 R_2 + C_1 R_1 C_3 R_3 + (1-k) C_2 R_2 C_3 R_3 + C_1 R_2 C_3 R_3 + C_1 R_1 C_2 R_3 \geq 0 \quad (8b)$$

$$a_1 = C_2 R_3 + C_1 R_3 + (1-k) C_2 R_3 + C_1 R_2 + C_1 R_1 + (1-k) C_2 R_2 \geq 0 \quad (8c)$$

we zullen zien dat aan deze voorwaarden altijd is voldaan, wanneer we de versterkingsfactor $k = 1$ kiezen. Aan de voorwaarden (6) is altijd voldaan met positieve R's en C's.

Substitutie van de coëfficiënten a_3 , a_2 en a_1 en $k = 1$ in (7) geeft:

$$\begin{aligned}
 a_1 \cdot a_2 &= \{ C_1 (R_1 + R_2 + R_3) + C_3 R_3 \} \{ C_1 C_3 R_3 (R_1 + R_2) + \\
 &\quad C_1 C_2 R_1 (R_2 + R_3) \} \\
 &= a_3 + C_1 C_2 C_3 R_1 R_3^2 + C_1^2 C_3 R_3 (R_1 + R_2 + R_3) (R_1 + R_2) \\
 &\quad + C_1^2 C_2 R_1 (R_1 + R_2 + R_3) (R_2 + R_3) \\
 &\quad + C_1 C_3^2 R_3^2 (R_1 + R_2) \tag{9}
 \end{aligned}$$

Uit (9) zien we dat voor de versterkingsfactor $k = 1$ ook altijd aan voorwaarde (7) is voldaan ; voor deze versterkingsfactor liggen de polen altijd in het linkervlak. We kunnen nu controleren of we ook het gehéle linkervlak kunnen realiseren met $k = 1$.

gelijking (8c) geeft:

$$R_3 = \frac{a_1 - C_1 (R_1 + R_2)}{C_1 + C_3} \tag{10}$$

Substitutie van (10) in (8a) en (8b) geeft:

$$a_3 = R_1 R_2 C_1 C_2 C_3 \left\{ \frac{a_1 - C_1 (R_1 + R_2)}{C_1 + C_3} \right\} \tag{11}$$

$$a_2 = \{ C_1 C_3 (R_1 + R_2) + C_1 C_2 R_1 \} \left\{ \frac{a_1 - C_1 (R_1 + R_2)}{C_1 + C_3} \right\} + C_1 C_2 R_1 R_2 \tag{12}$$

Vergelijking (11) geeft:

$$C_2 = \frac{a_2 (C_1 + C_3)}{R_1 R_2 C_1 C_3 \{a_1 - C_1 (R_1 + R_2)\}} \quad (13)$$

We zien uit vergelijking (10) en (13) dat we altijd positieve waarden voor R_3 en C_2 krijgen als geldt:

$$\geq C_1 (R_1 + R_2) \quad (14)$$

Aan deze voorwaarde (14) is voor $k = 1$ automatisch voldaan, immers uit (8c):

$$a_1 = C_1 (R_1 + R_2) + C_1 R_3 + C_3 R_3 \quad (15)$$

Substitutie van (13) in (12) geeft:

$$a_2 = C_1 C_3 (R_1 + R_2) \left\{ \frac{a_1 - C_1 (R_1 + R_2)}{C_1 + C_3} \right\} + C_1 R_1 \left\{ \frac{a_1 - C_1 (R_1 + R_2)}{C_1 + C_3} + R_2 \right\} \cdot \left\{ \frac{a_2 (C_1 + C_3)}{R_1 R_2 C_1 C_3 [a_1 - C_1 (R_1 + R_2)]} \right\} \quad (16)$$

Het is nu duidelijk dat we altijd positief reële waarden voor C_1 , R_1 , C_3 en R_2 kunnen vinden zodanig dat aan vergelijking (16) wordt voldaan. Met vergelijking (10) en (13) kunnen we dan vervolgens R_3 en C_2 nog bepalen.

Hieruit blijkt dat de versterkingsfactor $k = 1$ toereikend is om alle poelliggingen in het linkerhalfvlak te kunnen realiseren.

Een eerder gepubliceerd artikel (lit. 1) toont aan dat voor het filter met gelijke weerstanden de versterkingsfactor $k = 1$ niet meer toereikend is om het gehele linkerhalfvlak te realiseren; er is een verboden gebied langs de imaginaire as van het complexe frequentievlak.

Een ander artikel (lit. 2) laat zien dat voor het filter met gelijke capaciteiten de versterkingsfactor $k = 2$ volstaat om precies het gehele linkerhalfvlak te kunnen realiseren.

3.3 Zoeken naar een mogelijk rechtstreeks verband tussen pool-
ligging en weerstandswaarden van het filter met gelijke
capaciteiten

Als we de capaciteiten van het filter gelijk stellen en normeren op
1F dan luiden de coëfficiënten met vergelijking (5):

$$a_3 = R_1 R_2 R_3 \quad (17a)$$

$$a_2 = (2-k) R_2 R_3 + 2 R_1 R_3 + R_1 R_2 \quad (17b)$$

$$a_1 = (2-k)(R_2 + R_3) + R_1 + R_3 \quad (17c)$$

We leggen nu vervolgens één reële pool vast in -1 ; dit vastleggen
van deze ene pool vormt geen beperking omdat met een frequentie-
transformatie de pool nog naar iedere gewenste plaats op de reële
as kan worden geschoven.

Er moet nu gelden:

$$(p+1)(\alpha p^2 + \beta p + 1) = a_3 p^3 + a_2 p^2 + a_1 p + 1 \quad (18)$$

t (18) volgt:

$$\alpha = a_3 = R_1 R_2 R_3 \quad (19a)$$

$$\beta = a_2 - a_3 = (2-k) R_2 R_3 + 2 R_1 R_3 + R_1 R_2 - R_1 R_2 R_3 \quad (19b)$$

Verder moeten we eisen:

$$a_1 + a_3 = 1 + a_2 \quad (20)$$

De laatste eis (20) schrijven we nu uit door de vergelijkingen (17) te substitueren. Dit levert:

$$(2-k)(R_2+R_3)+R_1+R_3+R_1R_2R_3 = 1 + (2-k)R_2R_3 + 2R_1R_3 + R_1R_2 \quad (21)$$

aan nu proberen of we aan vergelijking (21) kunnen voldoen afhankelijk van één van de vier variabelen (R_1 , R_2 , R_3 en k), zodat we eigenlijk twee onafhankelijke variabelen hebben en twee afhankelijke variabelen; met de twee onafhankelijke variabelen kunnen we dan gaan kijken in welk gebied van het complexe frequentievlak we de resterende twee polen buiten $p_1 = -1$ kunnen krijgen. We zullen eerst bekijken of we aan vergelijking (21) kunnen voldoen onafhankelijk van de variabele R_1 :

R_1 vrij te kiezen:

Hiertoe herschrijven we vergelijking (21):

$$R_1 \{1 + R_2R_3 - 2R_3 - R_2\} = 1 + (2-k)(R_2R_3 - R_2 - R_3) - R_3. \quad (22)$$

We moeten nu twee eisen stellen:

$$\text{a) } 1 + R_2R_3 - 2R_3 - R_2 = 0 \quad (23)$$

$$b) (2-k)(R_2 R_3 - R_2 - R_3) + 1 - R_3 = 0 \quad (24)$$

Uit (23) volgt:

$$R_2 = \frac{2R_3 - 1}{R_3 - 1} \quad \Omega \quad (25)$$

Substitutie van (25) in (24) geeft:

$$\begin{aligned} R_3 &= \frac{1 + (k-2)R_2}{(3-k) + (k-2)R_2} \\ &= \frac{R_3(2k-3) - k + 1}{R_3(k-1) + 1} \end{aligned} \quad (26)$$

Verdere uitwerking van (26) levert het volgende resultaat:

$$R_3^2(k-1) - 2R_3(k-1) + (k-1) = 0$$

$$(R_3 - 1)^2(k-1) = 0$$

$$R_3 = 1 \quad \Omega \quad \text{of} \quad k = 1 \quad (27)$$

In paragraaf 3.4.1 zullen we zien dat met de versterkingsfactor $k=1$ slechts polen op de negatief reële as realiseerbaar zijn.

Substitutie van (27) in (25) levert:

$$R_2 = \infty \quad (28)$$

We zien dat het proberen te isoleren van R_1 leidt tot het oneindig worden van R_2 .

R_2 vrij te kiezen:

We herschrijven opnieuw vergelijking (21):

$$R_2 \{ 2-k + R_1 R_3 - (2-k) R_3 - R_1 \} = 1 + 2 R_1 R_3 - R_1 - (3-k) R_3 \quad (29)$$

Opdat deze vergelijking onafhankelijk van R_2 wordt moeten we nu eisen:

$$a) \quad 2-k + R_1 R_3 - (2-k) R_3 - R_1 = 0 \quad (30)$$

$$b) \quad 1 + 2 R_1 R_3 - R_1 - (3-k) R_3 = 0 \quad (31)$$

Uit (30) volgt:

$$R_1 = \frac{(2-k)(R_3 - 1)}{R_3 - 1} = 2-k \quad \Omega \quad (32)$$

(31) geeft:

$$\overline{R}_3 = \frac{1 - 2 + k}{1 + 2 - k - 2(2-k)} = 1 \quad (33)$$

Hieruit zien we dat we R_2 en k onafhankelijk kunnen kiezen mits :

$$k \leq 2 \quad (34)$$

Door aan vergelijking (32) en (33) te voldoen hebben we automatisch aan (21) voldaan en ligt dus één van de drie polen vast in $p_1 = -1$; de overige twee polen worden nu nog bepaald door R_2 en k .

Substitutie van (19), (32) en (33) in (18) geeft een uitdrukking waaruit we deze twee polen kunnen berekenen:

$$(p+1) \left\{ p^2 + \left(1 + \frac{2}{R_2}\right) p + \frac{1}{(2-k)R_2} \right\} = 0 \quad (35)$$

De discriminant van de vierkantsvergelijking tussen de accoladen kan kleiner dan nul zijn voor $k > 3/2$; dat wil zeggen dat we voor $3/2 \leq k \leq 2$ complex toegevoegde poolparen kunnen realiseren. We vragen ons nu nog af waar deze polen in het complexe frequentievlak gelegd kunnen worden. Om dit te onderzoeken bepalen we de oplossingen van de vierkantsvergelijkingen:

$$p^2 + \left(1 + \frac{2}{R_2}\right) p + \frac{1}{(2-k)R_2} = 0 \quad (36)$$

Stel de oplossingen:

$$P_{1,2} = x \pm jy$$

en

$$r = \sqrt{x^2 + y^2}$$

(afstand van een pool tot de oorsprong)

Uit (36) volgt:

$$x = -\frac{1}{2} - \frac{1}{R_2} \quad (37)$$

$$r^2 = \frac{1}{(2-k)R_2} \quad (38)$$

Met behulp van de vergelijkingen (37) en (38) kunnen we nu R_2 en k uitdrukken in de X -coördinaat van de polen en de afstand tussen de polen en de oorsprong:

$$R_2 = -\frac{2}{2x+1} \quad (39)$$

$$k = 2 + \frac{2x+1}{2r^2} \quad (40)$$

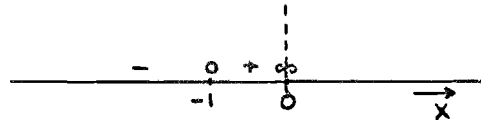
Vergelijking (39) laat ons zien dat we voor de X -coördinaat van de polen moeten eisen:

$$x \leq -\frac{1}{2} \quad (41)$$

Opdat we steeds een positieve waarde voor R_2 vinden. Met vergelijking (40) en (41) volgt nog:

$$k \geq 2 + \frac{2x+1}{2x^2} = f(x)$$

$$\frac{df}{dx} = -\frac{2x+2}{2x^3} = -\frac{x+1}{x^3} = 0$$



Voor $X = -1$ heeft $f(X)$ een minimum met $f(-1) = \frac{3}{2}$. (zie fig. 2)

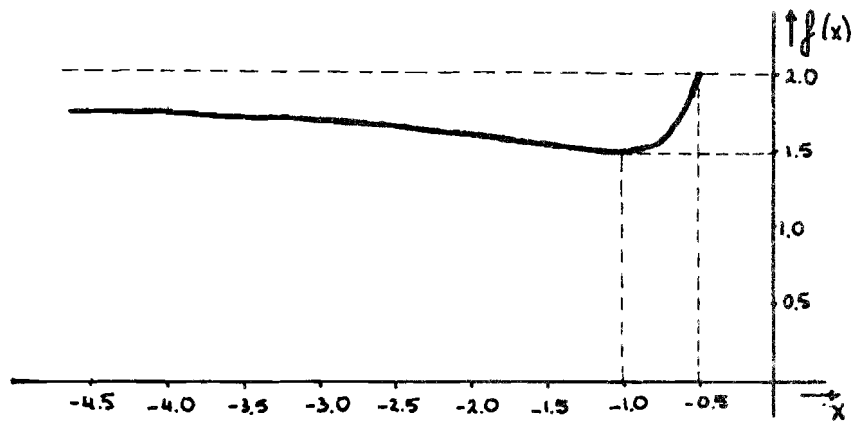


fig. 2 De ondergrens van k als functie van X

Als we vergelijking (40) nader bekijken dan zien we dat voor $X \leq -\frac{1}{2}$ altijd een bovengrens voor de versterkingsfactor k is aan te geven:

$$k \leq 2 \tag{42}$$

Uit de beschouwingen kunnen we concluderen dat alle poelligingen in het linkerhalfvlak realiseerbaar zijn mits aan voorwaarde (41) ($X \leq -\frac{1}{2}$) is voldaan.

We kunnen nu bovendien rechtstreeks uit de ligging van de polen de component waarden van het filter berekenen.

R_3 vrij te kiezen.

Afsplitsen van R_3 uit vergelijking (21) geeft:

$$R_3 \{ (2-k)(1-R_2) + 1 + R_1 R_2 - 2R_1 \} = 1 + R_1 R_2 - R_1 - (2-k)R_2 \quad (43)$$

Opdat (43) onafhankelijk van R_3 wordt moeten we eisen:

$$a. \quad (2-k)(1-R_2) + 1 + R_1 R_2 - 2R_1 = 0 \quad (44)$$

$$b. \quad 1 + 2R_1 R_2 - R_1 - (2-k)R_2 = 0 \quad (45)$$

Uit (44) volgt:

$$R_2 = \frac{2R_1 + k - 3}{R_1 + k - 2} \quad (46)$$

Na substitueren van (46) in (45) krijgen we:

$$R_1 = \frac{(2k-3)R_1 + (k-2)^2}{1-R_1} \quad (47)$$

Vergelijking (47) werken we nu uit:

$$R_1^2 + 2(k-2)R_1 + (k-2)^2 = 0$$

$$R_1 = (2-k)\alpha \quad (48)$$

Substitutie van (48) in (46) geeft:

$$R_2 = \infty \quad (49)$$

Het onafhankelijk maken van vergelijking (21) van R_3 leidt tot een onbruikbaar resultaat (49).

4. k vrij te kiezen.

Door vergelijking (21) te herschrijven krijgen we:

$$k \{ R_2 + R_3 - R_2 R_3 \} = R_1 R_2 R_3 + R_1 + 2R_2 + 3R_3 - 2R_2 R_3 - 1 - 2R_1 R_3 - R_1 R_2 \quad (50)$$

Teneinde deze vergelijking onafhankelijk van k te maken moeten we eisen:

a. $R_2 + R_3 - R_2 R_3 = 0 \quad (51)$

b. $R_1 R_2 R_3 + R_1 + 2R_2 + 3R_3 - 2R_2 R_3 - 1 - 2R_1 R_3 - R_1 R_2 = 0 \quad (52)$

Uit vergelijking (51) volgt:

$$R_3 = \frac{R_2}{R_2 - 1} \alpha \quad (53)$$

Als extra eis moeten we stellen:

$$R_2 \geq 1 \Omega \quad (54)$$

Substitutie van (51) in (52) levert:

$$R_1 (R_2 + R_3) + R_1 + 2R_2 + 3R_3 - 2R_2$$

$$- 2R_3 - 1 - 2R_1 R_3 - R_1 R_2 = 0$$

$$R_1 + R_3 - 1 - R_1 R_3 = 0$$

$$R_1 = 1 \Omega \quad (55)$$

Dit is weer een bruikbaar resultaat en we gaan dan nu ook onderzoeken welke poelligingen we op deze manier kunnen realiseren.

We substitueren (19), (53) en (55) in (18):

$$(p+1) \{ R_2 R_3 p^2 + [R_2 R_3 (2-k) + R_3] p + 1 \} = 0 \quad (56)$$

Stel de oplossingen van de vierkantsvergelijking:

$$p_{1,2} = x \pm jy \quad \text{en} \quad r = \sqrt{x^2 + y^2}$$

(afstand van de pool tot de oorsprong)

Met vergelijking (56) vinden we:

$$x = - \frac{2-k}{2} - \frac{1}{2R_2} \quad (57)$$

$$r^2 = \frac{1}{R_2 R_3} = \frac{R_2 - 1}{R_2^2} \quad (58)$$

(fig.3)

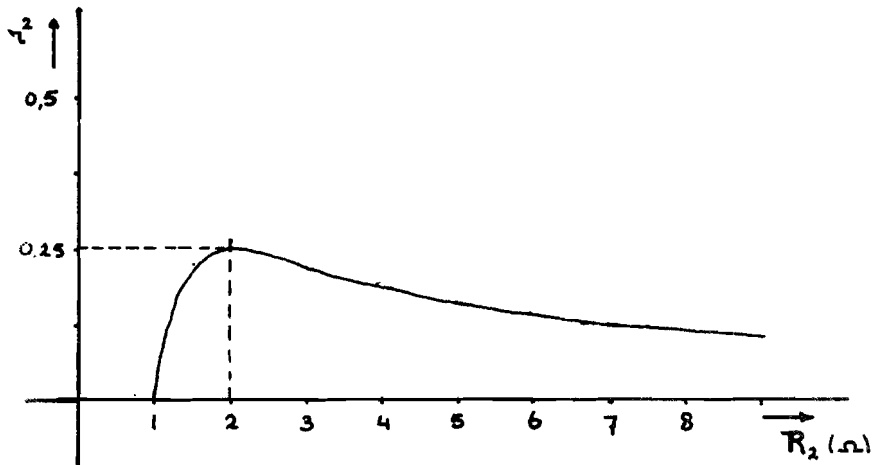


fig. 3 verloop van r^2 als functie van R_2

ofwel met (58) en (54)

$$R_2 = \frac{1}{2r^2} \left\{ 1 \pm \sqrt{1 - 4r^2} \right\} \quad (59)$$

We zien nu dat we alleen polen kunnen realiseren die binnen een cirkel liggen met straal (fig. 4)

$$r \leq \frac{1}{2} \quad (60)$$

De benodigde versterkingsfactor om een poolpaar te realiseren volgt door het herschrijven van (57):

$$k = 2x + z + \frac{1}{R_2} \quad (61)$$

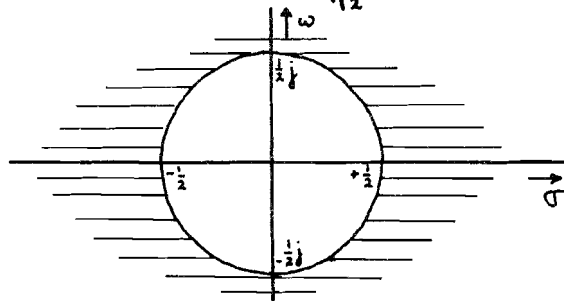


fig. 4 realiseerbare gebied

Uit vergelijking (57) en (58):

$$\begin{aligned}x^2 &= \frac{1}{R_2 R_3} \\&= \frac{1}{R_3} \{-2 + k - 2x\} \\x^2 + \frac{2}{R_3} x + y^2 &= \frac{k-2}{R_3} \\(x + \frac{1}{R_3})^2 + y^2 &= \frac{(k-2)R_3 + 1}{R_3^2} \quad (62)\end{aligned}$$

Hieruit zien we tot slot dat voor:

$$k \geq 2 - \frac{1}{R_3} = \frac{R_2 + 1}{R_2} \quad (63)$$

komplex toegevoegde poolparen mogelijk zijn.

Ofwel door substitutie van (61) in (63) voor:

$$x \geq -\frac{1}{2} \quad (64)$$

3.4 Realisatie mogelijkheden met het filter als functie van de versterkingsfactor

We hebben in paragraaf 3.2 reeds gezien, dat met een versterkingsfactor $k = 1$ en willekeurig te kiezen weerstands -en capaciteitswaarden omliggende in het gehele linkerhalfvlak van het frequentiedomein realiseerbaar zijn.

Een publicatie (lit. 2) is verder reeds aangetoond, dat bij het filter met gelijke capaciteiten een versterkingsfactor $k = 2$ toereikend is om het gehele linkerhalfvlak te bestrijken.

We willen nu onderzoeken welk gebied we kunnen realiseren als we de versterkingsfactor gaan variëren. Het blijkt niet mogelijk te zijn om analytisch rechtsreeks een uitdrukking af te leiden voor deze grenskromme als functie van de versterkingsfactor zoals we dat bij het tweede orde filter hebben gedaan. We zullen dan ook eerst een uiterste bovengrens voor het realiseerbare gebied afleiden en vervolgens met behulp van de computer de werkelijke grenzen met de versterkingsfactor als parameter gaan berekenen.

3.4.1 Afleiding van een uiterste bovengrens van het realiseerbare gebied als functie van de versterkingsfactor.

We gaan uit van de vergelijkingen (17a), (17b) en (17c). Voor versterkingsfactoren $k \leq 2$ kunnen we als volgt een schatting voor een bovengrens van het realiseerbare vlak afleiden:

$$\begin{aligned} a_1 a_2 = & \{ (2-k)R_2 R_3 + 2R_1 R_3 + R_1 R_2 \} \{ (2-k)(R_2 + R_3) + R_1 + R_3 \} \\ & + (2-k)^2 (R_2^2 R_3 + R_2 R_3^2) + 4 R_1 R_2 R_3 (2-k) + R_2 R_3^2 (2-k) + \\ & + 2 R_1 R_3^2 (2-k) + 2 R_1^2 R_3 + 2 R_1 R_3^2 + R_1 R_2^2 (2-k) + \end{aligned}$$

$$\begin{aligned}
 & + R_1^2 R_2 + R_1 R_2 R_3 \\
 = & a_3 (g-4k) + (2-k)^2 (R_2 + R_3) R_2 R_3 + (2-k) (2R_1 R_1^2 + R_1 R_2^2 + R_2 R_3^2) \\
 & + 2R_1^2 R_3 + 2R_1 R_3^2 + R_1^2 R_2 \quad (65)
 \end{aligned}$$

zien dat voor $k \leq 2$ geldt:

$$2 \cdot a_2 \geq a_3 (g-4k) \quad (66)$$

Stel nu:

$$\begin{aligned}
 a_3 p^3 + a_2 p^2 + a_1 p + 1 &= -\frac{1}{p_1 p_2 p_3} (p-p_1)(p-p_2)(p-p_3) \\
 &= -\frac{1}{p_1 p_2 p_3} p^3 + \frac{p_1 + p_2 + p_3}{p_1 p_2 p_3} p^2 - \frac{p_1 p_2 + p_1 p_3 + p_2 p_3}{p_1 p_2 p_3} p + 1 \quad (67)
 \end{aligned}$$

eruit volgt:

$$a_3 = -\frac{1}{p_1 p_2 p_3} \quad (68a)$$

$$a_2 = \frac{p_1 + p_2 + p_3}{p_1 p_2 p_3} \quad (68b)$$

$$a_1 = -\frac{p_1 p_2 + p_1 p_3 + p_2 p_3}{p_1 p_2 p_3} \quad (68c)$$

We stellen nu, omdat het polynoom zeker één reële wortel heeft:

$$p_1 = x, \quad (69a)$$

$$p_2 = x + jy \quad (69b)$$

$$p_3 = x - jy \quad (69c)$$

Substitutie van (69) in (68) geeft:

$$a_3 = - \frac{1}{x_1(x^2 + y^2)} \quad (70a)$$

$$a_2 = \frac{x_1 + 2x}{x_1(x^2 + y^2)} \quad (70b)$$

$$a_1 = - \frac{2x_1x + x^2 + y^2}{x_1(x^2 + y^2)} \quad (70c)$$

bovengrens vinden we door de vergelijkingen (70) in te vullen (66) en het gelijkteken te gebruiken:

$$- \frac{2x_1^2x + 5x_1x^2 + y^2(x_1 + 2x) + 2x^3}{x_1^2(x^2 + y^2)^2} = - \frac{g - 4k}{x_1(x^2 + y^2)} \quad (71)$$

Verder uitwerken van vergelijking (71) levert:

$$y^2 = - \frac{x \{ x^2 - 2(1-k)x + x^2 \}}{x - 2x_1(2-k)} \quad (72)$$

stellen we nu $x' = -\frac{x}{x_1}$ en substitueren we dit in (72) dan vinden we:

$$y^2 = - \frac{x' \{ x'^2 + 2(1-k)x' + 1 \}}{x' + 2(2-k)} x_1^2 \quad (73)$$

U ziet een paar krommen geschetst in fig. 5.

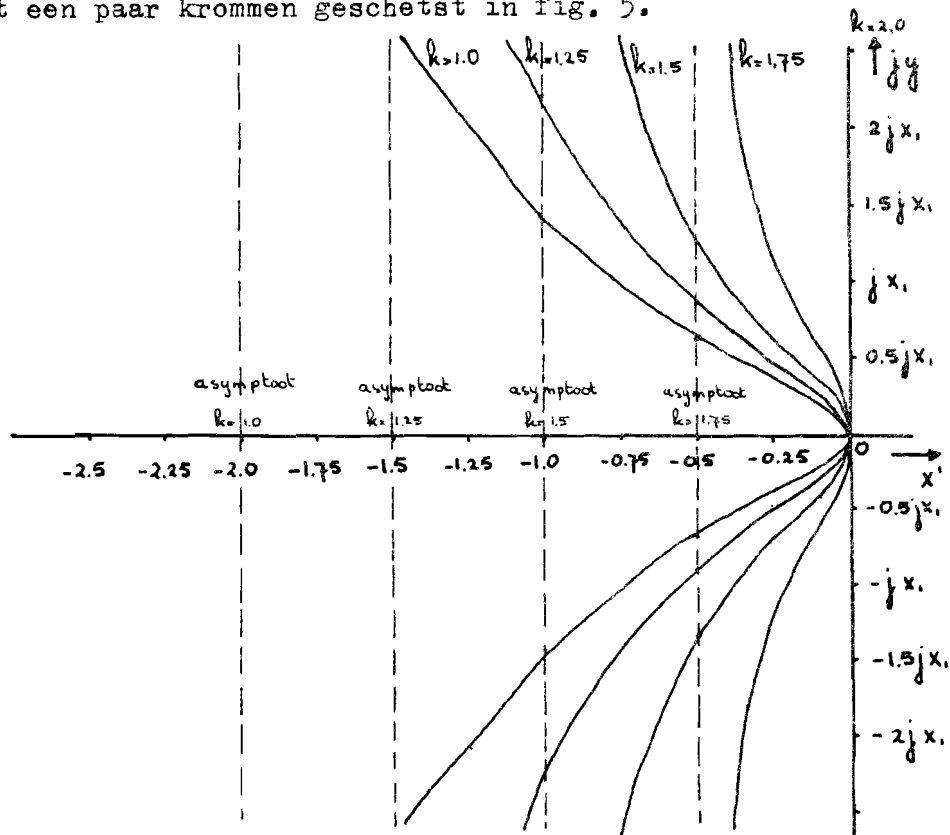


fig. 5 Enkele krommen van (73) voor een aantal waarden van de versterkingsfactor k .

Dat de hier afgeleide krommen niet de werkelijke grenskrommen zijn maar slechts bovengrenzen blijkt direkt uit het volgende bewijs dat voor $k = 1$ en gelijke capaciteiten de polen slechts op de negatief reële as kunnen liggen.

De geldt dan ($k = 1$):

$$a_3 = R_1 R_2 R_3 \quad (74a)$$

$$a_2 = R_2 R_3 + 2 R_1 R_3 + R_1 R_2 \quad (74b)$$

$$a_1 = R_1 + R_2 + 2 R_3 \quad (74c)$$

t zijn de coëfficiënten van het noemerpolynoom van de over-
achtsfunctie.

e polen volgen uit:

$$a_3 p^3 + a_2 p^2 + a_1 p + 1 = 0 \quad (75)$$

Delen we de reële pool uit dan krijgen we:

$$(R_1 p + 1) \{ R_2 R_3 p^2 + (2 R_3 + R_2) p + 1 \} = 0 \quad (76)$$

De twee overblijvende polen kunnen slechts op de negatieve reële as liggen. Dit volgt uit de discriminant van de tweede orde vergelijking:

$$D = (2 R_3 + R_2)^2 - 4 R_2 R_3$$

$$D = 4 R_3^2 + R_2^2 \geq 0 \quad (77)$$

3.4.2 Berekening van de grenskrommen met de computer

Voor de berekening van de echte grenskrommen gaan we weer uit van de overdrachtsfunctie zoals we die in paragraaf 3.1 hebben afgeleid (vergelijking (5)) met de coëfficiënten (17a), (17b) en (17c). De polen van het filter volgen uit de volgende vergelijking:

$$a_3 p^3 + a_2 p^2 + a_1 p + 1 = 0 \quad (78)$$

is

$$a_3 = \pi_1 \pi_2 \pi_3 \quad (17a)$$

$$a_2 = (2-k)\pi_2 \pi_3 + 2\pi_1 \pi_3 + \pi_1 \pi_2 \quad (17b)$$

$$a_1 = (2-k)(\pi_2 + \pi_3) + \pi_1 + \pi_3 \quad (17c)$$

We splitsen één pool, de reële pool, af en we leggen deze pool met behulp van een frequentietransformatie vast in $p = -1$.

Dit geeft:

$$(p+1)(\alpha p^2 + \beta p + 1) = a_3 p^3 + a_2 p^2 + a_1 p + 1 = 0 \quad (79)$$

Aan vergelijking (79) is te voldoen door het volgende te eisen:

$$a_1 + a_3 = 1 + a_2 \quad (80)$$

$$a_3 = \alpha \quad (81)$$

$$a_2 - a_3 = \beta \quad (82)$$

Herschrijven van deze drie vergelijkingen en substitutie van de vergelijkingen (17a), (17b) en (17c) leidt tot:

$$\alpha_1 = 1 + \beta = (2-k)(R_2 + R_3) + R_1 + R_3 \quad (83)$$

$$\alpha_2 = \alpha + \beta = (2-k)R_2 R_3 + 2R_1 R_3 + R_1 R_2 \quad (84)$$

$$\alpha_3 = \alpha = R_1 R_2 R_3 \quad (85)$$

Vergelijking (83) laat ons zien dat voor $k \leq 2$ altijd moet gelden:

$$0 \leq R_1 \leq 1 + \beta \quad (86)$$

Opdat we positieve, reële waarden voor R_2 en R_3 kunnen vinden. Uit vergelijking (85) volgt:

$$R_2 = \frac{\alpha}{R_1 R_3} \quad (87)$$

Substitueer (87) in (84):

$$(2-k) \frac{\alpha}{R_1} + 2R_1 R_3 + \frac{\alpha}{R_3} = \alpha + \beta \quad (88)$$

Substitutie van (87) in (83) levert:

$$(2-k) \left(R_3 + \frac{\alpha}{R_1 R_3} \right) + R_1 + R_3 = 1 + \beta \quad (89)$$

Als we deze laatste twee vergelijkingen vermenigvuldigen met $R_1 R_3$ vinden we:

$$2R_1^2 R_3^2 + \{(2-k)\alpha - (\alpha + \beta)R_1\} R_3 + \alpha R_1 = 0 \quad (90)$$

$$(3-k)R_1 R_3^2 + \{R_1^2 - (1 + \beta)R_1\} R_3 + (2-k)\alpha = 0 \quad (91)$$

Lossen we deze vierkantsvergelijkingen in R_3 op dan krijgen we:

$$R_{31} = \frac{(\alpha + \beta)R_1 - (2-k)\alpha \pm \sqrt{[(\alpha + \beta)R_1 - (2-k)\alpha]^2 - 8\alpha R_1^3}}{4R_1^2} \quad (92)$$

$$R_{32} = \frac{(\alpha + \beta)R_1 - R_1^2 \pm \sqrt{R_1^2 \{(\alpha + \beta - R_1)^2 - 4(2-k)(3-k)\alpha R_1\}}}{2(3-k)R_1} \quad (93)$$

We hebben met de vergelijkingen (92) en (93) vier uitdrukkingen gekregen waarin R_3 wordt uitgedrukt als functie van R_1 .

Wanneer we nu de weerstandswaarden willen bepalen die de polen van vergelijking (79) realiseren dan zullen we het snijpunt of de snijpunten van één van de vergelijkingen (92) met één van de vergelijkingen (93) moeten zoeken; de waarde van R_1 en de bijbehorende waarde van $R_{31} = R_{32} = R_3$ zijn dan de gezochte waarden voor de weerstanden R_1 en R_3 , de waarde van de weerstand R_2 kunnen we nu berekenen uit vergelijking (37).

Op deze wijze kunnen we berekenen wat voor alle coëfficiënten α en β de bijbehorende weerstandswaarden R_1 , R_2 en R_3 zijn, mits deze oplossing bestaat voor de gebruikte versterkingsfactor k . Zoals we zien kunnen we de versterkingsfactor nog vrij kiezen.

Wat we nu willen gaan doen is bij een aantal versterkingsfactoren berekenen welke coëfficiënten α en β , dus welke poelliggingen in het complexe frequentievlak, nog positief reële waarden voor de weerstanden R_1 , R_2 en R_3 opleveren. De krommen die de grens van de realiseerbare poelliggingen aangeven, de grenskrommen, gaan we met behulp van de rekenmachine door gebruik te maken van bovenstaande vergelijkingen (92) en (93) bepalen.

We gaan hiertoe het complexe frequentievlak systematisch aftasten. Dit aftasten doen we met behulp van concentrische cirkels om de oorsprong (zie fig. 6).

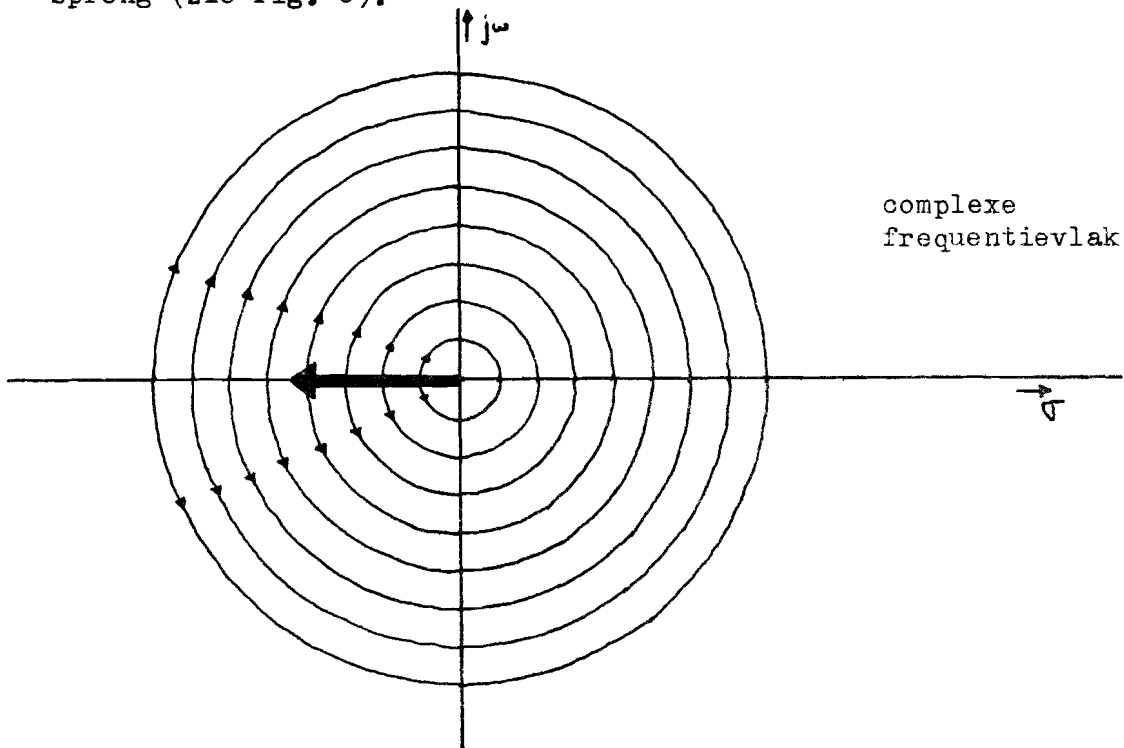


fig. 6 De wijze waarop we het complexe frequentievlak aftasten om de grenskrommen te bepalen

We gaan -beginnend met een heel kleine cirkel om de oorsprong- op deze cirkel twee complex toegevoegde polen proberen te realiseren te beginnen op de negatief reële as (we hebben één van de drie polen vastgelegd in $p = -1$) door snijpunten te zoeken van een vergelijking (92) met een vergelijking (93); dit doen we net zolang totdat we een punt op de cirkel bereiken dat niet meer realiseerbaar is. Dit punt is dan, onder voorwaarde dat de stappen op de cirkel oneindig klein worden gekozen, een punt van de grenskromme. Het is mogelijk dat we op de hele cirkel realiseerbare poolliggingen vinden; dit betekent dan dat de hele cirkel in het realiseerbare gebied ligt. Hebben we een cirkel helemaal op deze wijze afgetast, dan gaan we over op een iets grotere cirkel en we beginnen deze weer op dezelfde wijze te onderzoeken.

Op deze manier kunnen we punt voor punt een grenskromme bepalen. Uit vergelijking (79) zien we dat de straal van de cirkel, waarop de polen liggen die we willen realiseren, wordt bepaald door de coëfficiënt α .

Door de coëfficiënt α te veranderen, wijzigen we de straal van de cirkel. De plaats op de cirkel wordt bepaald door de coëfficiënt β ; veranderen we β bij constant houden van α dan wil dit ook zeggen dat we de polen over de cirkel met straal $r = \frac{1}{\sqrt{\alpha}}$ verschuiven. Met vergelijking (79) zien we dat, wanneer we dubbele reële polen op de negatief reële as willen realiseren, voor β de waarde

$$\beta = 2\sqrt{\alpha} \quad (94)$$

moet worden gekozen. Als we β vanaf $2\sqrt{\alpha}$ af laten nemen tot de waarde $-2\sqrt{\alpha}$ betekent dit in het frequentievlak een verschuiven van de polen over een cirkel met straal $\frac{1}{\sqrt{\alpha}}$ van een dubbele negatief reële pool naar een dubbele positief reële pool (fig. 7)

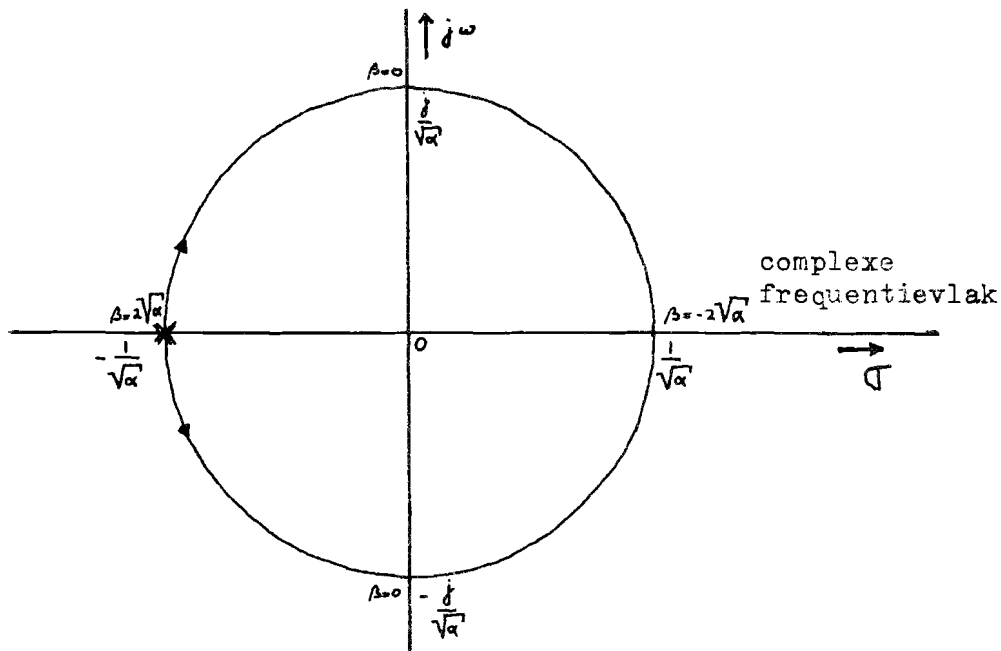


fig. 7 beschreven poolbaan bij variëren van β en constant houden van α .

Met het bovenstaande hebben we de mogelijkheid om numeriek de grenskrommen te berekenen. De as waar alles om draait is het zoeken van punten R_1 waar een vergelijking (92)

$$R_{31a} = f(R_1) \quad (95a)$$

$$R_{31b} = g(R_1) \quad (95b)$$

elijk is aan een vergelijking (93):

$$R_{32a} = F(R_1) \quad (96a)$$

$$R_{32b} = G(R_1) \quad (96b)$$

Zoals we uit (95) en (96) direct inzien zijn er vier mogelijkheden om zulke punten te vinden:

$$f(R_1) = F(R_1) \quad (97a)$$

$$f(R_1) = G(R_1) \quad (97b)$$

$$g(R_1) = F(R_1) \quad (97c)$$

$$g(R_1) = G(R_1) \quad (97d)$$

We gaan het zoeken naar punten R_1 , die aan vergelijking (97) voldoen als volgt numeriek aanpakken;

We kunnen achtereenvolgens zoeken naar oplossingen van (97) (a), (b), (c) en tot slot (97d); we kennen aan R_1 een beginwaarde toe en we rekenen de functiewaarden in dat punt uit en tevens het verschil tussen deze functiewaarden; vervolgens verhogen we R_1 met een bepaalde waarde en rekenen opnieuw de functiewaarden en het verschil tussen de functiewaarden uit; we onderzoeken nu of

het verschil van de functiewaarden van teken gewisseld is; als dit zo is dan betekent het dat er een oplossing tussen de twee punten R_1 , waarvoor de functiewaarden zijn uitgerekend, ligt en als er geen tekenwisseling is opgetreden dan ligt er ook geen oplossing tussen de twee punten. We verhogen zo telkens R_1 met met een bepaalde waarde

ΔR_1 (ΔR_1 is zo klein dat er in dit kleine interval maar één tekenwisseling op kan treden!!), rekenen de functiewaarden en hun verschil uit en we onderzoeken of er tussen de twee beschouwde punten een tekenwisseling is opgetreden.

Om dit probleem met de computer op te kunnen lossen moeten we natuurlijk een bovengrens zien te vinden voor de waarde van R_1 , waarboven zeker geen oplossingen meer liggen, opdat de rekenmachine niet eeuwig door moet blijven gaan met het zoeken naar tekenwisselingen om alle oplossingen te vinden. Zoals we in vergelijking (86) hebben gezien is deze bovengrens voor versterkingsfactoren $k \leq 2$ eenvoudig aan te geven:

$$0 \leq R_1 \leq 1 + \beta \quad (86)$$

We zullen zien dat we ook voor versterkingsfactoren $k > 2$ zo'n grens kunnen aangeven.

andere beschouwing van de functie onder het wortelteken van vergelijking (92)

$$h(R_1) = \{(\alpha + \beta)R_1 - (2-k)\alpha\}^2 - 8\alpha R_1^3 \quad (98)$$

leert ons dat deze functie vanaf een bepaalde waarde voor R_1 zeker negatief wordt als α positief is, en dat is altijd het geval.

Een grenswaarde voor R_1 , boven welke we zeker geen bruikbare oplossingen zullen vinden, kunnen we hieruit als volgt afleiden:

$$(\alpha + \beta)^2 R_1^2 - 2(\alpha + \beta)(2-k)\alpha R_1 + (2-k)^2 \alpha^2 \leq 8\alpha R_1^3$$

We onderscheiden nu twee gevallen ($b \geq 0$ en $b \leq 0$) waarvoor we aan eis (100) gaan voldoen:

1. $b \geq 0$

In dit geval kunnen we schrijven:

$$(x+b)^3 - b^3 = x^3 + 3bx^2 + 3b^2x \geq x(x+b)^2 = x^3 + 2bx^2 + b^2x$$

voor $X \geq 0$ en $b \geq 0$ (101)

Wanneer we nu zorgen dat we altijd voldoen aan

$$(x+b)^3 - b^3 \leq a$$

dan is zeker altijd aan de gestelde eis voldaan. De waarde voor X die hieruit volgt is:

$$x \leq \sqrt[3]{a + b^3} - b \quad (102)$$

2. $b \leq 0$

Dit geval splitsen we in twee gedeelten, namelijk de situatie dat geldt:

$$-\frac{4}{27} b^3 \geq a \quad (103)$$

en als tweede:

$$-\frac{4}{27} b^3 < a \quad (104)$$

$$a. \quad -\frac{4}{27} b^3 \geq a$$

We trekken nu een raaklijn aan $f(X)$ in de oorsprong (zie ook fig. 9) en we geven nu als schatting voor de waarde van X tot welke we rekenen, die waarde waarvoor deze raaklijn de waarde a bereikt. Voor de richtingscoëfficiënt van $f_1(X)$ in de oorsprong vinden we:

$$f_1'(x) = 3x^2 + 4bx + b^2$$

$$f_1'(0) = b^2 \quad (105)$$

De vergelijking voor de raaklijn en het gezochte punt X volgt hier als volgt uit:

$$y = b^2 x = a$$

$$x = \frac{a}{b^2} \quad (106)$$

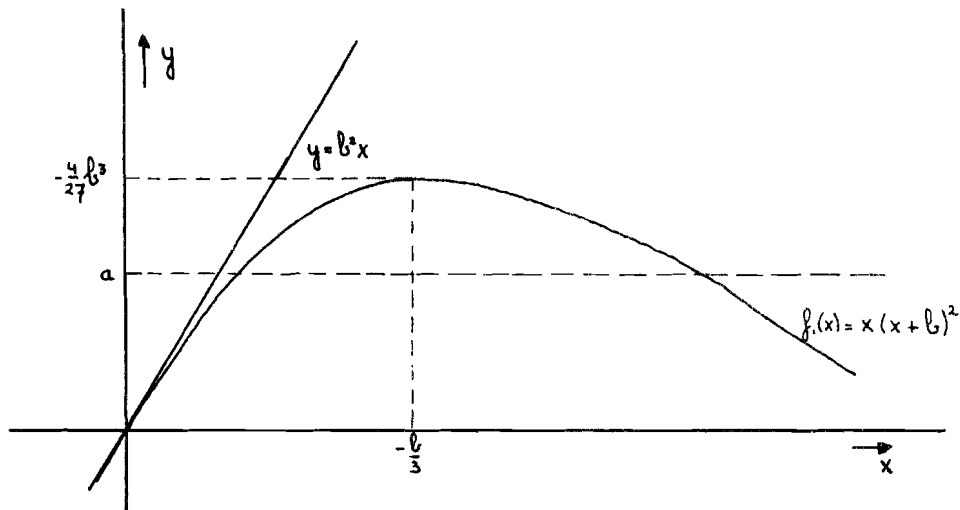


fig. 9 Schets van de gebruikte schatting voor $-\frac{4}{27} b^3 \geq a$

$$b. \quad -\frac{4}{27} b^3 < a$$

We voeren een nieuwe variabele in met:

$$y = x + b \quad (107)$$

Substitutie van (107) in (100) geeft:

$$(y-b) y^2 \leq a \quad (108)$$

We moeten nu zoeken naar een maximale waarde voor Y zodat voor het interval $b \leq Y \leq Y_{\max}$ altijd aan de vergelijking (108) is voldaan (met $b \leq 0$!).

Uit fig. 8b en uit vergelijking (104) zien we, dat we voor de waarde X_{\max} een waarde kunnen kiezen waarvoor geldt:

$$X_{\max} \geq -b$$

zodat we voor Y alleen de waarden

$$Y \geq 0$$

hoeven te beschouwen.

We kunnen nu dan ook voor dit gebied schrijven:

$$(y-b)^3 + b^3 \geq y^3 - by^2 \quad (109)$$

Wanneer we nu een Y_{\max} zodanig kiezen dat steeds geldt:

$$(y-b)^3 + b^3 \leq a \quad (110)$$

dan is zeker altijd aan vergelijking (108) voldaan.

Uit (110) volgt:

$$y \leq \sqrt[3]{a - b^3} + b \quad (111)$$

Met (107) vinden we hieruit voor X:

$$x \leq \sqrt[3]{a - b^3} \quad (112)$$

Met de vergelijkingen (102), (106) en (112) hebben we nu grenswaarden afgeleid waarvoor in de diverse gevallen steeds aan voorwaarde (100) voldaan.

Voor in deze vergelijkingen de coëfficiënten a en b en de variabele X (100a) te substitueren vinden we de grenswaarden voor R_1 waarboven er geen bruikbare oplossingen bestaan voor het filter en tevens de omstandigheden waarvoor we een van de drie grenswaarden moeten gebruiken.

$$1. \quad b \geq 0 \implies \frac{\alpha + \beta}{(k-2)\alpha} \geq 0$$

De grenswaarde voor R_1 vinden we uit (102):

$$\begin{aligned} R_{1, \max} &= \frac{1}{\sqrt[3]{a + b^3} - b} \\ &= \frac{(k-2)\alpha}{\sqrt[3]{8(k-2)\alpha^2 + (\alpha + \beta)^2} - (\alpha + \beta)} \end{aligned} \quad (113)$$

$$2. \quad b \leq 0 \implies \frac{\alpha + \beta}{(k-2)\alpha} \leq 0$$

De twee gevallen die we hebben onderscheiden worden nu:

$$a. \quad -\frac{4}{27} b^3 \geq a$$

$$\text{ofwel} \quad -\frac{4}{27} \frac{(\alpha+\beta)^3}{(k-2)^3 \alpha^3} \geq \frac{8}{(2-k)^2}$$

$$\frac{(\alpha+\beta)^3}{(k-2) \alpha^2} \leq -54 \quad (114)$$

De bijbehorende maximale waarde voor R_1 vinden we uit (106):

$$R_1 \max = \frac{(\alpha+\beta)^2}{8\alpha} \quad (115)$$

$$b. \quad \frac{(\alpha+\beta)^3}{(k-2) \alpha^2} > -54$$

De gezochte maximale waarde wordt nu (112):

$$R_1 \max = \frac{(k-2) \alpha}{\sqrt[3]{\frac{8}{(2-k)^2 \alpha} - \frac{(\alpha+\beta)^3}{(k-2)^3 \alpha^3}}} \quad (116)$$

We hebben nu bijna het wiskundige gereedschap klaar om het probleem met de computer aan te pakken. Het enige waar we nog voor moeten zorgen is dat de functies (92) en (93) in de te doorlopen trajecten voor R_1 en k geen singuliere punten bevatten.

Als we de vergelijkingen nader beschouwen zien we direct dat er twee singuliere punten zijn:

$$k = 3 \quad \text{en} \quad R_1 = 0$$

zien direct dat voor de versterkingsfactor $k = 3$ geen oplossingen mogelijk zijn. Omdat we de versterkingsfactor als een parameter gaan voeren hoeven we hier niets aan te doen. Teneinde het singuliere punt $R_1 = 0$ kwijt te raken vermenigvuldigen we de beide vergelijkingen (92) en (93) met R_1^2 :

$$f_{1,2}(R_1, \alpha, \beta, k) = R_1^2 \cdot R_{31} = \frac{1}{4} \{ (\alpha + \beta) R_1 - (2-k)\alpha \}$$

$$\pm \sqrt{\frac{1}{16} [\{ (\alpha + \beta) R_1 - (2-k)\alpha \}^2 - 8\alpha R_1^3]} \quad (117)$$

en

$$g_{1,2}(R_1, \alpha, \beta, k) = R_1^2 \cdot R_{32} = \frac{R_1^2}{2(3-k)} \{ 1 + \beta - R_1 \}$$

$$\pm \sqrt{\frac{R_1^3}{4(3-k)^2} [R_1 (1 + \beta - R_1)^2 - 4(2-k)(3-k)\alpha]} \quad (118)$$

Nu kunnen we schematisch aangeven hoe het programma, dat de oplossingen R_1 en R_3 moet gaan bepalen, eruit ziet. Dit schema ziet U afgebeeld in fig. 10.

Het werkelijke programmadeel dat deze oplossingen berekent (procedure RIBEP; zie ook bijlage 1) bevat nog enige extra stappen die in het schema niet zijn aangegeven. Deze extra stappen dienen voornamelijk om de benodigde rekentijd te verkorten; tijdens het doorrekenen van de functies de eerste keer (plus - tekens voor de wortels, formules (82) en (93)) wordt opgeslagen, waar de argumenten van de wortels negatief worden, zodat dit gebied bij de volgende drie stappen (plus - min teken, min - plus teken en tot slot twee min - tekens) wordt overgeslagen.

De behulp van deze procedure kan worden nagegaan of er voor de opgegeven poolligging een realisatie mogelijk is en indien dit zo is worden de waarden van de componenten berekend. De ligging van de polen moet in de coëfficiënten α en β worden opgegeven en verder moet de versterkingsfactor k als parameter worden ingevoerd. Als resultaat van de procedure krijgen we de waarden van de weerstanden R_1 en R_3 die bij de opgegeven poolligging horen. Uit vergelijking (87) kunnen we hiermee de bijbehorende waarde van R_2 berekenen.

Teneinde de krommen te berekenen die voor een bepaalde versterkingsfactor de grenzen aangeven van het deelgebied van het complexe frequentievlak waar poolliggingen realiseerbaar zijn gaan we te werk als aangegeven in fig. 6 en fig. 7. We kiezen als beginwaarde voor β (in het programma mu genaamd) de waarde $\beta = 2\sqrt{\alpha}$ (de variabele α wordt in het programma eps genaamd); in dit punt (een dubbele reële pool op de negatieve reële as) wordt met RIBEP een oplossing bepaald, mits deze bestaat; vervolgens wordt β met de factor Δ/β verlaagd en wordt weer een oplossing gezocht. Indien er achtereenvolgens een punt wordt gevonden waar wel een (respectievelijk geen) oplossing mogelijk is en vervolgens een punt waar geen (respectievelijk wel een) oplossing mogelijk is, wordt met intervalhalvering de waarde van β bepaald, waar de grens ligt.

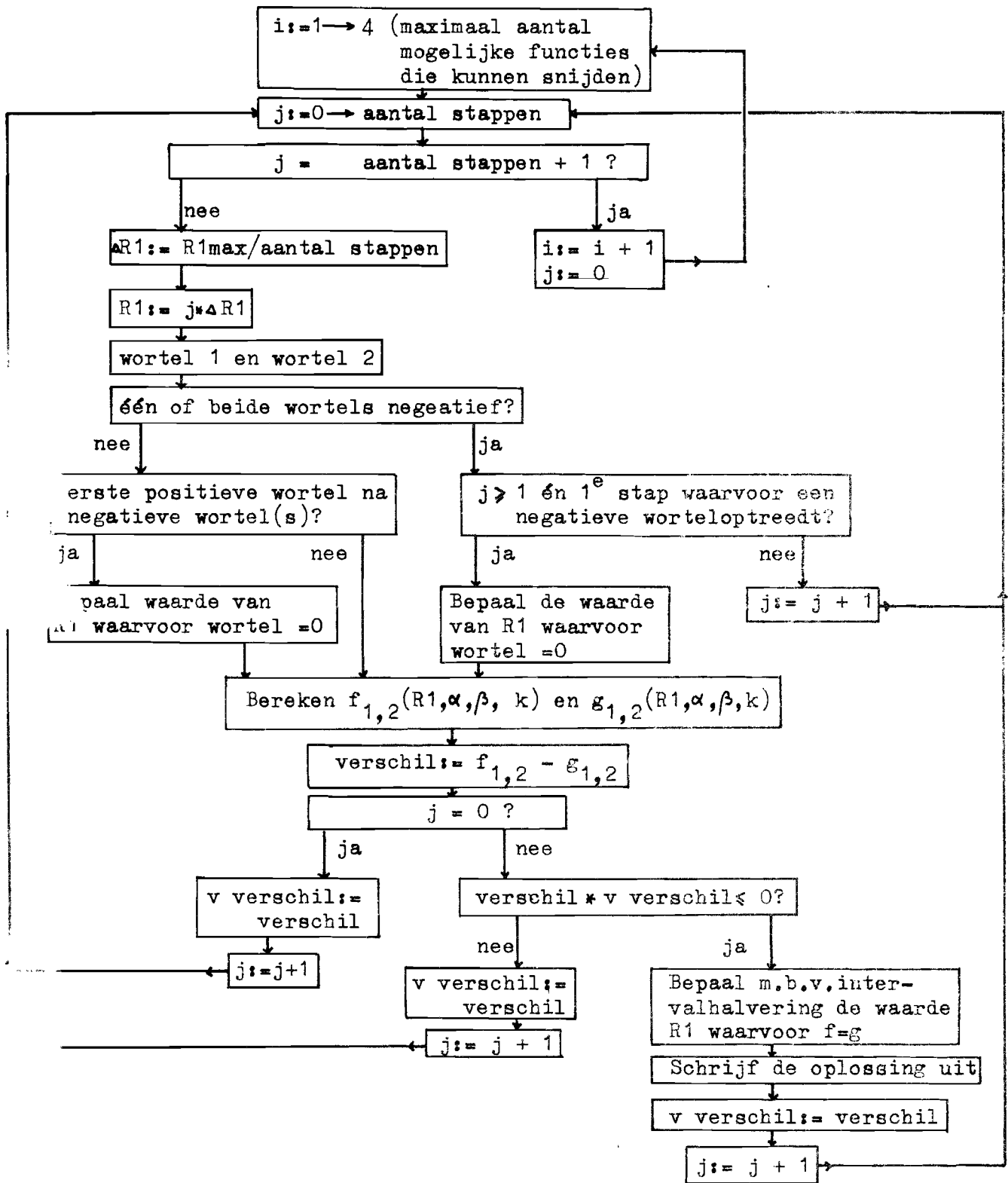


fig. 10: Schema dat globaal de werking van de procedure RIBEP aangeeft.

Met verlagen van β met stapjes $\Delta\beta$ doen we totdat β de waarde $\beta_c = 2\sqrt{a}$ heeft bereikt. In fig. 11 ziet U schematisch aangegeven hoe deze procedure er uit ziet.

De twee hierboven beschreven procedures vormen de kern van het programma dat de "grenskrommen" bepaalt. Dit programma, waarvan de volledige tekst in bijlage 1 is afgedrukt, levert ons de werkelijke grenskrommen voor de diverse versterkingsfactoren, zoals U ze ziet afgebeeld in fig. 12 ($1 \leq k \leq 2$), fig 13 ($2 \leq k \leq 3$) en fig 14 ($k > 3$).

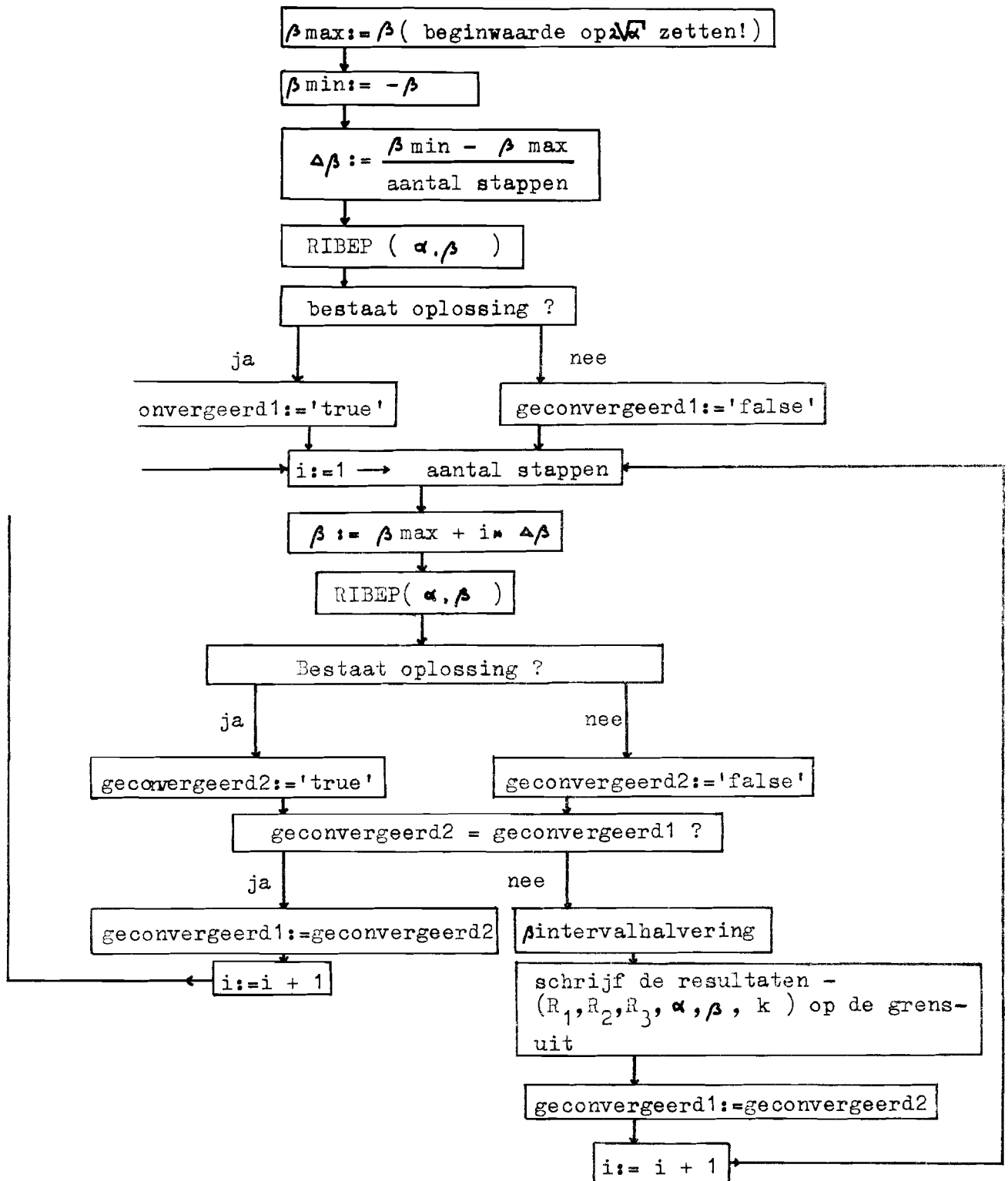
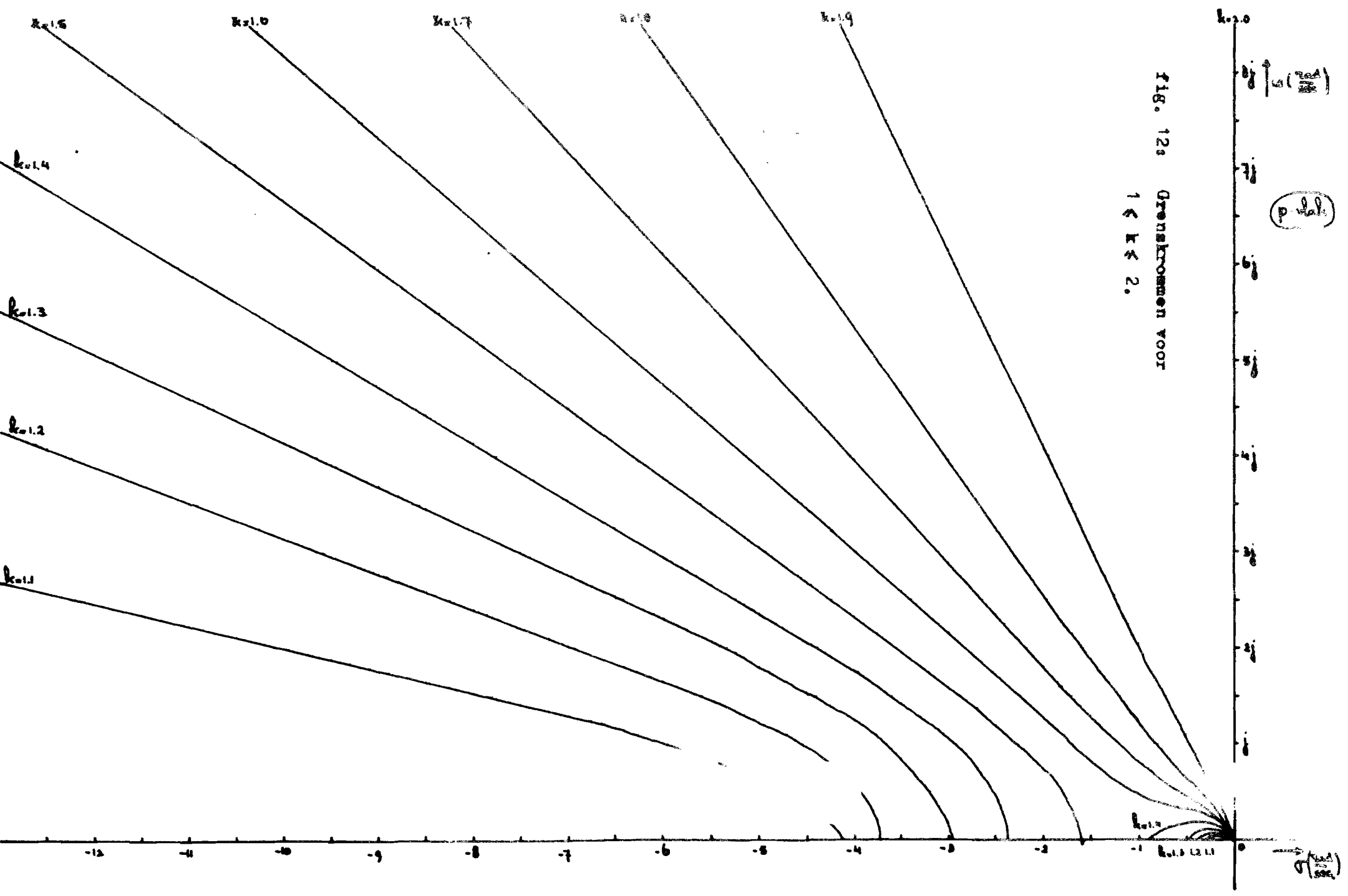


fig. 11: Schematisch overzicht van de procedure MUSTAP

fig. 12a
Dreankrommen voor
1 & k < 2.



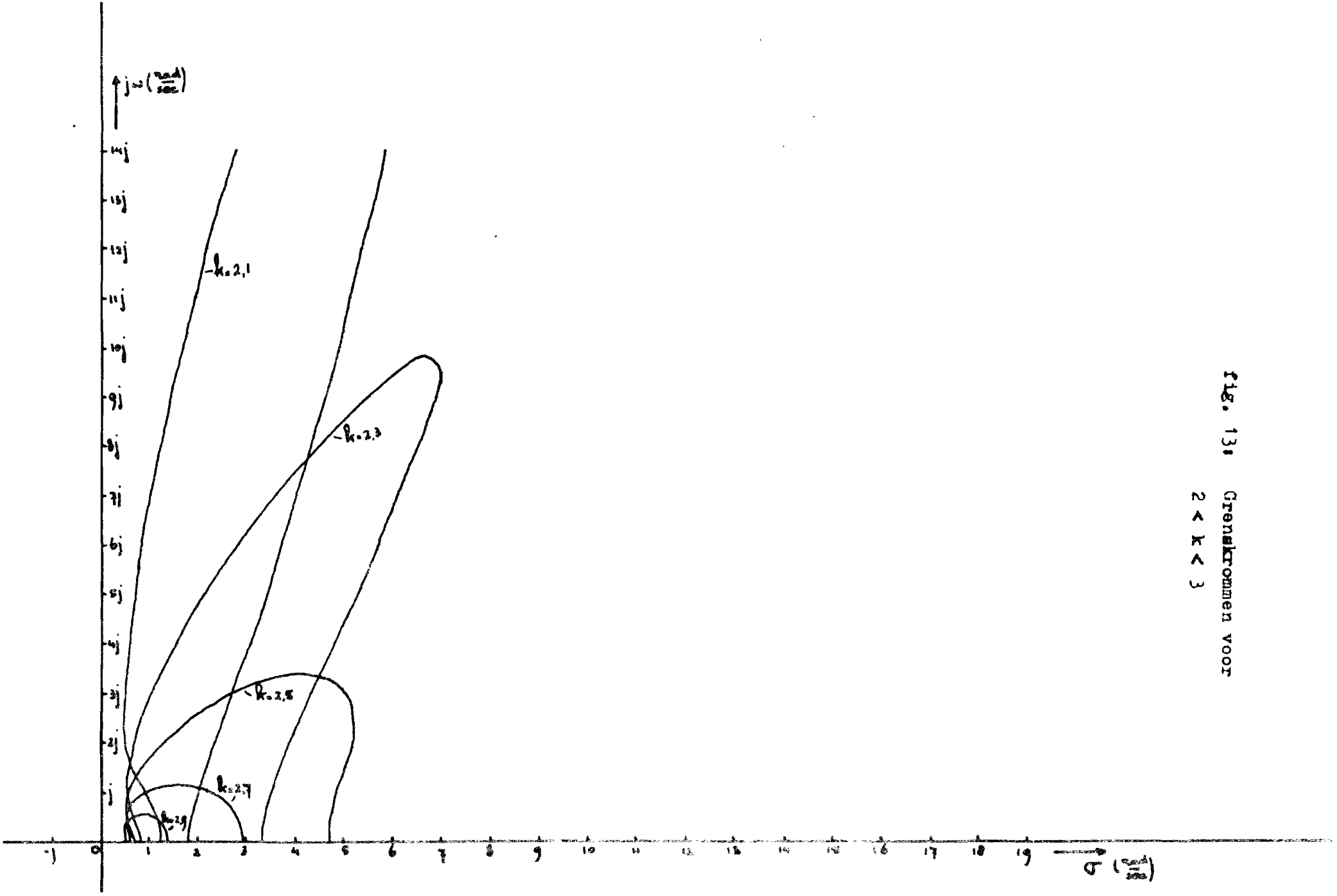


fig. 13: Grenskrommen voor
 $2 < k < 3$

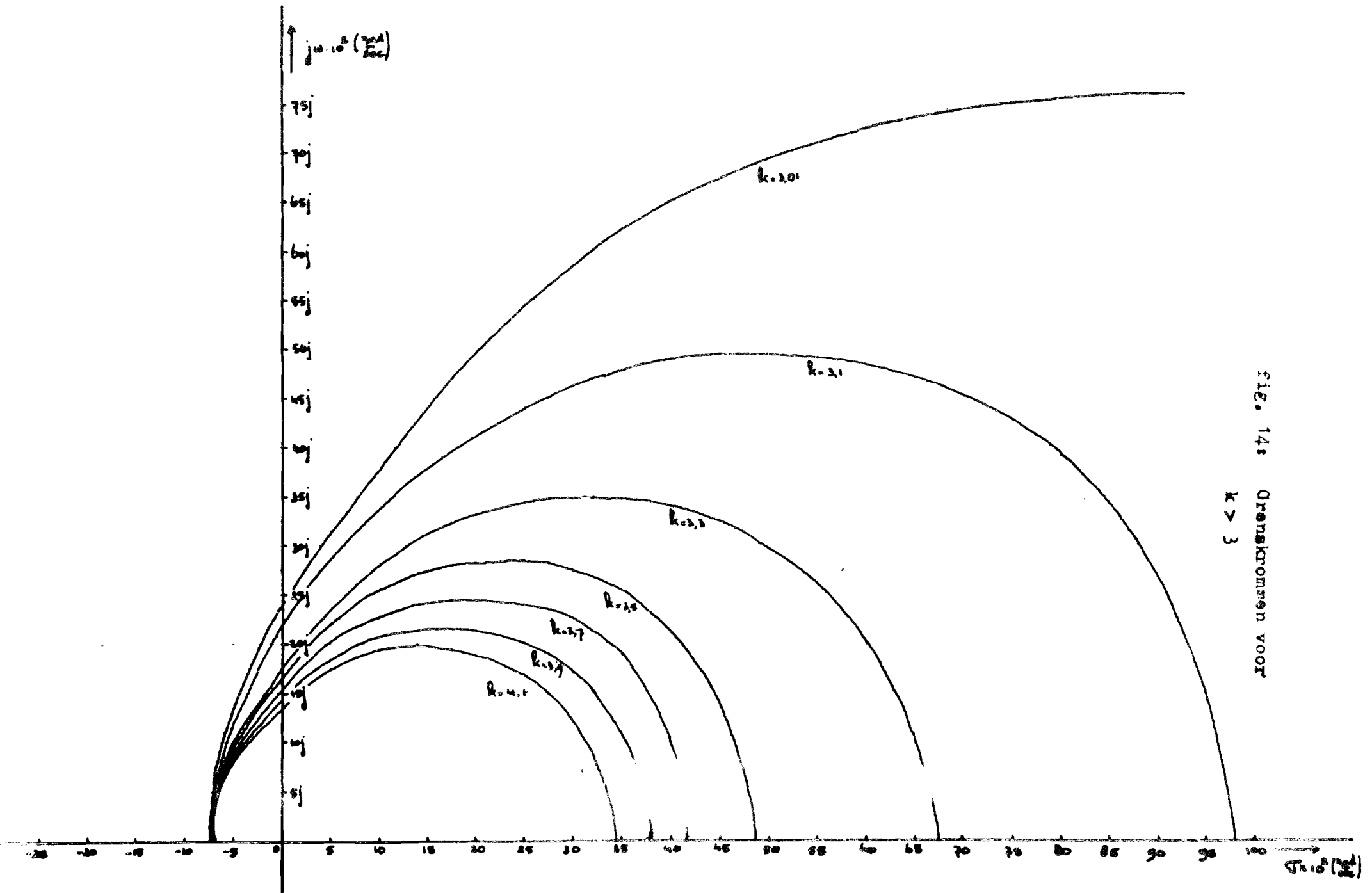


FIG. 14a
 Grenskrommen voor
 $k > 3$

3.5 Bepaling van nomogrammen voor enkele filters

De in de vorige paragraaf beschreven procedure RIBEP kunnen we ook gebruiken voor het berekenen van nomogrammen voor diverse filters, waarin de weerstandswaarden worden uitgezet als functie van de versterkingsfactor.

Nu gebruiken we de procedure echt voor het bepalen van de oplossingen R_1 , R_2 en R_3 als functie van de versterkingsfactor. De poolligging in het filter voeren we, nadat we de reële pool met behulp van de frequentietransformatie in $p = -1$ hebben gelegd, in met de parameters α en β (formules (79), (81) en (82)).

De eerste nomogram dat ik heb berekend is dat voor het derde orde Butterworth filter (zie fig. 15). In dit nomogram ziet U, dat er realisaties mogelijk zijn voor versterkingsfactoren $1,863 \leq k < 3,0$. Voor versterkingsfactoren $1,863 \leq k < 2,0$ zijn er zelfs voor elke versterkingsfactor twee realisaties mogelijk. Vergelijken we dit met een eerdere publicatie van Huelsman (lit. 3) dan zien we, dat hij slechts één oplossing over het gehele traject $1,863 \leq k < 3,0$ heeft gevonden.

De wijze waarop de twee oplossingen voor versterkingsfactoren $k < 2$, slechts één oplossing voor $k > 2$ en géén voor $k < 1,863$ ontstaan kunt U zien in fig. 16, waarin de functies (92) en (93) zijn getekend voor de versterkingsfactoren $k = 1,869$, $k = 1,863$, $k = 1,870$, $k = 1,980$, $k = 2,000$ en $k = 2,100$.

We zien, dat we op deze manier wel alle oplossingen vinden (vergelijk fig. 15 met lit. 3).

Ik heb tevens het nomogram bepaald voor de derde orde sectie van het vijfde orde Butterworth filter, waarbij de derde orde sectie de polen realiseert die het verst van de imaginaire as liggen (fig. 17).

Voor het vijfde orde Butterworth filter is het noemerpolynoom:

$$N(p) = (p+1)(p^2 + 1,618033989 p + 1)(p^2 + 0,6180339888 p + 1) \quad (119)$$

$f_3(\Omega)$

fig 16a: functies (g_2) en (g_3)
voor $k = 1,860$ ($\alpha = 1$ en $\beta = 1$)

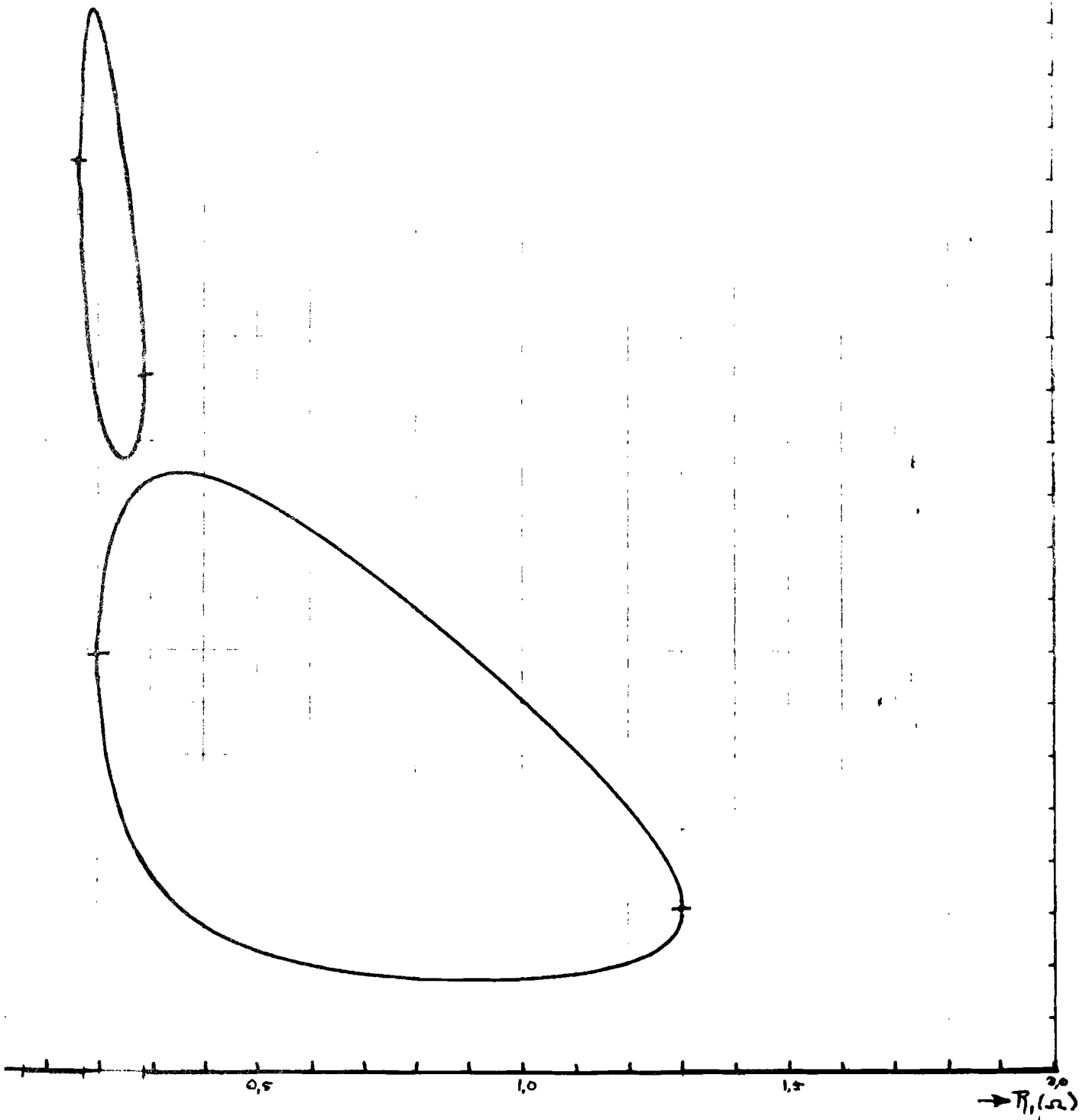
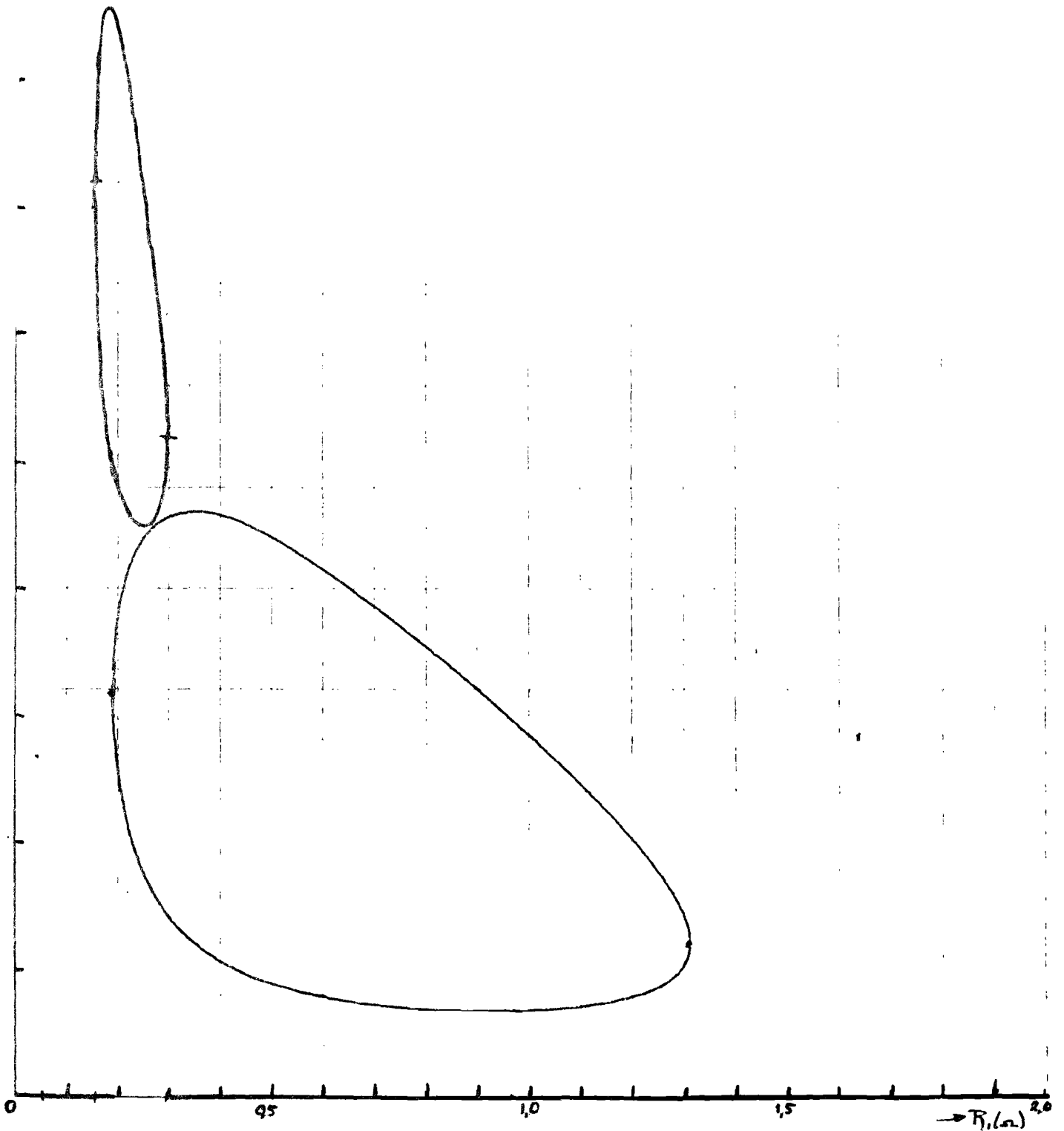
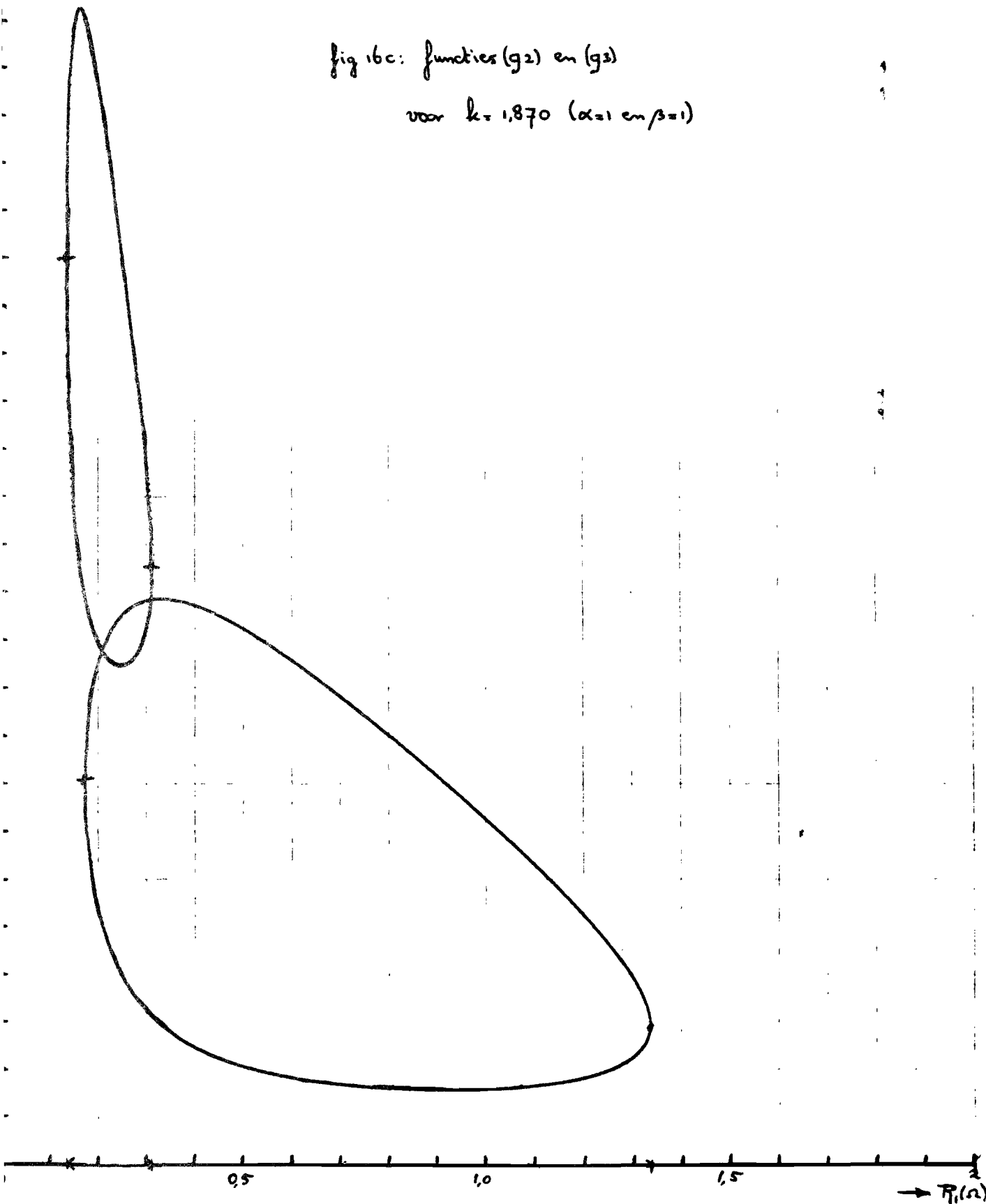


fig. 16 b : functies (g_2) en (g_3)
voor $k = 1,863$ ($\alpha = 1$ en $\beta = 1$)



(a)

fig 16c: functies (g_2) en (g_3)
voor $k = 1,870$ ($\alpha = 1$ en $\beta = 1$)



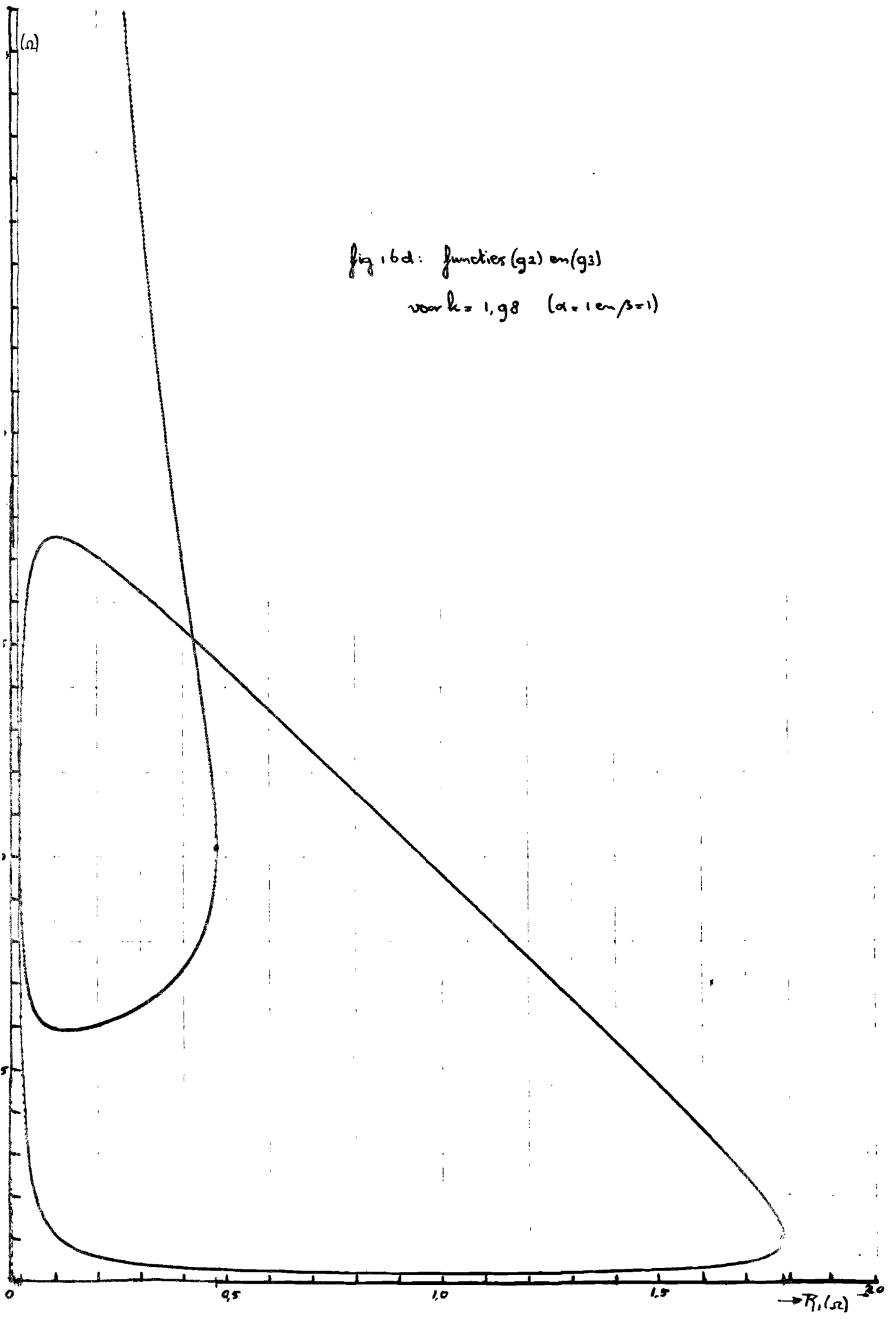


fig. 1.6d: functies g_2 en g_3
voor $k = 1,98$ ($\alpha = 1$ en $\beta = 1$)

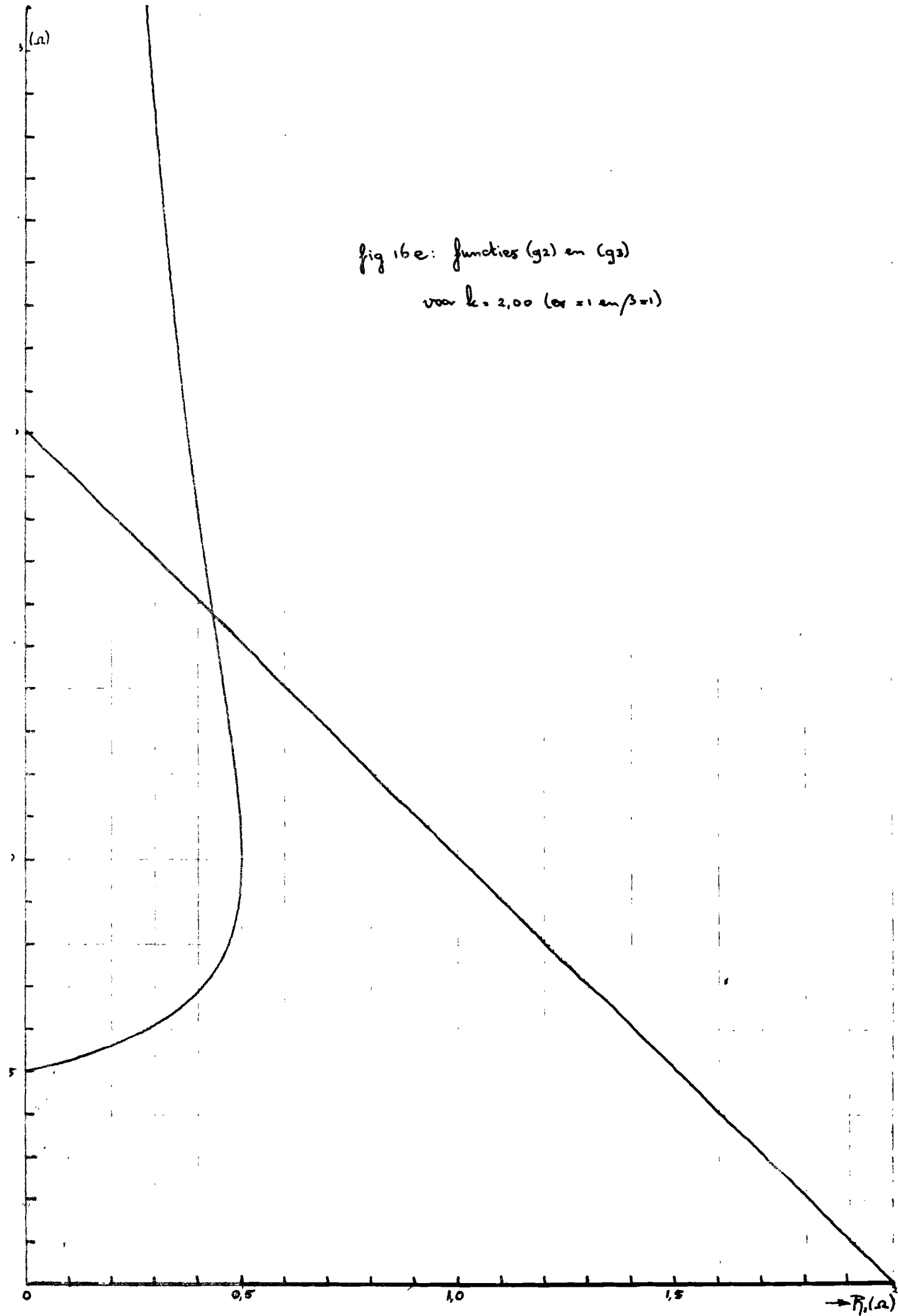


fig 16 e: functies (g_2) en (g_3)
voor $k = 2,00$ ($\alpha = 1$ en $\beta = 1$)

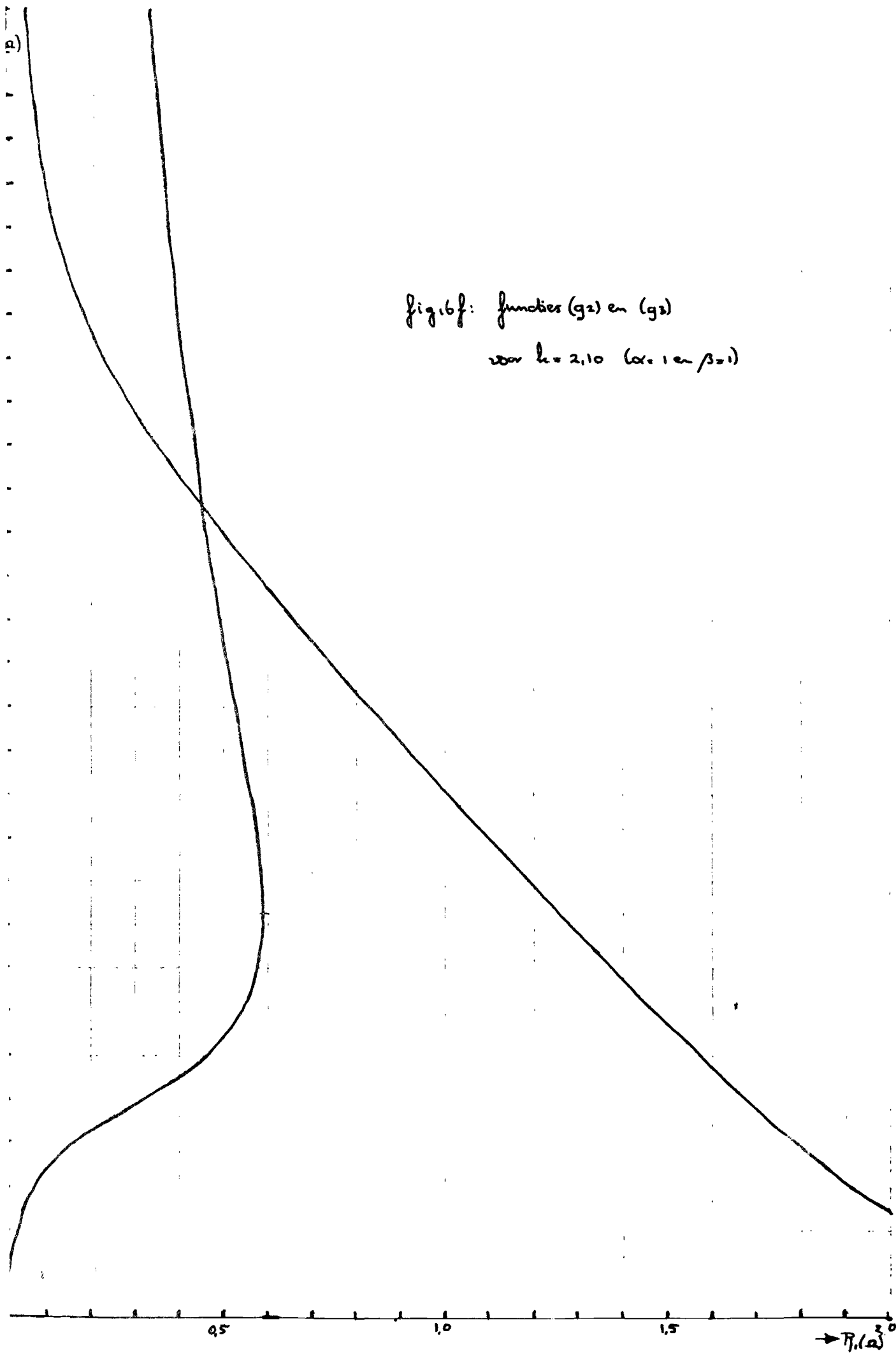
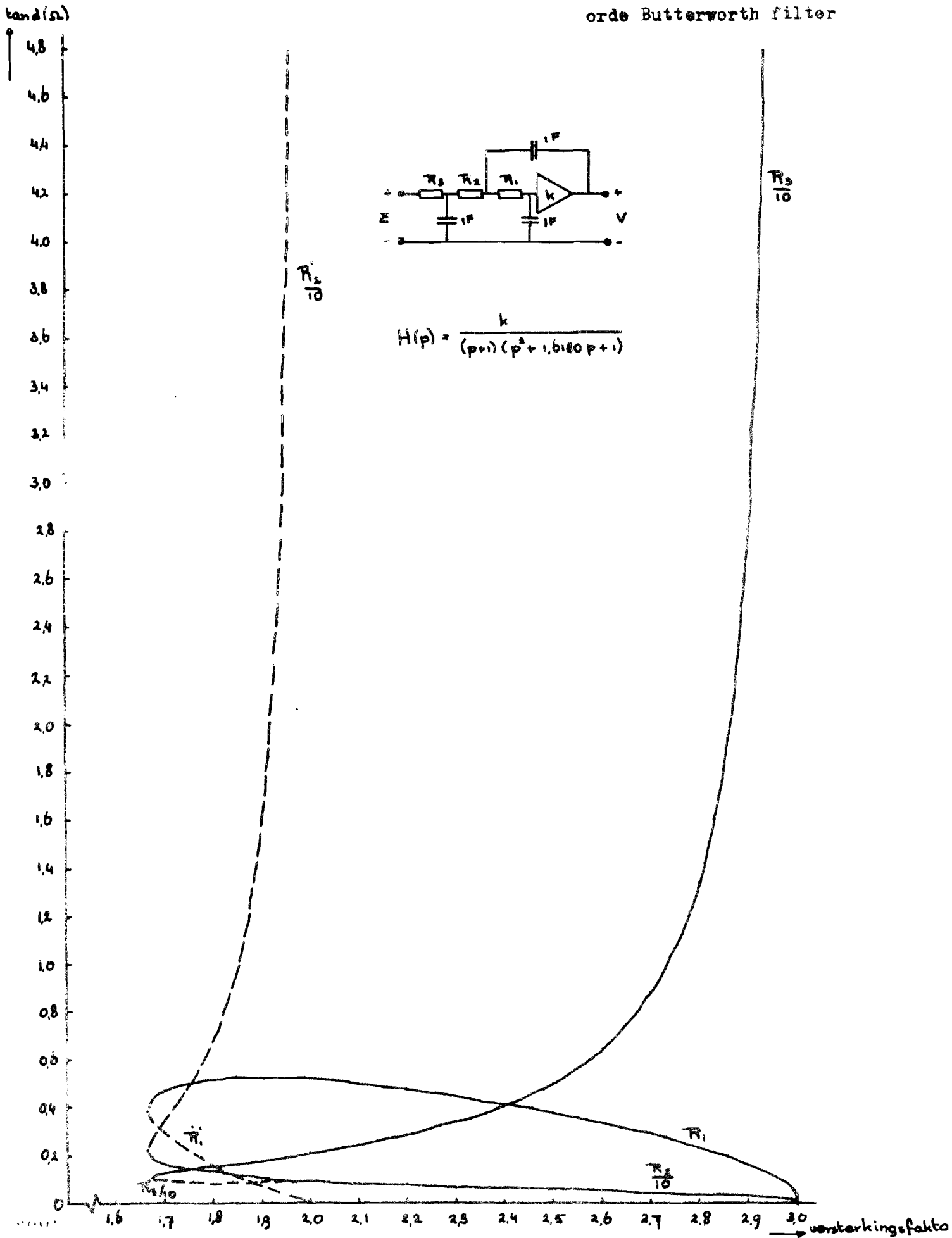


fig. 6 f: functies (g_2) en (g_3)

voor $h = 2,10$ ($\alpha = 1$ en $\beta = 1$)

fig. 17: nomogram voor de derde
 orde sectie van het vijfde
 orde Butterworth filter



Het nomogram geldt voor het polynoom:

$$P(p) = (p+1)(p^2 + 1,618033989p + 1) \quad (120)$$

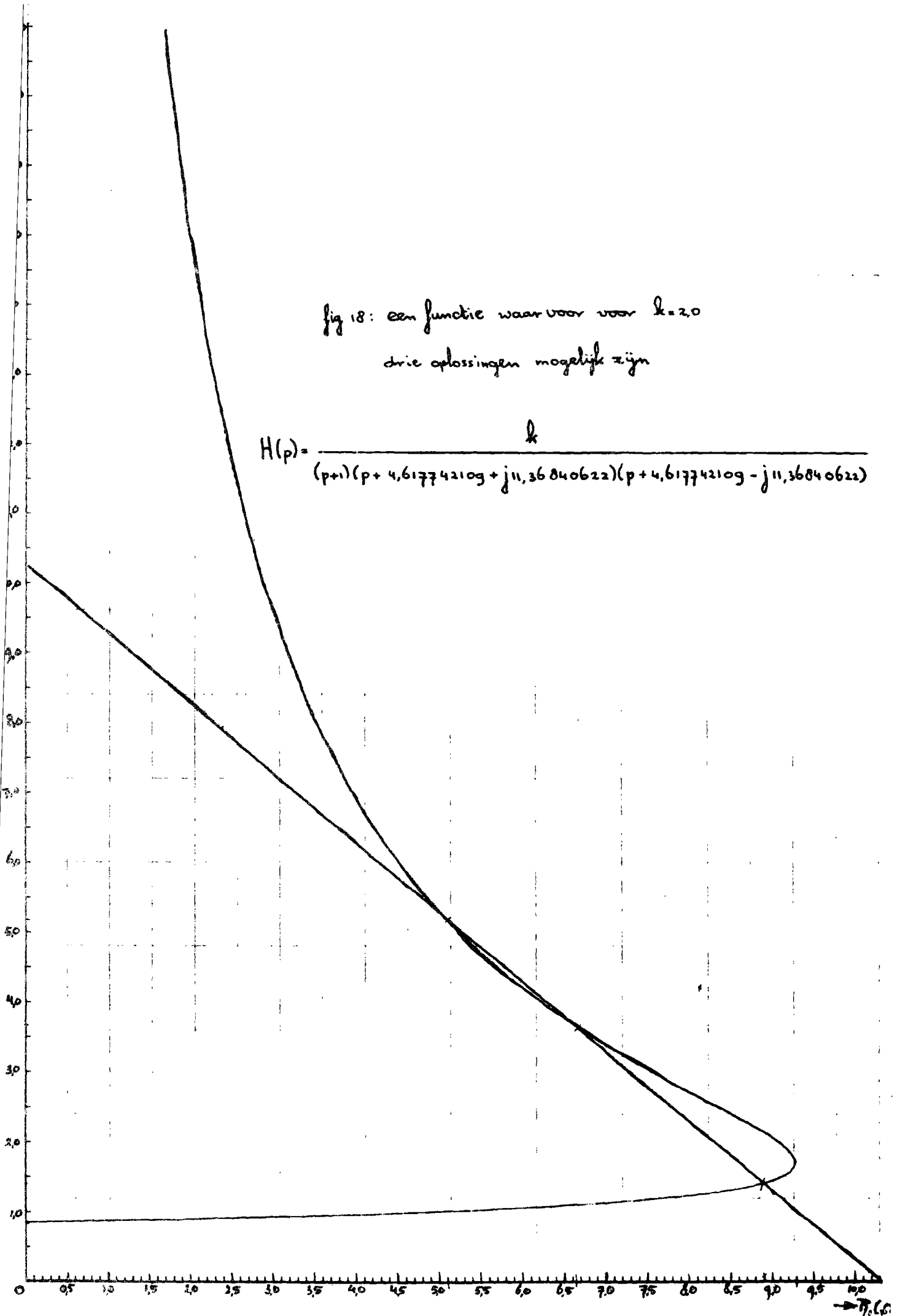
We zien nu dat het gebied waar twee oplossingen bestaan groter geworden ($1,66 \leq k < 2$).

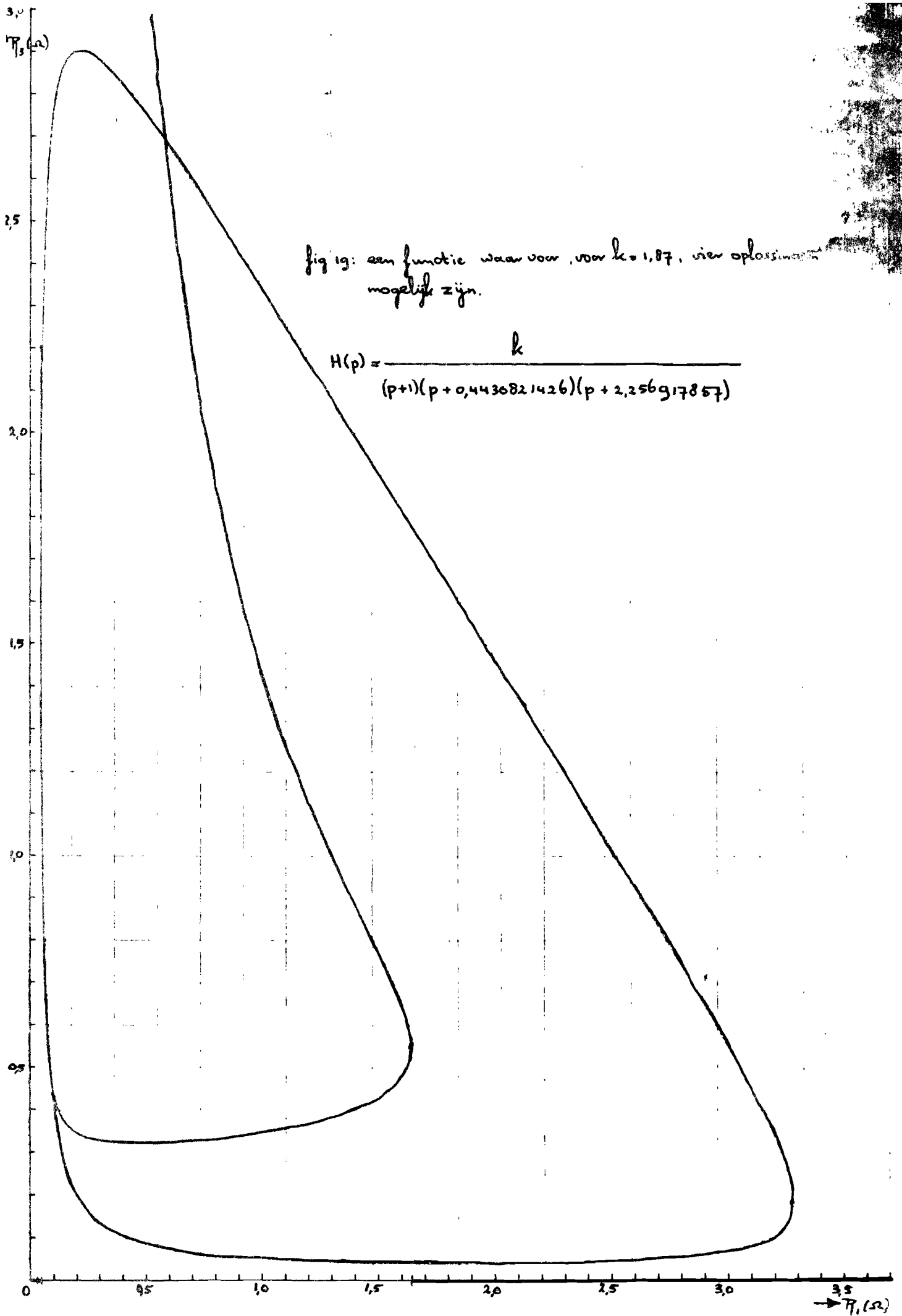
Zoals we hebben gezien kunnen we nu een nomogram maken voor elk type derde orde filter.

Het blijkt voor sommige poolliggingen mogelijk dat er zelfs meer dan twee oplossingen bestaan voor dezelfde versterkingsfactoren (zie fig. (18) en (19)).

fig 18: een functie waarvoor voor $k=20$
drie oplossingen mogelijk zijn

$$H(p) = \frac{k}{(p+1)(p+4,617742109 + j11,36840622)(p+4,617742109 - j11,36840622)}$$





4. Vierde Orde Sallen en Key Filter

4.1 Afleiding van de overdrachtsfunctie

Voor het afleiden van de overdrachtsfunctie van dit filter (fig. 1) gaan we uit van de formules (3.3) en (3.4) die we in paragraaf 3.1 hebben afgeleid.

Met formule (3.3)

$$i_3 = \left[p^3 C_1 R_1 C_2 R_2 C_3 + p^2 \{ C_1 R_1 C_3 + C_1 R_1 C_3 + (1-k) C_2 R_2 C_3 + C_1 R_1 C_2 \} + p \{ C_3 + C_1 + (1-k) C_2 \} \right] \cdot \frac{v_u}{k} \quad (1)$$

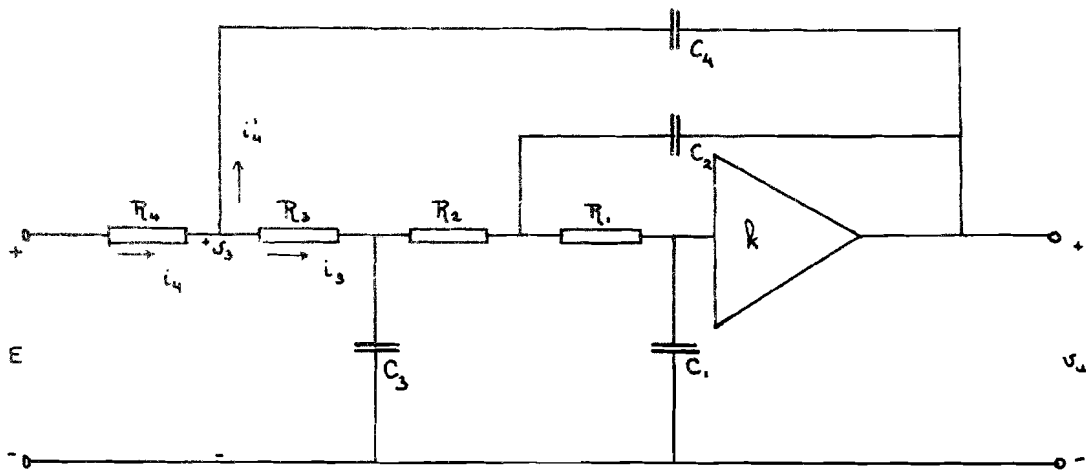


fig. 1: Vierde orde Sallen en Key filter

en met (3.4)

$$\begin{aligned}
 u_3 = & [p^3 C_1 \pi_1 C_2 \pi_2 C_3 \pi_3 + p^2 \{ C_1 \pi_2 C_3 \pi_3 + (1-k) C_2 \pi_2 C_3 \pi_3 + C_1 \pi_1 C_3 \pi_3 \\
 & + C_1 \pi_1 C_2 \pi_3 + C_1 \pi_1 C_2 \pi_2 \} + p \{ C_3 \pi_3 + C_1 \pi_3 + (1-k) C_2 \pi_3 + C_1 \pi_2 \\
 & + C_1 \pi_1 + (1-k) C_2 \pi_2 \} + 1] \cdot \frac{u_4}{k} \quad (2)
 \end{aligned}$$

$$\begin{aligned}
 i_4' = & [p^4 C_1 \pi_1 C_2 \pi_2 C_3 \pi_3 C_4 + \\
 & p^3 \{ C_1 \pi_2 C_3 \pi_3 C_4 + C_1 \pi_1 C_3 \pi_3 C_4 + (1-k) C_2 \pi_2 C_3 \pi_3 C_4 + C_1 \pi_1 C_2 \pi_2 C_4 \\
 & + C_1 \pi_1 C_2 \pi_2 C_4 \} + p^2 \{ C_3 \pi_3 C_4 + C_1 \pi_3 C_4 + (1-k) C_2 \pi_3 C_4 + C_1 \pi_2 C_4 \\
 & + C_1 \pi_1 C_4 + (1-k) C_2 \pi_2 C_4 \} + p (1-k) C_4] \frac{u_4}{k} \quad (3)
 \end{aligned}$$

$$i_4 = i_3 + i_4' \quad (4)$$

$$\begin{aligned}
 E = & [p^4 C_1 \pi_1 C_2 \pi_2 C_3 \pi_3 C_4 \pi_4 + \\
 & p^3 \{ C_1 \pi_2 C_3 \pi_3 C_4 \pi_4 + C_1 \pi_1 C_3 \pi_3 C_4 \pi_4 + (1-k) C_2 \pi_2 C_3 \pi_3 C_4 \pi_4 + C_1 \pi_1 C_2 \pi_2 C_4 \pi_4 \\
 & + C_1 \pi_1 C_2 \pi_2 C_4 \pi_4 + C_1 \pi_1 C_2 \pi_2 C_3 \pi_3 + C_1 \pi_1 C_2 \pi_2 C_3 \pi_4 \} \\
 & + p^2 \{ C_3 \pi_3 C_4 \pi_4 + C_1 \pi_3 C_4 \pi_4 + (1-k) C_2 \pi_3 C_4 \pi_4 + C_1 \pi_2 C_4 \pi_4 \\
 & + C_1 \pi_1 C_4 \pi_4 + (1-k) C_2 \pi_2 C_4 \pi_4 + C_1 \pi_2 C_3 \pi_4 + C_1 \pi_1 C_3 \pi_4 \\
 & + (1-k) C_2 \pi_2 C_3 \pi_4 + C_1 \pi_1 C_2 \pi_4 + C_1 \pi_2 C_3 \pi_3 + C_1 \pi_1 C_3 \pi_3
 \end{aligned}$$

$$\begin{aligned}
 & + (1-k)C_2R_2C_3R_3 + C_1R_1C_2R_3 + C_1R_1C_2R_2 \} \\
 & + p \{ (1-k)C_4R_4 + C_3R_4 + C_1R_4 + (1-k)C_2R_4 + C_3R_3 + C_1R_3 \\
 & \quad + (1-k)C_2R_3 + C_1R_2 + C_1R_1 + (1-k)C_2R_2 \} \\
 & + 1 \cdot \frac{v_u}{k} \tag{5}
 \end{aligned}$$

Uit (5) volgt voor de overdrachtsfunctie:

$$\begin{aligned}
 H(p) &= \frac{E}{v_u} \\
 &= \frac{k}{a_4 p^4 + a_3 p^3 + a_2 p^2 + a_1 p + 1} \tag{6}
 \end{aligned}$$

met:

$$a_4 = C_1R_1C_2R_2C_3R_3C_4R_4 \tag{7a}$$

$$\begin{aligned}
 a_3 &= C_1R_1C_3R_3C_4R_4 + C_1R_1C_3R_3C_4R_4 + (1-k)C_2R_2C_3R_3C_4R_4 \\
 & + C_1R_1C_2R_3C_4R_4 + C_1R_1C_2R_2C_4R_4 + C_1R_1C_2R_2C_3R_3 \\
 & + C_1R_1C_2R_2C_3R_4 \tag{7b}
 \end{aligned}$$

$$\begin{aligned}
 a_2 &= C_3R_3C_4R_4 + C_1R_1C_4R_4 + (1-k)C_2R_2C_4R_4 + C_1R_1C_4R_4 + C_1R_1C_4R_4 \\
 & + (1-k)C_2R_2C_4R_4 + C_1R_1C_3R_3C_4R_4 + C_1R_1C_3R_3C_4R_4 + (1-k)C_2R_2C_3R_3C_4R_4 + C_1R_1C_3R_3 \\
 & + C_1R_1C_3R_3 + (1-k)C_2R_2C_3R_3 + C_1R_1C_2R_3 + C_1R_1C_2R_2 \tag{7c} \\
 & + C_1R_1C_2R_4
 \end{aligned}$$

$$\begin{aligned}
 a_1 = & (1-k) C_4 R_4 + C_3 R_4 + C_3 R_3 + C_1 R_3 + (1-k) C_2 R_3 \\
 & + C_1 R_2 + C_1 R_1 + (1-k) C_2 R_2 + C_1 R_4 \\
 & + (1-k) C_2 R_4
 \end{aligned} \tag{7d}$$

Als we aan alle condensatoren gelijke waarden toekennen ($C=1F$) worden deze coëfficiënten:

$$a_4 = R_1 R_2 R_3 R_4 \tag{8a}$$

$$a_3 = (2-k) R_2 R_3 R_4 + 2 R_1 R_3 R_4 + 2 R_1 R_2 R_4 + R_1 R_2 R_3 \tag{8b}$$

$$a_2 = (R_3 R_4 + 2 R_2 R_4 + R_2 R_3)(2-k) + R_3 R_4 + 3 R_1 R_4 + 2 R_1 R_3 + R_1 R_2 \tag{8c}$$

$$a_1 = (2 R_4 + R_3 + R_2)(2-k) + R_1 + R_3 \tag{8d}$$

Vergelijken we dit resultaat met de coëfficiënten zoals deze zijn afgeleid door Stephenson (lit. 4) dan zien we dat zij inderdaad overeenstemmen.

4.2 Enige beschouwingen aan het filter

Voor het tweede en derde orde filter met gelijke capaciteiten hebben we gezien dat met een versterkingsfactor $k = 2$ alle poelligingen in het linkerhalfvlak van het complexe frequentiedomein realiseerbaar waren en dat er geen enkele oplossing in het rechterhalfvlak gerealiseerd kan worden.

Met een enkel voorbeeld zullen we zien dat voor het vierde orde filter deze versterkingsfactor niet meer voldoende is om te allen tijde een stabiel filter te garanderen.

Voor het vierde orde filter luiden de Hurwitzvoorwaarden voor de vergelijking:

$$a_4 p^4 + a_3 p^3 + a_2 p^2 + a_1 p + 1 = 0 \quad (9)$$

opdat deze oplossingen heeft met een negatief reëel deel:

$$a \quad a_1, a_2, a_3 \text{ en } a_4 > 0 \quad (10a)$$

$$b \quad a_1 a_2 a_3 - a_3^2 - a_1^2 a_4 > 0 \quad (10b)$$

Kiezen we als voorbeeld:

$$R_1 = R_3 = R_4 = 1 \Omega$$

$$R_2 = 12 \Omega \quad (11)$$

en substitueren we dit met $k = 2$ in de vergelijking (8a) tot en met (8d) dan vinden we voor de coëfficiënten:

$$\begin{aligned} a_4 &= 12 \quad [s^4] \\ a_3 &= 38 \quad [s^3] \\ a_2 &= 18 \quad [s^2] \\ a_1 &= 1 \quad [s^1] \end{aligned} \tag{12}$$

Als we dit invullen in (10b) zien we dat aan deze eis niet is voldaan.

Dit betekent dat er één of meerdere polen in het rechterhalfvlak liggen.

We kunnen wel een voorwaarde voor de ligging van twee genormeerde polen afleiden waarvoor de andere twee zeker nooit in het rechterhalfvlak kunnen liggen. Leg daartoe de genormeerde polen op de eenheidscirkel.

Dit levert:

$$a_4 p^4 + a_3 p^3 + a_2 p^2 + a_1 p + 1 = (p^2 + \beta p + 1) \{ a_4 p^2 + (a_3 - \beta a_4) p + 1 \} \tag{13}$$

Opdat aan het gelijkteken wordt voldaan moeten we eisen:

$$a \quad 1 + a_4 + \beta(a_3 - \beta a_4) = a_2 \tag{14a}$$

$$b \quad \beta + a_3 - \beta a_4 = a_1 \tag{14b}$$

Ofwel:

$$a_2 + (\beta^2 - 1) a_4 = 1 + \beta a_3 \tag{15a}$$

$$a_1 + \beta a_4 = \beta + a_3 \tag{15b}$$

Stel nu $a_3 - \beta a_4 = \delta$ met $\delta < 0$ (d.w.z. polen in het rechterhalfvlak)

Dit geeft met vergelijking (15b):

$$a_1 - \beta + \delta = \eta_1 + \eta_3 \quad (16)$$

hieruit volgt:

$$\eta_1 \leq \beta \quad \text{en} \quad \eta_3 \leq \beta \quad (17)$$

Uit de eis volgt ook:

$$a_3 < \beta a_4$$

ofwel

$$\eta_1 \eta_2 \eta_3 + 2 \eta_1 \eta_2 \eta_4 + 2 \eta_1 \eta_3 \eta_4 < \beta \eta_1 \eta_2 \eta_3 \eta_4 \quad (18)$$

Er geldt zeker:

$$2 \eta_1 \eta_2 \eta_4 < \beta \eta_1 \eta_2 \eta_3 \eta_4$$

$$2 < \beta \eta_3 \quad (19)$$

En met (17):

$$2 < \beta^2$$

$$\beta > \sqrt{2} \quad (20)$$

Voor $\beta < \sqrt{2}$ zijn hiermee zeker nooit polen in het rechterhalfvlak mogelijk (zie ook fig. 2).

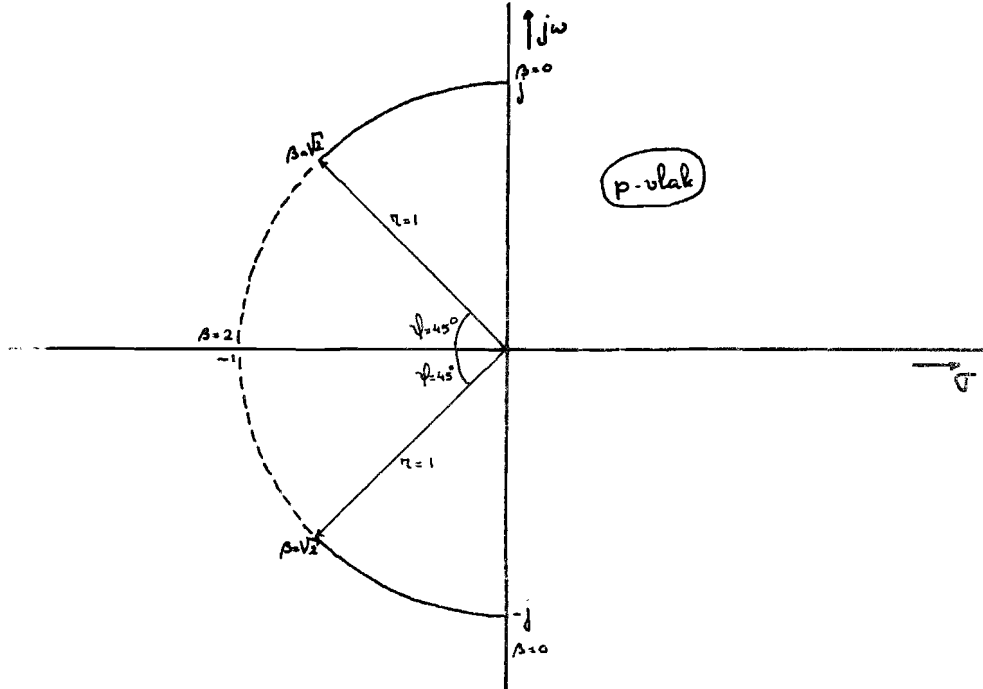


fig. 2: Als de twee genormeerde polen op het doorgetrokken deel van de eenheidscirkel liggen, kunnen de twee overblijvende polen nooit in het rechterhalfvlak komen.

4.3 Mogelijke realisaties door het filter met versterkingsfactor $k = 2$ en gelijke capaciteiten als functie van de ligging van twee vastgelegde, genormeerde polen

We gaan nu proberen om, net als bij het derde orde geval in het vorige hoofdstuk is gedaan, krommen te bepalen, die aangeven in welk deel van het complexe frequentievlak poolliggingen met dit filter (fig.1)gerealiseerd kunnen worden. Bij het derde orde filter hebben we daartoe de reële pool genormeerd en hebben we volgens de krommen, die deze grenzen aangeven, bepaald als functie van de versterkingsfactor k van het filter. Deze werkwijze is evenwel bij het vierde orde filter niet meer toepasbaar, omdat we nu niet meer kunnen zeggen dat er altijd een reële pool is die we kunnen normeren. We gaan nu dan ook anders te werk. We leggen eerst de versterkingsfactor van het filter vast om niet te veel vrijheidsgraden te krijgen. Een nadere beschouwing van de formules (8a) tot en met (8d) leert ons, dat deze heel wat gemakkelijker hanteerbaar zullen worden als we de versterkingsfactor de waarde $k = 2$ zullen geven (deze waarde is vanuit praktisch oogpunt ook het gemakkelijkst realiseerbaar). We gaan verder twee van de vier polen vastleggen, normeren, op de eenheidscirkel of reële as van het complexe frequentievlak; we doen dit door van het noemerpolynoom van vergelijking (6) een factor

$$p^2 + \beta p + 1 = 0 \quad (21)$$

af te splitsen.

Dit normeren van twee van de vier polen geeft geen verlies van de algemeenheid van de beschouwing, omdat door een frequentie-transformatie toch iedere willekeurige poolligging realiseerbaar blijft.

De polen van (6) kunnen we hiermee bepalen uit

$$a_4 p^4 + a_3 p^3 + a_2 p^2 + a_1 p + 1 = (p^2 + \beta p + 1)(\epsilon p^2 + \mu p + 1) = 0 \quad (22)$$

nadat we tenminste twee van de vier polen hebben getransformeerd tot een complex toegevoegd paar op de eenheidscirkel of tot twee reële polen, die aan de kwadratische vergelijking (21) voldoen.

Wat het eerste gelijkteken van vergelijking (22) waar is moeten we eisen:

$$a_2 - \beta(a_3 - \beta a_4) = 1 + a_4 \quad (23)$$

$$\underline{b} \quad a_1 - a_3 + \beta a_4 = \beta \quad (24)$$

$$\underline{c} \quad a_4 = \epsilon \quad (25)$$

$$\underline{d} \quad a_3 - \beta a_4 = \mu \quad (26)$$

Substitutie van de vergelijking (26) in (23) en (24) geeft:

$$a_2 = 1 + a_4 + \beta \mu \quad (27)$$

$$a_1 = \beta + \mu \quad (28)$$

Met vergelijking (8d), $k=2$ en (28) vinden we:

$$\bar{\eta}_3 = \beta + \mu - \bar{\eta}_1 \quad (29)$$

We zien hieruit direct dat we moeten eisen:

$$0 \leq \bar{\eta}_1 \leq \beta + \mu \quad (30)$$

opdat we steeds een positieve waarde voor R_1 vinden.

De vergelijkingen (8a), (25) en (29) leveren:

$$R_2 = \frac{\epsilon}{R_1(\beta + \mu - R_1)R_u} \quad (31)$$

Substitutie van de vergelijkingen (29) en (31) in (27) levert na uitwerking van de resulterende vergelijking:

$$\{(\mu + \beta - R_1) + 3R_1\}R_u^2 + \{2R_1(\mu + \beta - R_1) - 1 - \epsilon - \beta\mu\}R_u + \frac{\epsilon}{\beta + \mu - R_1} = 0$$

zodat:

$$R_{u1} = \frac{-2R_1(\mu + \beta - R_1) + 1 + \epsilon + \beta\mu}{2(\mu + \beta + 2R_1)}$$

$$\pm \frac{1}{2(\mu + \beta + 2R_1)} \sqrt{[1 + \epsilon + \beta\mu - 2R_1(\mu + \beta - R_1)]^2 - \frac{4(\mu + \beta + 2R_1)\epsilon}{\mu + \beta - R_1}} \quad (32)$$

Substitutie van (29) en (31) in (26) geeft na uitwerking:

$$2R_1(\mu + \beta - R_1)R_u^2 + \left\{ \frac{2\epsilon}{\mu + \beta - R_1} - \beta\epsilon - \mu \right\}R_u + \epsilon = 0$$

zodat ook moet gelden:

$$R_{u2} = \frac{\beta\epsilon + \mu - \frac{2\epsilon}{\mu + \beta - R_1}}{4R_1(\mu + \beta - R_1)}$$

$$\pm \frac{1}{4R_1(\mu + \beta - R_1)} \sqrt{\left[\mu + \beta\epsilon - \frac{2\epsilon}{\mu + \beta - R_1} \right]^2 - 8R_1(\mu + \beta - R_1)\epsilon} \quad (33)$$

We zien dat we ook nu naar die waarden van R_1 moeten zoeken die bij gegeven β , ϵ , en μ eenzelfde R_{41} (vergelijking (32)) en R_{42} (vergelijking (33)) opleveren; die waarden voor R_4 en R_1 en de bijbehorende waarde voor R_2 (uit vergelijking (31)) en R_3 (uit vergelijking (29)) zijn dan de oplossingen voor het filter met versterkingsfactor $k = 2$, die de gewenste poolligging - bepaald door β , μ en ϵ - realiseren.

We hebben hiermee een probleem dat analoog is aan de situatie in het vorige hoofdstuk (de vergelijkingen (92) en (93)).

De grenskrommen kunnen we ook weer op dezelfde wijze gaan bepalen als ik heb aangegeven in de figuren 6 en 7 van het vorige hoofdstuk.

We gaan voor elke gegeven waarde van β - hiermee ligt één poolpaar vast - zoeken naar ϵ 's en μ 's die behoren bij poolparen die nog juist realiseerbaar zijn.

Voor de coördinaten van de polen geldt:

$$\sigma = - \frac{\mu}{2\epsilon} \quad (\text{reële deel van de polen}) \quad (34)$$

$$\omega = \pm \sqrt{\frac{1}{\epsilon} - \frac{\mu^2}{4\epsilon^2}} \quad (\text{imaginaire deel van de polen}) \quad (35)$$

$$\pi = \sqrt{\frac{1}{\epsilon}} \quad (\text{afstand van de polen tot de oorsprong}) \quad (36)$$

Om de gezochte krommen te bepalen gaan we ook nu weer over de cirkel met straal $r = \sqrt{\frac{1}{\epsilon}}$, te beginnen bij een dubbele, negatief reële pool, bepalen, welke polen - bij een gegeven β - nog net realiseerbaar zijn; vervolgens verlagen we ϵ iets, zodat we een iets grotere cirkel gaan beschrijven, en we doen weer precies hetzelfde. Op deze wijze gaan we door. Dit levert ons dan de gewenste kurven.

De grenswaarde van R_1 tot welke we oplossingen kunnen verwachten hebben we al gevonden met vergelijking (30).

We moeten er verder nog voor zorgen dat de functies (32) en (33) in het beschouwde interval voor R_1 geen singulariteiten meer hebben, voordat we met de computer oplossingen gaan bepalen.

Wanneer we beide vergelijkingen met de factor

$$f(R_1) = (\mu + \beta - \pi R_1)^2 R_1 \quad (37)$$

vermenigvuldigen, bevatten de nieuwe functies inderdaad geen singuliere punten meer in het onderzochte interval.

De nieuwe functies worden:

$$\begin{aligned}
 f_{1,2}(\eta_1, \mu, \beta, \epsilon) &= (\mu + \beta - \eta_1)^2 \cdot \eta_1 \cdot \eta_{41} \\
 &= \frac{1}{4} \{ (\mu + \beta \epsilon)(\mu + \beta - \eta_1) - 2\epsilon \} \\
 &\quad \pm \sqrt{\frac{1}{16} \{ [(\mu + \beta \epsilon)(\mu + \beta - \eta_1) - 2\epsilon]^2 - 8\eta_1(\mu + \beta - \eta_1)^2 \cdot \epsilon \}} \quad (38)
 \end{aligned}$$

$$\begin{aligned}
 g_{1,2}(\eta_1, \mu, \beta, \epsilon) &= (\mu + \beta - \eta_1)^2 \cdot \eta_1 \cdot \eta_{42} \\
 &= \frac{\eta_1(\mu + \beta - \eta_1)^2}{2(\mu + \beta + 2\eta_1)} \{ -2\eta_1(\mu + \beta - \eta_1) + 1 + \epsilon + \beta\mu \} \\
 &\quad \pm \sqrt{\frac{\eta_1^2(\mu + \beta - \eta_1)^3}{4(\mu + \beta + 2\eta_1)^2} \left[(\mu + \beta - \eta_1) \{ 1 + \epsilon + \beta\mu - 2\eta_1(\mu + \beta - \eta_1) \}^2 \right.} \\
 &\quad \left. - 4(\mu + \beta + 2\eta_1)\epsilon \right]} \quad (39)
 \end{aligned}$$

Van deze functies kunnen we nu met de procedure RIBEP van het vorige hoofdstuk (fig. 3.10) de oplossingen R_1 en R_4 gaan bepalen.

Voor het zoeken van de grenskrommen kunnen we weer gebruik maken van de procedure MUSTAP (fig. 3.11) van het vorige hoofdstuk; we moeten daartoe α door ϵ vervangen en β door μ .

De volledige programmatekst vindt U in bijlage 2.

De grenskrommen, die ik heb berekend met behulp van bovenstaande, vindt U in fig. 3; de krommen zijn hier uitgezet voor diverse waarden

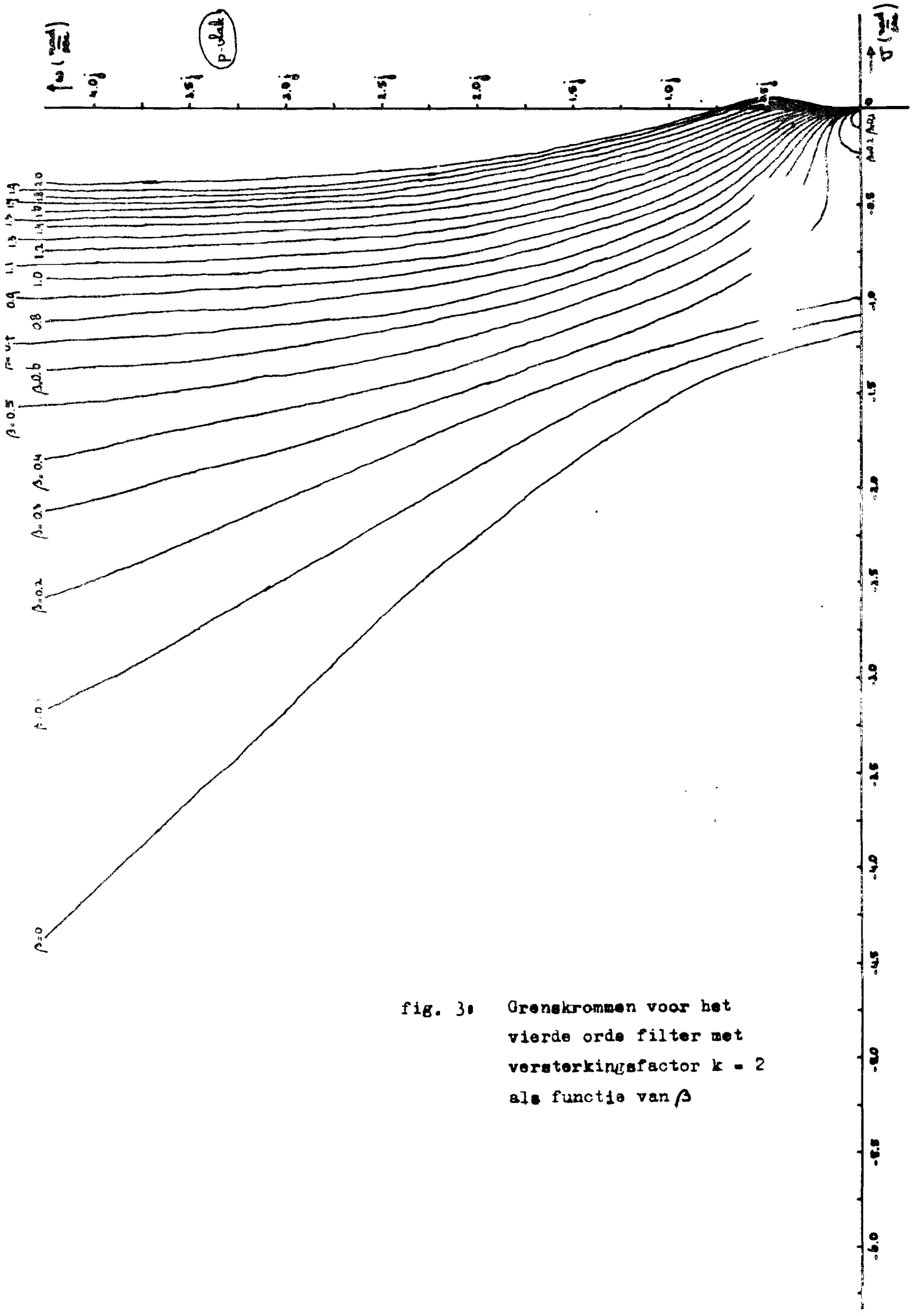


fig. 30 Grenskrommen voor het vierde orde filter met versterkingsfactor $k = 2$ als functie van β

van β ($0 \leq \beta \leq 2,0$) overeenkomend met diverse liggingen van de twee genormeerde polen op de eenheidscirkel in het complexe frequentievlak (van een zuiver imaginair complex toegevoegd poolpaar tot een dubbele negatieve pool).

Mit fig. 3 zien we dat voor $\beta \leq 1,4$ geen polen realiseerbaar zijn in het rechterhalfvlak en dat voor $\beta \geq 1,5$ de grenskromme een kleine uitstulping in het rechterhalfvlak krijgt. Dit is volledig in overeenstemming met hetgeen we in paragraaf 4.2 hebben afgeleid en gezien.

4.4 Bepaling van nomogrammen voor diverse filters

In de voorgaande paragraaf hebben we onder meer gebruik gemaakt van een programma waarmee we voor $k = 2$, bij een gegeven ligging van de polen, de waarden voor de weerstanden konden berekenen.

Als $k = 2$ praktisch wel een erg interessante waarde, omdat zij gemakkelijk realiseerbaar is, maar we willen toch ook voor andere versterkingsfactoren aangeven, welke waarden de weerstanden voor een ald type filter moeten hebben. We gaan dan ook enige formules afleiden, waarmee we, als we een goede startwaarde hebben -en deze kunnen we met de procedure RIBEP (zie vorige paragraaf) berekenen- nomogrammen voor elk type filter berekenen (itererend), waarin de weerstandswaarden worden gegeven als functie van de versterkingsfactor, bij een gegeven poolligging.

We gaan uit van uitdrukking (22):

$$a_4 p^4 + a_3 p^3 + a_2 p^2 + a_1 p + 1 = (p^2 + \beta p + 1)(\epsilon p^2 + \mu p + 1)$$

Substitutie van de coëfficiënten (8a) tot en met (8d) geeft:

$$a_4 = R_1 R_2 R_3 R_4 = \epsilon \quad (40a)$$

$$a_3 = (2-k)R_2 R_3 R_4 + 2R_1 R_3 R_4 + 2R_1 R_2 R_4 + R_1 R_2 R_3 = \epsilon\beta + \mu \quad (40b)$$

$$a_2 = (R_3 R_4 + 2R_2 R_4 + R_2 R_3)(2-k) + R_3 R_4 + 3R_1 R_4 + 2R_1 R_3 + R_1 R_2 = \beta\mu + \epsilon + 1 \quad (40c)$$

$$a_1 = (2R_4 + R_2 + R_3)(2-k) + R_1 + R_3 = \mu + \beta \quad (40d)$$

Als we hierin R_2 gegeven dan kunnen we met (40a) schrijven:

$$R_4 = \frac{\epsilon}{R_1 R_2 R_3} \quad (41)$$

Vergelijking (40b) geeft:

$$\mu = (2-k) \frac{\epsilon}{r_1} + \frac{2\epsilon}{r_2} + \frac{2\epsilon}{r_3} + r_1 r_2 r_3 - \epsilon \beta \quad (42)$$

Uit vergelijking (40c) vinden we:

$$\beta \mu = \left(\frac{\epsilon}{r_1 r_2} + \frac{2\epsilon}{r_1 r_3} + r_2 r_3 \right) (2-k) + \frac{\epsilon}{r_1 r_2} + \frac{2\epsilon}{r_2 r_3} + 2r_1 r_3 + r_1 r_2 - 1 - \epsilon \quad (43)$$

Vergelijking (40d) levert tot slot:

$$\mu = \left(\frac{2\epsilon}{r_1 r_2 r_3} + r_2 + r_3 \right) (2-k) + r_1 + r_3 - \beta \quad (44)$$

Uit vergelijking (42) en (44) lossen we k op:

$$2-k = \frac{\frac{2\epsilon}{r_2} + \frac{2\epsilon}{r_3} + r_1 r_2 r_3 - r_1 - r_3 - \beta(\epsilon-1)}{\frac{2\epsilon}{r_1 r_2 r_3} + r_2 + r_3 - \frac{\epsilon}{r_1}} \quad (45)$$

eruit verkrijgen we voor k de volgende uitdrukking:

$$k = 2 - \frac{2\epsilon r_1 r_3 + 2\epsilon r_1 r_2 + r_1^2 r_2^2 r_3^2}{2\epsilon + r_1 r_2^2 r_3 + r_1 r_2 r_3^2 - \epsilon r_2 r_3} + \frac{-r_1^2 r_2 r_3 - r_1 r_2 r_3^2 - r_1 r_2 r_3 \beta(\epsilon-1)}{2\epsilon + r_1 r_2^2 r_3 + r_1 r_2 r_3^2 - \epsilon r_2 r_3} \quad (46)$$

Substitueren we deze vergelijking (46) in (42) dan vinden we:

$$\mu = \varepsilon \cdot \frac{2\varepsilon R_3 + 2\varepsilon R_2 + R_1 R_2^2 R_3^2 - R_1 R_2 R_3 - R_2 R_3^2 - R_2 R_3 / \beta (\varepsilon - 1)}{2\varepsilon + R_1 R_2^2 R_3 + R_1 R_2 R_3^2 - R_2 R_3 \varepsilon}$$

$$+ \frac{2\varepsilon}{R_2} + \frac{2\varepsilon}{R_3} + R_1 R_2 R_3 - \varepsilon / \beta \quad (47)$$

Herleiden van deze uitdrukking resulteert in de volgende kwadratische vergelijking in R_1 :

$$\{-R_2^3 R_3^2 - R_2^2 R_3^3\} R_1^2$$

$$+ \{R_2^2 R_3 (\mu + \beta \varepsilon) + R_2 R_3^2 (\mu + \beta \varepsilon) - 5\varepsilon R_2 R_3 - 2\varepsilon R_3^2 - 2\varepsilon R_2^2\} R_1$$

$$+ \{2\varepsilon (\mu + \beta \varepsilon) - R_2 R_3 \varepsilon (\mu + \beta) - \frac{4\varepsilon^2}{R_2} - \frac{4\varepsilon^2}{R_3} + \varepsilon R_2 R_3^2\} = 0 \quad (48)$$

Voor gegeven waarden voor ε , β en μ (bepaald door de poelligging) en voor R_2 en R_3 kunnen we hieruit R_1 berekenen. De oplossingen die voldoen zijn alleen die waarden voor R_1 en R_3 die, voor gegeven ε , β , μ en R_2 , tevens voldoen aan vergelijking (43) en daarin (46) gesubstitueerd:

$$\beta \mu = \left(\frac{\varepsilon}{R_1 R_2} + \frac{2\varepsilon}{R_1 R_3} + R_2 R_3 \right).$$

$$\frac{2\varepsilon R_1 R_3 + 2\varepsilon R_1 R_2 + R_1^2 R_2^2 R_3^2 - R_1^2 R_2 R_3 - R_1 R_2 R_3^2 - R_1 R_2 R_3 / \beta (\varepsilon - 1)}{2\varepsilon + R_1 R_2^2 R_3 + R_1 R_2 R_3^2 - \varepsilon R_2 R_3} +$$

$$\begin{aligned}
 & + \frac{\epsilon}{R_1 R_2} + \frac{3\epsilon}{R_2 R_3} + 2R_1 R_3 + R_1 R_2 \\
 -1 - \epsilon & = f_{pp}(\epsilon, \beta, \mu, R_1, R_2, R_3) \quad (49)
 \end{aligned}$$

nu al itererend een nomogram voor een bepaald type filter bepalen gaan we uit van een goede startwaarde voor de weerstanden, die voor een bepaalde, willekeurige versterkingsfactor in de vergelijkingen (40a) tot en met (40d) en daarmee ook aan vergelijkingen (48) en (49) voldoen. We kennen nu aan R_2 een iets gewijzigde waarde toe:

$$R_2 + \Delta R_2 \rightarrow R_2 \quad (50)$$

Vervolgens gaan we, door R_3 te variëren in een iteratieproces, achtereenvolgens bij die nieuwe waarden voor R_2 en R_3 met vergelijking (48) de bijbehorende waarde voor R_1 berekenen -we nemen hiervoor de oplossing voor R_1 , die het dichtst bij de beginwaarde ligt- en met deze waarden voor R_1 , R_2 en R_3 het rechterlid van vergelijking (49) bepalen. We gaan nu de waarde voor R_3 net zo lang veranderen zolang nodig is om aan het gelijkteken van vergelijking (49) te voldoen. Op het moment dat we waarden voor R_1 en R_3 hebben gevonden die, met de nieuwe waarde voor R_2 , aan vergelijking (49) voldoen, hebben we een nieuwe oplossing gevonden. Met vergelijking (41) en (46) berekenen we de bij deze oplossing behorende waarden voor R_4 en de vergelijkingfactor k .

Door dit proces te herhalen (gebruik de zojuist gevonden oplossing als startwaarde voor een volgende iteratiecyclus) kunnen we, door de waarde van R_2 te variëren tussen 0 en ∞ , het volledige nomogram voor het beschouwde filter afleiden.

In fig. 4 vindt U een schematisch overzicht van het programma dat

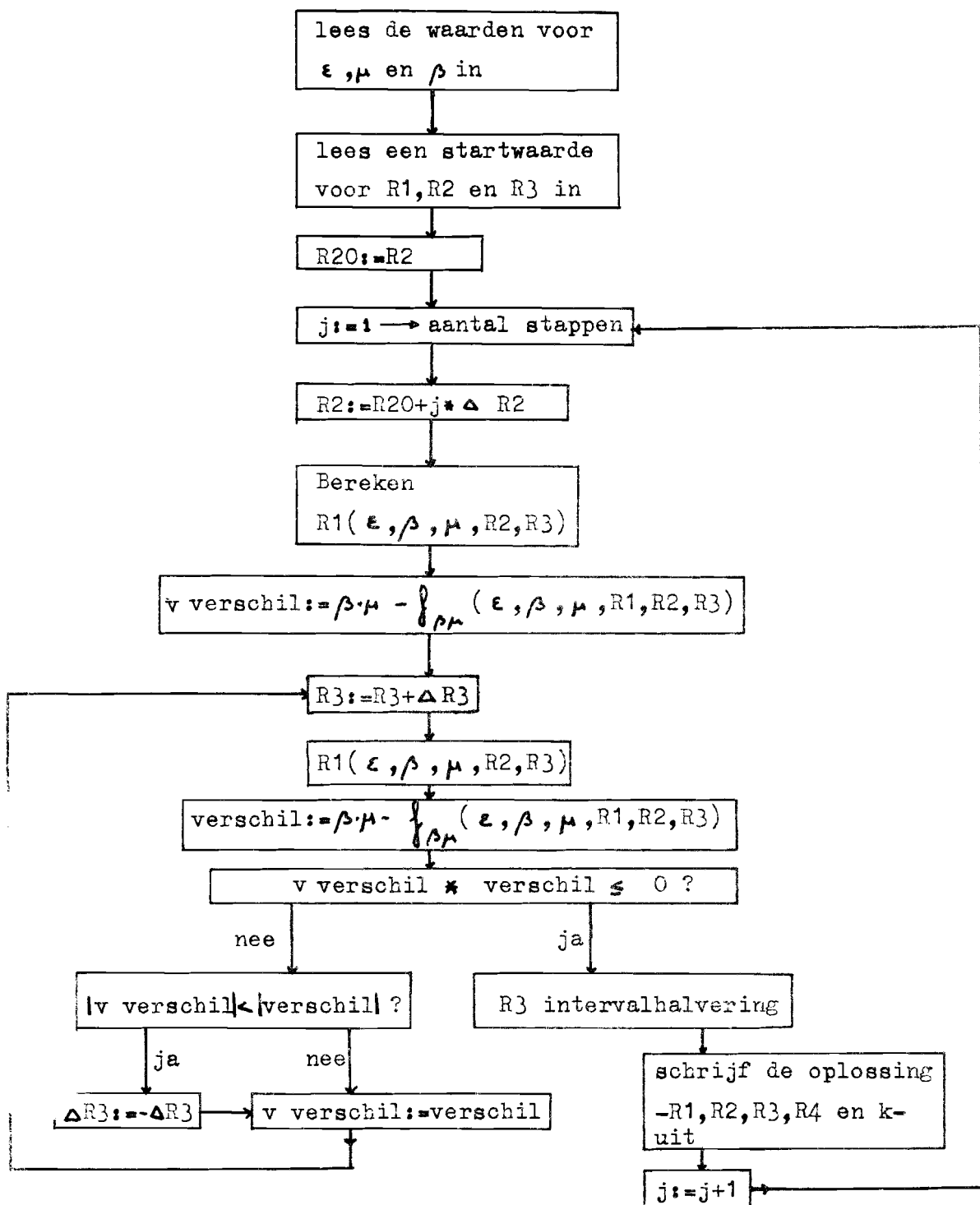


fig. 4: programma dat, uitgaande van een goede startwaarde, het nomogram voor een bepaald type vierde orde filter berekent.

Bovenstaande bewerkingen uitvoert.

De listing van het programma, dat de nomogrammen berekent, vindt U in bijlage 3.

Met behulp van dit programma heb ik eerst het nomogram voor het vierde orde Butterworth filter berekend. De overdrachtsfunctie voor dit filter luidt:

$$|H(p)|^2 = \frac{k}{(p^2 + 1.8477p + 1)(p^2 + 0.76537p + 1)} \quad (51)$$

Als startwaarde heb ik de weerstandswaarden van het filter voor de versterkingsfactor $k = 2$ berekend met de procedure RIBEP van de vorige paragraaf (zie ook fig. 3.10 en bijlage 2). Als startwaarden heb ik gevonden

$$\begin{aligned} R_1 &= 0.1339 \Omega \\ R_2 &= 3.8931 \Omega \\ R_3 &= 2.4791 \Omega \\ R_4 &= 0.7736 \Omega \end{aligned}$$

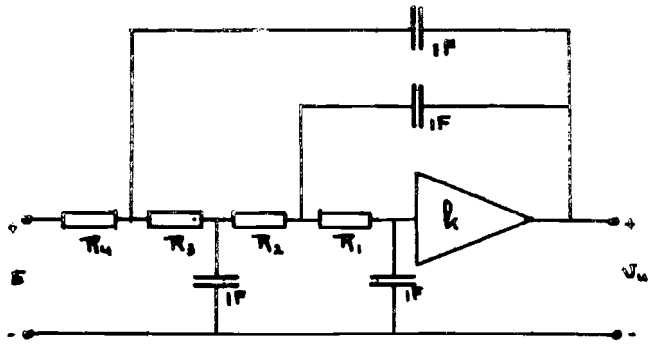
Het nomogram dat ik hiermee heb verkregen ziet U in fig. 5a en fig. 5b.

Vergelijken we dit met het nomogram dat Huelsman heeft berekend (lit. 5) dan zien we direkt dat dit bijzonder onvolledig is.

Huelsman heeft voor versterkingsfactoren $k < 2$ geen enkele oplossing aangegeven, terwijl er, zoals we in fig. 5 zien, in een klein gebied zelfs twee oplossingen bestaan ($1.9836 \leq k < 2$). Ook heeft Huelsman niet aangegeven dat er tot $k = 3$ een oplossing mogelijk is.

versterking (k)

5.0
4.8
4.6
4.4
4.2
4.0
3.8
3.6
3.4
3.2
3.0
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0



$$H(p) = \frac{k}{(p^2 + 1.8477p + 1)(p^2 + 0.7654p + 1)}$$

R_2

$10R_1$

$\frac{R_3}{10}$

$10R_1$

R_4

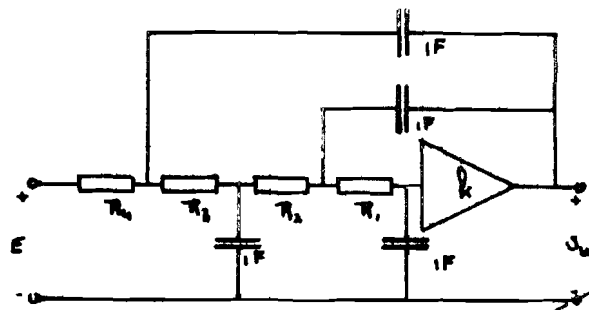
1.9

2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3.0

versterkingfactor (k)

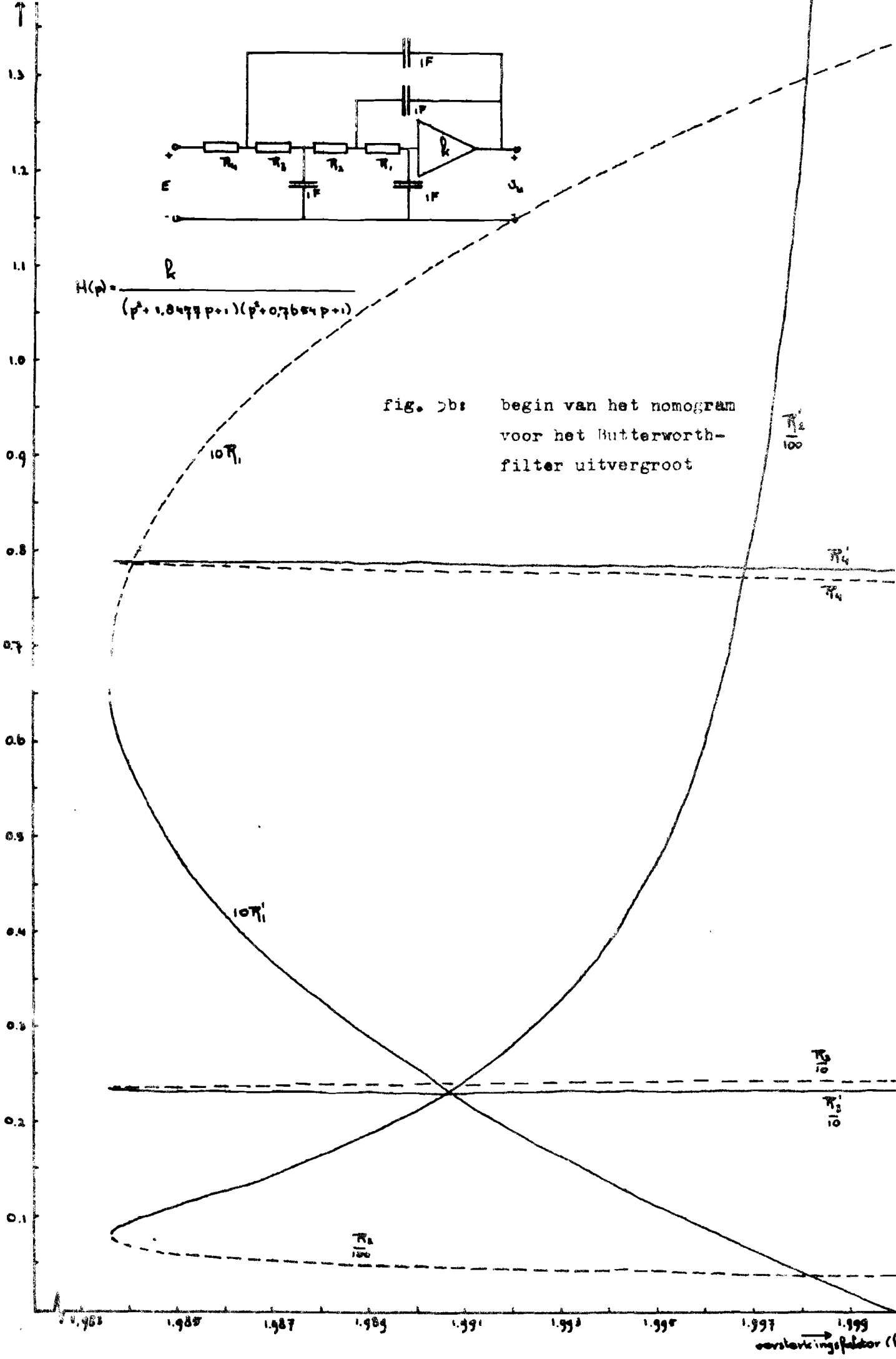
fig. 5a:

nomogram voor het vierde orde Butterworth filter



$$H(p) = \frac{R}{(p^2 + 1.8477p + 1)(p^2 + 0.7654p + 1)}$$

fig. 2b: begin van het nomogram voor het Butterworth-filter uitvergroot



Vergelijken we het hierboven berekend nomogram (fig. 5a en 5b) met het in het vorige hoofdstuk berekend nomogram voor het derde orde Butterworth filter, dat zien we dat ook hier weer $k = 2$ en $k = 3$ limietgevallen zijn.

Vaast het nomogram voor dit filter heb ik ook nog het nomogram bepaald voor het Bessel filter met de overdrachtsfunctie:

$$|p| = \frac{k}{(p^2 + 1,2414p + 1)(1,2567p^2 + 2,1480p + 1)} \quad (52)$$

waarden voor de coëfficiënten in bovenstaande overdrachtsfunctie heb ik berekend uit de poolligging zoals die is gegeven in lit. 6. startwaarden heb ik weer met de procedure RIBEP berekend. heb hiervoor gevonden:

$$R_1 = 0,25478 \Omega$$

$$R_2 = 2,2235 \Omega$$

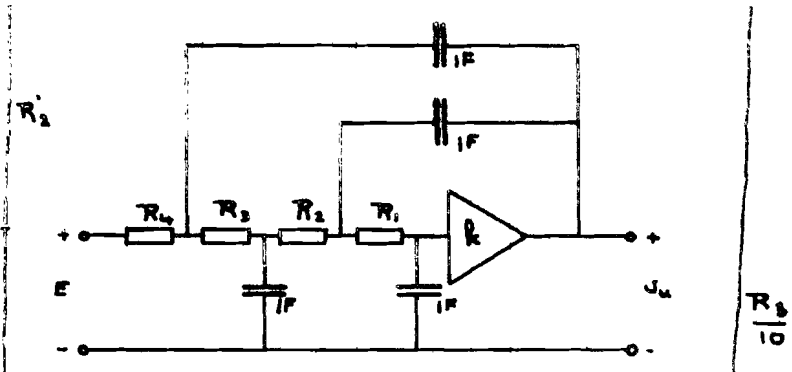
$$R_3 = 3,1346 \Omega$$

$$R_4 = 0,70777 \Omega$$

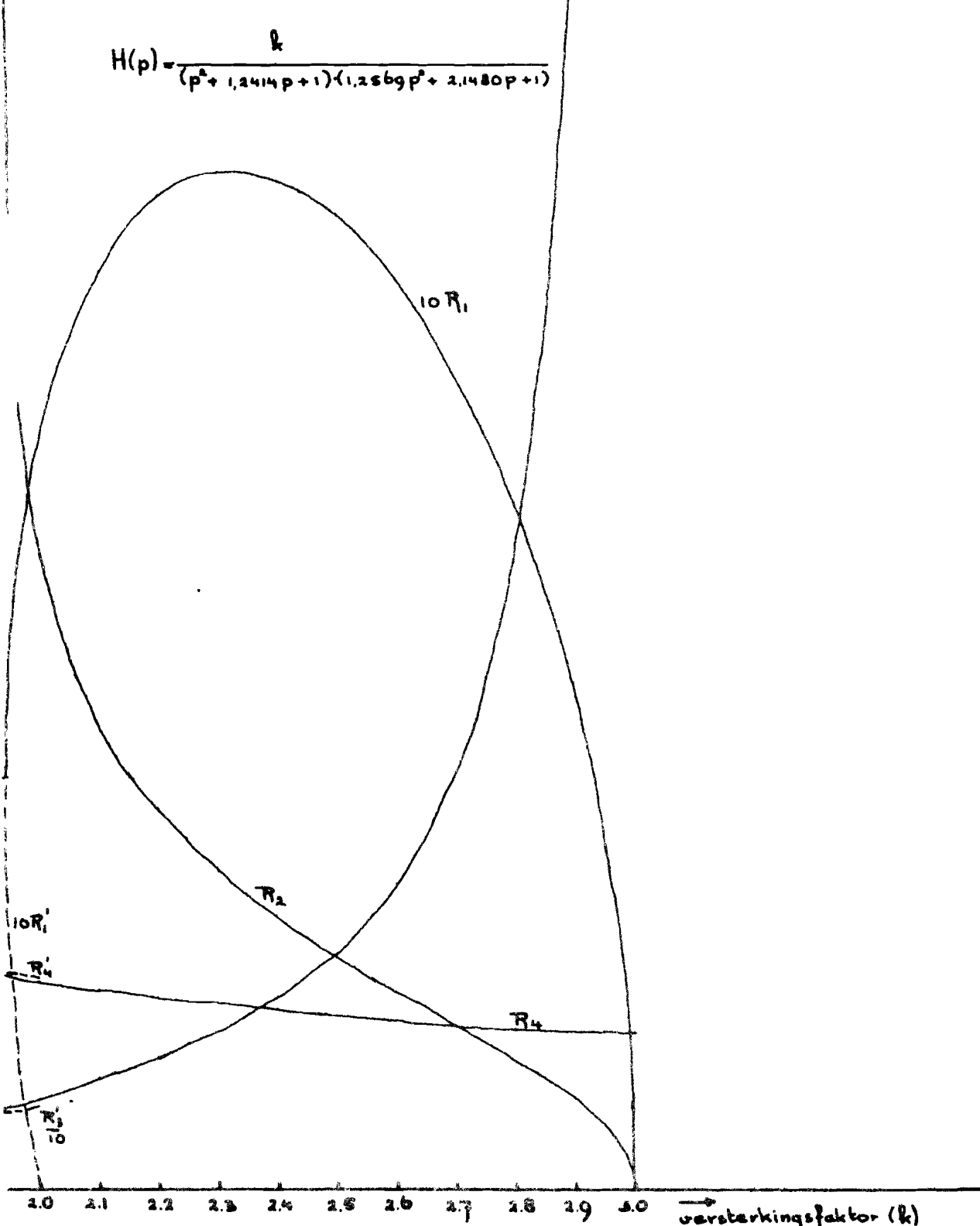
Het nomogram ziet U getekend in fig. 6. Zoals we zien is het gebied, waar twee oplossingen mogelijk zijn, groter geworden ($1,942 \leq k < 2$). Ook hier vormen de versterkingsfactoren $k = 2$ en $k = 3$ weer limietgevallen.

weerstand (Ω)

fig. 6: Nomogram voor het Bessel filter



$$H(p) = \frac{k}{(p^2 + 1.2414p + 1)(1.2569p^2 + 2.1480p + 1)}$$



versterkingsfaktor (k)

Tot slot heb ik het nomogram bepaald voor een filter met de overdrachtsfunctie

$$H(p) = \frac{k}{(p^2 + 1,9p + 1)^2} \quad (53)$$

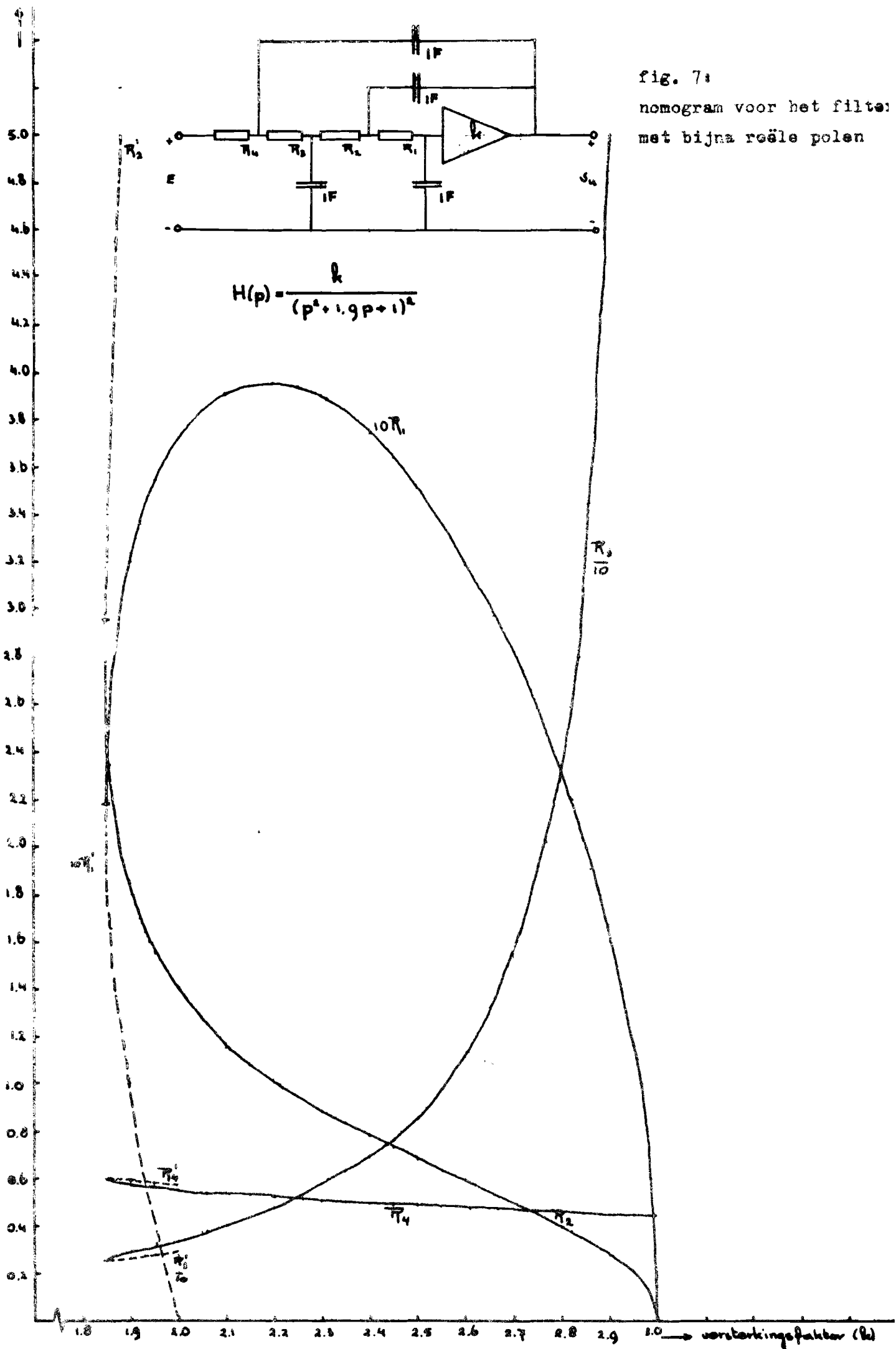
voor de startwaarden heb ik met behulp van de procedure RIBEP voor $k = 2$ gevonden:

$$\begin{aligned} R_1 &= 3,7231 \cdot 10^{-1} \Omega \\ R_2 &= 1,4050 \Omega \\ R_3 &= 3,4277 \Omega \\ R_4 &= 5,5770 \cdot 10^{-1} \Omega \end{aligned}$$

Het nomogram voor dit filter ziet U afgebeeld in fig. 7.

Het gebied, waar twee oplossingen mogelijk zijn, is weer groter geworden ($1,85 \leq k < 2$); de versterkingsfactoren $k = 2$ en $k = 3$ zijn opnieuw limietgevallen.

fig. 7:
nomogram voor het filter
met bijna reële polen



5. Fysische Beschouwing van de diverse Filters

De zijn tijdens de berekening van de realisatiemogelijkheden met de diverse orde filters met gelijke capaciteiten enige waarden voor de versterkingsfactoren tegengekomen, waarbij iets opmerkelijks gebeurde. Zo bleek vanaf de versterkingsfactor $k = 2$ voor alle orde filters één oplossing eerder mogelijk. Voor $k \geq 3$ bleek voor het derde en vierde orde filter zelfs nagenoeg geen enkele oplossing meer mogelijk.

gaan nu bekijken of we dit verschijnsel fysisch kunnen verklaren.

We beginnen met het tweede orde filter (fig. 1)

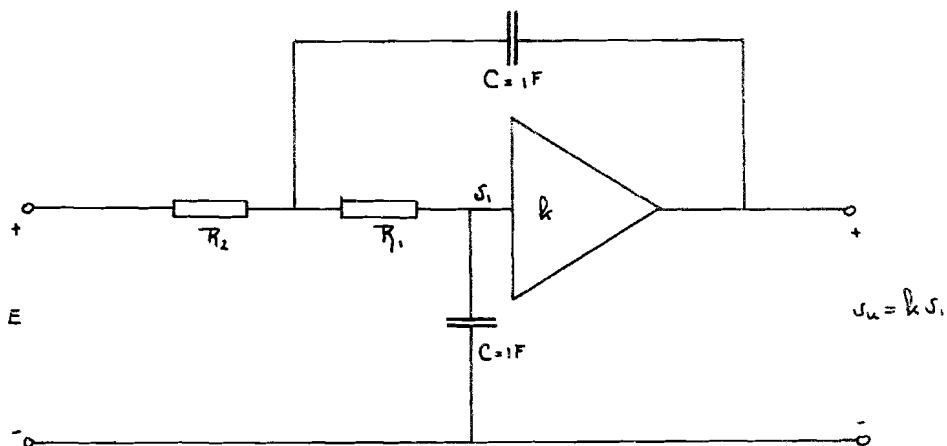
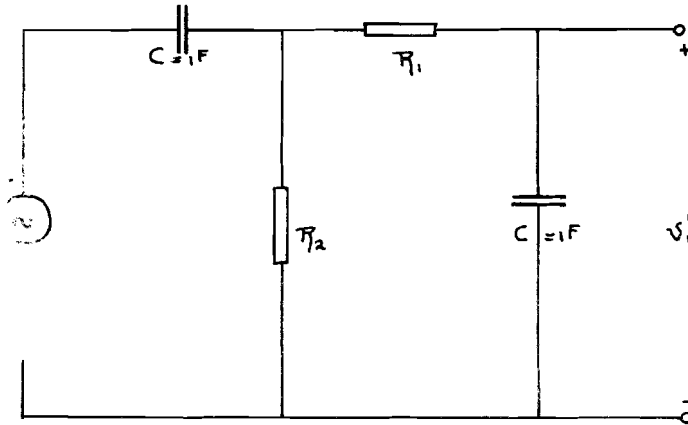


fig. 1: tweede orde Sallen en Key filter

ellen we de ingangsspanning E gelijk aan $E = 0$, dan kunnen we de rondgaande versterking voor de teruggekoppelde versterker bekijken (zie fig. 2).



g. 2: vervangingschema voor de bepaling van de rondgaande versterking.

∴ geldt:

$$V_1' = \frac{U_0}{2 + pR_1C + \frac{R_1}{R_2} + \frac{1}{pR_2C}} \quad (1)$$

De waarden van de weerstanden konden we, zoals in hoofdstuk 2.4 is afgeleid, als volgt berekenen uit de coëfficiënten a_2 en a_1 , die worden bepaald uit de ligging van de polen:

$$R_2 = \frac{a_2}{R_1} \quad (2)$$

$$R_1 = \frac{a_1}{2} \pm \frac{1}{2} \sqrt{a_1^2 - 4(2-k)a_2} \quad (3)$$

Uit deze formules zagen we, dat bij vasthouden van de pool-
ligging en ophogen van de versterkingsfactor vanaf de waarde
waarvoor realisatie mogelijk is, tot $k = 2$, één van de twee
oplossingen de volgende limiet heeft voor $k \rightarrow 2$:

$$U_1 \rightarrow 0$$

$$\rightarrow \infty$$

In de limiet geldt dan (zie fig. 2):

$$U_1' \rightarrow \frac{1}{2} \cdot U_0 = \frac{1}{2} \cdot 2 U_1 = U_1$$

Substitueren we dit in vergelijking (1) dan blijkt dat
voor deze oplossing de rondgaande versterking steeds toe-
neemt totdat zij voor $k = 2$ gelijk aan één is geworden:
er is nu precies aan de oscillatie voorwaarde voldaan
($U_1' = U_1$)

Voor alle frequenties is de rondgaande versterking nu gelijk
aan één, zodat er geen sprake meer is van een realisatie
van een bepaalde poolligging.

Beschouwing van het derde orde filter levert analoge resul-
taten op:

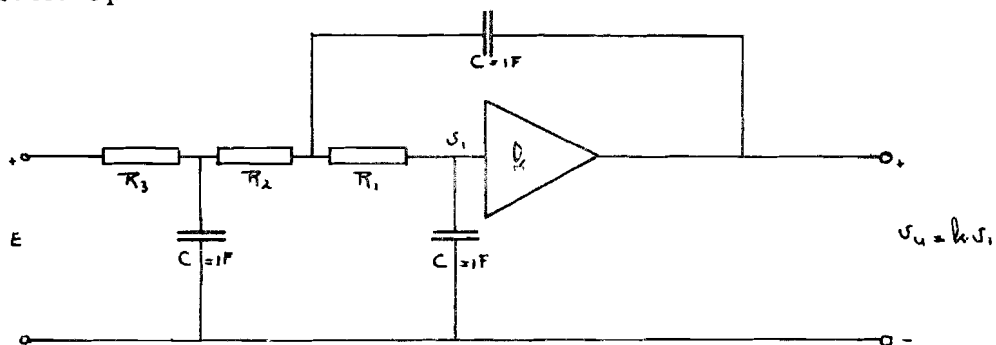


fig. 3 derde orde Sallen en Key filter.

voor de berekening van de rondgaande versterking vinden we nu (zie fig. 4):

$$U_i' = \frac{R_2 + \frac{R_3}{1 + pR_3C}}{\left\{ R_2 + \frac{R_3}{1 + pR_3C} \right\} \{ 2 + pR_1C \} + R_1 + \frac{1}{pC}} \cdot U_o \quad (4)$$

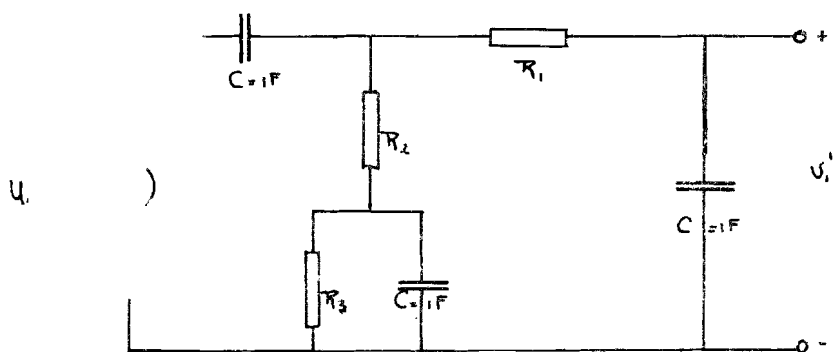


fig. 4: vervangingschema voor de bepaling van de rondgaande versterking voor het derde orde filter.

Wanneer we, vanaf de versterkingsfactor, waarvoor het beschouwde filter realiseerbaar is, de versterkingsfactor op-hogen, dan bleek dat voor één van de oplossingen geldt: voor $k \rightarrow 2$:

$$R_1 \rightarrow 0$$

$$R_3 \rightarrow \infty$$

dat in de limiet weer geldt:

$$U_i' \rightarrow \frac{1}{2} U_o = \frac{1}{2} \cdot 2 U_i = U_i$$

Voor $k = 2$ ontstaat voor deze oplossing dezelfde situatie als bij het tweede orde filter: de rondgaande versterking wordt weer gelijk aan één. (zie ook fig. 4)

Verder opvoeren van de versterkingsfactor en beschouwing van de andere oplossing leverde voor $k \rightarrow 3$:

$$\omega \rightarrow 0$$

$$\rightarrow 0$$

$$\rightarrow \infty$$

zien we uit fig 4 en door substitutie van bovenstaande in vergelijking (4):

$$U_i' \rightarrow \frac{1}{3} \cdot U_o = \frac{1}{3} \cdot 3U_i = U_i$$

Ook nu wordt de rondgaande versterking gelijk aan één voor alle frequenties.

De beschouwde poolligging is hiermede dan ook niet meer realiseerbaar.

Gaan we tot slot op dezelfde wijze te werk voor het vierde orde filter (fig. 5) dan verkrijgen we ook weer dezelfde resultaten.

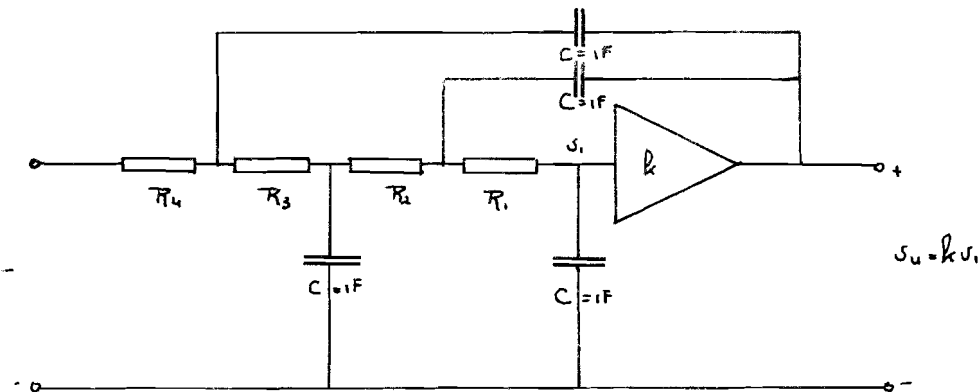


fig. 5: vierde orde Sallen en Key filter.

$$U_1' = \frac{p^2 R_2 R_3 C^2 + pC \left\{ 2R_2 + R_3 + \frac{R_2 R_3}{R_4} \right\} + \frac{R_3}{R_4} + \frac{R_2}{R_4}}{p^3 C^2 R_1 R_2 R_3 + p^3 C^2 \left\{ 2R_1 R_3 + 2R_2 R_3 + 2R_1 R_2 + \frac{R_1 R_2 R_3}{R_4} \right\} + pC \left\{ 3R_3 + 4R_4 + 3R_1 + \frac{2R_1 R_3}{R_4} + \frac{2R_2 R_3}{R_4} + \frac{R_1 R_2}{R_4} \right\} + \frac{3R_3}{R_4} + \frac{2R_2}{R_4} + \frac{R_1}{R_4} + 4 + \frac{1}{pR_4 C}} \cdot U_0 \quad (5)$$

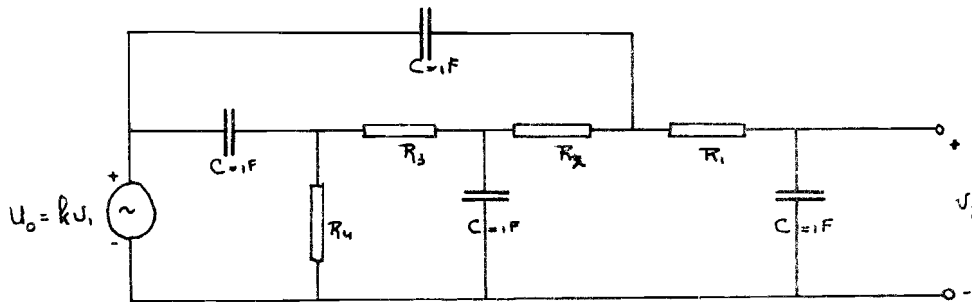


fig. 6: vervangingschema voor de bepaling van de rondgaande versterking van het vierde orde filter.

We zagen dat bij ophogen van de versterkingsfactor vanaf de waarde waarvoor het gewenste filter realiseerbaar is, voor $k \rightarrow 2$ één van de oplossingen naar de volgende limiet gaat:

$$\rightarrow 0$$

$$2 \rightarrow \infty$$

In het limiet geval geldt dan:

$$U_1' \rightarrow \frac{1}{2} U_0 = \frac{1}{2} \cdot 2 \cdot U_1 = U_1$$

Bij ophogen van de versterkingsfactor zien we uit fig. 6 dat de rondgaande versterking steeds toeneemt totdat deze in het limietgeval weer gelijk aan één wordt voor alle frequenties.

de schakeling oscilleert weer op alle frequenties en er is geen sprake van een realisering van een bepaalde poolligging meer.

Beschouwen we de tweede oplossing dan bleek bij nog verder verhogen van de versterkingsfactor tot $k = 3$ in de limiet te gelden:

$$\begin{aligned} \omega_1 &\rightarrow 0 \\ \omega_2 &\rightarrow 0 \\ \omega_3 &\rightarrow \infty \end{aligned}$$

We zien hiermee uit fig. 6 dat dan geldt:

$$U_1 \rightarrow \frac{1}{3} \cdot U_0 = \frac{1}{3} \cdot 3U_1 = U_1$$

meer wordt de rondgaande versterking gelijk aan één voor alle frequenties en komt de beschouwde oplossing te vervallen.

Conclusies en opmerkingen

In het voorgaande hebben we toch wel enige opmerkelijke resultaten verkregen.

We zagen dat voor het tweede orde filter, waarin alle componenten nog vrij te kiezen zijn, een versterkingsfactor

$= 1$ toereikend is om alle poolliggingen in het gehele

linkerhalfvlak te realiseren. Door alle capaciteiten gelijk

te kiezen kunnen nog slechts poolliggingen op de negatieve

reële as worden verwezenlijkt. Verhogen van de versterkings-

factor resulteert in een uitbreiding van het realiseerbare

gebied totdat voor $k = 2$ het gehele linkerhalfvlak en voor

$k > 2$ het hele complexe frequentievlak realiseerbaar is.

Gelijk kiezen van de weerstanden maakt voor $k = 1$ weer het

gehele linkerhalfvlak realiseerbaar.

Bij het derde orde filter is, voor $k = 1$ en voor willekeurig

te kiezen componenten, het gehele linkerhalfvlak realiseer-

baar. Kiezen we de weerstanden gelijk dan ontstaat er een

kleine verboden zone in het linkerhalfvlak vlak langs de

imaginair reële as. Gelijk kiezen van de condensatoren leidt tot

een beperking van het realiseerbare gebied tot de negatief

reële as; verhogen we in dit geval de versterkingsfactor dan

wordt het realiseerbare gebied weer groter, totdat voor $k = 2$

het gehele linkerhalfvlak te realiseren is. Voor versterkings-

factoren $k > 2$ blijft er een verboden gebied in het rechter-

halfvlak tot de versterkingsfactor $k = 3 - \epsilon$ (met $\epsilon \rightarrow 0$)

waarvoor het gehele complexe frequentievlak realiseerbaar is.

Voor $k > 3$ is er een klein realiseerbaar gebied dat groten-

deels in het rechterhalfvlak ligt en met toenemende verster-

kingsfactor steeds kleiner wordt.

Voor het vierde orde filter met gelijke capaciteiten is de

versterkingsfactor $k = 2$ niet meer toereikend om alle pool-
liggingen in het linkerhalfvlak te realiseren; tevens kunnen
er voor deze versterkingsfactor poolliggingen in het rechter-
halfvlak optreden.

Ook bij dit filter blijkt voor de versterkingsfactoren $k \geq 3$
nagenoeg geen enkele poolligging realiseerbaar.

Voor alle filters was er een bepaalde poolligging vanaf een
nageregrens voor de versterkingsfactor op twee (voor sommige
poolliggingen zelfs meer) manieren mogelijk. Voor $k = 2$
blijft één van de oplossingen. Voor het tweede orde filter
blijft realisatie verder op één manier mogelijk voor alle
versterkingsfactoren. Voor het derde en vierde orde filter
blijft de tweede oplossing slechts bestaan tot de versterkings-
factor $k = 3$; hierboven zijn nagenoeg geen poolliggingen
meer realiseerbaar.

het wegvallen van één oplossing bleek te ontstaan doordat de
nodige versterking van de teruggekoppelde versterker
niet mogelijk aan één wordt voor alle frequenties, zodat er dan geen
overblijfsel meer is van een gerealiseerde poolligging.

;

Literatuur opgave

1. Tirtoprodjo, S.
van Meer, A. "Forbidden regions for the poles of a 3rd order equal valued resistor active lowpass filter".
proc. IEE, vol. 119, no. 9, 1972
blz. 1281-1282
2. Tirtoprodjo, S. "Constraint removal for Huelsman's equal valued capacitor active RC circuit".
Electronic letters, 1971, vol. 7
blz. 448-449
3. Huelsman, L.P. "Equal valued capacitor active RC network realisation of a 3rd order lowpass Butterworth characteristic".
Electronic letters, 1971, vol. 7
blz. 271-272
4. Stephenson, F.W.
Dunning-Davies, J. "Recursive formulae for the analysis of double ladder single amplifier feedback systems".
Int. J. Electronics, 1975, vol. 39
blz. 625-633
5. Huelsman, L.P. "An equal valued capacitor active RC network realisation of a fourth order lowpass Butterworth characteristic".
IEEE proc., 1974, blz. 1709

6. Weinberg "Network analysis and synthesis".
Mc Graw Hill.
7. Dutta Roy, S.C. "Active RC realisation of a 3rd order
Malik, K.K. lowpass Butterworth characteristic".
Electronic Letters, 1972, vol. 8
blz. 630-631
- Tirtoprodjo, S. "Fourth order equal valued capacitor
Backx, A. active RC filters".
proc. 20th Midwest Symposium on Circuits
and Systems, part 1, 1977, blz 311-315
- Stephenson, F.W. "Equal capacitor Sallen en Key filters".
Int. J. Electronics, 1972, vol. 33, no. 5
blz. 539-544

bijlages:

FORTRAN 7700 BEATHE COMPILER, VERSION 2.0.000, MUNDAY, 06/27/

programma voor de berekening van grenstrommen voor het derde orde filter

T U N L I A N N E / I N T E R V A L H A L V
 = = = = =

```

'BEGIN' FILE 'INPUT', 'OUTPUT';
'REAL' MU, EPS, BETA, Y, MU1, MU2, MU3, VERSCHIL, VVERSCHIL, TM, T;
'INTEGER' K, I, J, N, L, M, Z, MMAX;
'LOGICAL' OPLUSSING, CONV, KLEINERINT, GECONVERG;
'INTEGER' 'ARRAY' LGECUNVERGEERD[1:4, 1:260], GECONVERGEERD[1:4, 1:260];
'INTEGER' 'ARRAY' K1[1:4, 1:260], K4[1:4, 1:260], R4VERSCH[1:4, 1:260];
'REAL' 'ARRAY' EDUKE 'INTERVALHALV'(X1, X2, Y, F1Y, F2Y, G1Y, G2Y, VERSCHIL, VVERSCHIL,
  KLEINERINT);
'CALL' EDUKE(X2, Y, F1Y, F2Y, VERSCHIL, VVERSCHIL, G1Y, G2Y;
  CONV, KLEINERINT);
  A=B+VVERSCHIL; BVERSCHIL=F1; F2=OVERSCHIL; WURTELK1=WURTELK2;
  K=TELLER;
  CONV=NIEGTGECUNV;
  IF(X<=AVERSCHIL:=VERSCHIL; BVERSCHIL:=VVERSCHIL; NIEGTGECUNV:=
  OE);
  (A+B)*.5; TELLER:=0; OVERSCHIL:=VERSCHIL; KLEINERINT:='FALSE';
  WURTELK1:=G1Y; WURTELK2:=G2Y; CONV:='TRUE';
  WURTELK1<0' OR 'WURTELK2<0' THEN
  IN 'KLEINERINT:='TRUE'; CONV:='FALSE'; TELLER:=250
  ');
  Y=0' THEN 'NIEGTGECUNV:='FALSE';
  (X1<=0' AND 'X2=0)' OR '(X2<=0' AND 'X1=0)' THEN
  IN 'KLEINERINT:=CONV:=NIEGTGECUNV:='FALSE'
  ');
  WHILE 'TELLER<250' AND 'NIEGTGECUNV' DO
  'BEGIN'
  Y:=(A+B)/2; WURTELK1:=G1Y; WURTELK2:=G2Y;
  'IF' 'WURTELK1<0' OR 'WURTELK2<0' THEN
  'BEGIN'
  TELLER:=250; KLEINERINT:='TRUE'; CONV:='FALSE'
  'END'
  'ELSE'
  'BEGIN'
  F1:=F1Y; F2:=F2Y; OVERSCHIL:=F1-F2; TELLER:=TELLER+1;
  'IF' 'SIGN(OVERSCHIL)*SIGN(AVERSCHIL)<=0' THEN
  'BEGIN'
  B:=Y; BVERSCHIL:=OVERSCHIL
  'END'
  'ELSE'
  'BEGIN'
  A:=Y; AVERSCHIL:=OVERSCHIL
  'END'
  'END';
  'IF' 'Y=0' THEN 'NIEGTGECUNV:='FALSE'
  'ELSE' 'BEGIN'
  'IF' 'ABS((A-B)/Y)<=10' THEN 'NIEGTGECUNV:='FALSE'
  'END'
  ');
  'CONV' THEN
  'BEGIN'
  'IF' 'Y=0' THEN 'CONV:='FALSE'
  
```

I

```

I
  'END' INTERVALHALV;
  'REAL' 'PROCEDURE' GEMINTERVALHALV(X1,X2,Y,G1Y);
  'REAL' X1,X2,Y,G1Y;
  'BEGIN'
    'REAL' A,B,C;
    'BOOLEAN' GCONV;
    'INTEGER' TELLER;
    A:=X1;B:=X2;Y:=X2;C:=G1Y;Y:=.5*(A+B);TELLER:=0;GCONV:='FALSE';
    'WHILE' 'NOT' GCONV 'AND' TELLER<250 'DO'
      'BEGIN'
        Y:=(A+B)/2;C:=G1Y;TELLER:=TELLER+1;
        'IF' C>0 'THEN'
          B:=Y
        'ELSE'
          A:=Y;
        'IF' Y'NEQ'0 'THEN'
          'BEGIN' 'IF' ABS((A-B)/Y)<=-10 'THEN' GCONV:='TRUE'
            'ELSE' GCONV:='FALSE'
          'END';
        'IF' Y=0 'AND' A=0 'AND' B=0 'THEN' GCONV:='TRUE'
      'END';

```

```

  'END' INTERVALHALV;
  'REAL' 'CONTROL' (MU, EPS, BETA, X, Y, Z, DPLVOLD);
  'REAL' X,Y,Z;
  'REAL' EPS,BETA,X,Y,Z;
  'BOOLEAN' DPLVOLD;
  'REAL' A1,A2,A3,B1,B2,B3;
  DPLVOLD:='TRUE';A1:=1+MU;A2:=EPS+MU;A3:=EPS;
  B1:=(2-BETA)*(Y+Z)+X+Z;B2:=(2-BETA)*Y*Z+2*X*Z+X*Y;B3:=X*Y*Z;
  'IF' ABS(2*(A1-B1)/(A1+B1))>=2 'THEN' DPLVOLD:='FALSE';
  'IF' ABS(2*(A2-B2)/(A2+B2))>=2 'THEN' DPLVOLD:='FALSE';
  'IF' ABS(2*(A3-B3)/(A3+B3))>=2 'THEN' DPLVOLD:='FALSE'
  'END' CONTROL;

```

```

  'PROCEDURE' WAARDETOEK(R1,R4,GCONVERGEERD,I,P,R41,R42,R4VERSCH,Y,
    GCONVERG,WURTEL);
  'REAL' 'ARRAY' R1[**,],R4[**],R4VERSCH[**];
  'BOOLEAN' 'ARRAY' GCONVERGEERD[**];
  'BOOLEAN' GCONVERG;
  'INTEGER' I,P;
  'REAL' R41,R42,Y,WURTEL;
  'BEGIN'
    'REAL' R4B,R2B;
    'BOOLEAN' DPLVOLD;
    R4B:=(R41+R42)/2;GCONVERG:='FALSE';
    'IF' Y>0 'AND' R4B>0 'AND' (Y*Y)>0 'THEN'
      'BEGIN'
        'IF' ABS((R41-R42)/R4B)<=(EPS+ABS(WURTEL/R4B)) 'THEN'
          'BEGIN'
            R4B:=Y*EPS/R4B;
            CONTROL(MU, EPS, BETA, Y, R2B, R4B/(Y*Y), DPLVOLD);
            'IF' DPLVOLD 'THEN'
              'BEGIN' R4B[R1[I],P]:=R4B;R1[I,P]:=Y;GCONVERGEERD[I,P]:='TRUE';
                R4VERSCH[I,P]:=R41-R42;P:=P+1;GCONVERG:='TRUE'
              'END'
            'END'
          'END'
        'END'
      'END'
    'END'
  'END' WAARDETOEK;

```



```

*PROCEDURE RIHEP(K1,R4,LGECONVERGEERD,GECONVERGEERD,Y,F1Y,F2Y,G1Y,G2Y);
*REAL* ARRAY K1(*,*),R4(*,*);
*BOOLEAN* ARRAY LGECONVERGEERD(*,*),GECONVERGEERD(*,*);
*REAL* Y,F1Y,F2Y,G1Y,G2Y;
*BEGIN
  *INTEGER* I,TEKEN1,TEKEN2,J,T,P,AANTALST,AANTALST2;
  *BOOLEAN* CONTINUEUPL1,CONTINUEUPL2,GEMINTERVH,WURTELNEG,NCUNTOPL,
    BEGNW,EINDENW,UPL,VCONTINUEUPL1,VCONTINUEUPL2,
    NUGLEINERINT;
  *REAL* WURTEL1,WURTEL2,R3,R41,R42,DELTA R1,RB,RB1,RA,DELTA R4,RK,
    VERSCHILK4,R1STAP;
  *INTEGER* ARRAY NEGWORTELS[1:2];
  BEGNW:='TRUE';EINDENW:='TRUE';
  *FOR* I:=1*STEP*1*UNTIL*4*DO
    *BEGIN
      *IF* I=1*THEN
        *BEGIN
          TEKEN1:=TEKEN2:=1;
          *FOR* J:=1*STEP*1*UNTIL*K*DO
            GECONVERGEERD[I,J]:='FALSE'
          *END*;
          *IF* I=2*THEN
            *BEGIN
              TEKEN1:=J;TEKEN2:=-1;
              *FOR* J:=1*STEP*1*UNTIL*K*DO
                GECONVERGEERD[I,J]:='FALSE'
              *END*;
            *IF* I=3*THEN
              *BEGIN
                TEKEN1:=-1;TEKEN2:=1;
                *FOR* J:=1*STEP*1*UNTIL*K*DO
                  GECONVERGEERD[I,J]:='FALSE'
                *END*;
            *IF* I=4*THEN
              *BEGIN
                TEKEN1:=TEKEN2:=-1;
                *FOR* J:=1*STEP*1*UNTIL*K*DO
                  GECONVERGEERD[I,J]:='FALSE'
                *END*;
            *IF* BETA=0*AND* MU=0*THEN* MU:=0-10;
            AANTALST1:=100;VERSCHIL:=VVERSCHIL:=0;AANTALST2:=10;
            *IF* BETA<=2*THEN* R1STAP:=(1+MU)/AANTALST;
            *IF* BETA>2*THEN
              *BEGIN
                *IF* ((EPS+MU)/(EPS*(BETA-2)))>=0*THEN
                  R1STAP:=(BETA-2)*EPS/(AANTALST*(-EPS-MU+(8*(BETA-2)*EPS*EPS
                    +(EPS+MU)**3)**(1/3)));
                *IF* ((EPS+MU)/(EPS*(BETA-2)))<0*THEN
                  *BEGIN
                    *IF* ((EPS+MU)**3)/((BETA-2)*EPS*EPS)<=54*THEN
                      R1STAP:=(EPS+MU)**2/(8*EPS*AANTALST)
                    *ELSE
                      R1STAP:=(BETA-2)*EPS/(AANTALST*(2*((BETA-2)*EPS*EPS)**
                        (1/3))-EPS-MU)
                    *END*
                  *END*
            *END*
            *IF* I>1;DELTA R1:=R1STAP;WURTELNEG:='FALSE';
            CONTINUEUPL1:=CONTINUEUPL2:='TRUE';NCUNTOPL:='FALSE';
            LEINERINT:='FALSE';
            *FOR* J:=0*STEP*1*UNTIL*AANTALST*DO

```

V

```

'BEGIN'
  'IF' 'KLEINERINT' THEN 'RB:=J*DELTAR1
  'ELSE' 'BEGIN'
    T:=T+1;RB:=RK+T*DELTAR1
  'END';
Y:=RB;WURTEL1:=G1Y;WURTEL2:=G2Y;
  'IF' 'WURTELNEG' THEN
    GEMINTERVH:='FALSE'
  'ELSE'
    GEMINTERVH:='TRUE';
  'IF' 'WURTEL1<0' OR 'WURTEL2<0' THEN
  'BEGIN'
    WURTELNEG:='TRUE';
    'IF' 'I=1' AND 'J=AANTALST' AND 'EINDENW' AND 'KLEINERINT' THEN
      NEGWURTELS[2]:=AANTALST;
    'IF' 'GEMINTERVH' AND 'I=1' AND 'BEGNW' AND 'KLEINERINT' THEN
    'BEGIN'
      NEGWURTELS[1]:=J;BEGNW:='FALSE'
    'END'
  'END'
  'ELSE'
  'BEGIN'
    WURTELNEG:='FALSE';
    'IF' 'GEMINTERVH' AND 'I=1' AND 'EINDENW' AND 'KLEINERINT' THEN
    'BEGIN'
      NEGWURTELS[2]:=J-1;EINDENW:='FALSE'
    'END'
  'END';
  'IF' 'WURTEL1>=0' AND 'WURTEL2>=0' THEN
  'BEGIN'
    Y:=PB;R41:=F1Y+TEKEN1*SQRT(WURTEL1);
    R42:=F2Y+TEKEN2*SQRT(WURTEL2);
    VERSCHIL:=R41-R42;VCONTINUEOPL1:=CONTINUEOPL1;
    VCONTINUEOPL2:=CONTINUEOPL2;
    'IF' 'CONTINUEOPL1' OR 'CONTINUEOPL2' THEN
    'BEGIN'
      NCNTUPL:='TRUE';VVERSCHIL:=VERSCHIL;
      'IF' 'CONTINUEOPL1' THEN
      'BEGIN'
        GEMINTERVALHALV(RB-DELTAR1,RB,Y,G1Y);
        WURTEL1:=SQRT(G1Y);
        WURTEL2:=G2Y;
        'IF' 'WURTEL2>=0' THEN
        'BEGIN'
          R42:=F2Y+TEKEN2*SQRT(WURTEL2);
          R41:=F1Y;
          WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,
            R4VERSCH,Y,GECONVERG,WURTEL1);
          'IF' 'GECONVERG' THEN
          'BEGIN'
            DELTAR4:=R41-R42;R41:=R41+TEKEN1*WURTEL1;
            VVERSCHIL:=R41-R42;RB1:=Y;
            'IF' 'SIGN(DELTAR4)*SIGN(VVERSCHIL)<=0' THEN
              WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,
                R42,R4VERSCH,Y,GECONVERG,0)
          'END'
        'END'
      'ELSE'
      'BEGIN'
        GEMINTERVALHALV(RB-DELTAR1,RB,Y,G2Y);

```

```

WURTEL2:=SQRT(G2Y);WURTEL1:=G1Y;
R41:=F1Y+TEKEN1*SQRT(WURTEL1);R42:=F2Y;
WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,
R4VERSCH,Y,GECONVERG,WURTEL2);
'IF'~GECONVERG'THEN'
'BEGIN'
DELTAR4:=R41-R42;R42:=R42+TEKEN2*WURTEL2;
VVERSCHIL:=R41-R42;RB1:=Y;
'IF'SIGN(DELTAR4)*SIGN(VVERSCHIL)<=0'THEN'
WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,
R4VERSCH,Y,GECONVERG,0)
'END'
'END'
'END';
'IF'~CONTINUEOPL2'THEN'
'BEGIN'
GEMINTERVALHALV(RB-DELTAR1,RB,Y,G2Y);
WURTEL2:=SQRT(G2Y);WURTEL1:=G1Y;
'IF'WURTEL1>=0'THEN'
'BEGIN'
R41:=F1Y+TEKEN1*SQRT(WURTEL1);R42:=F2Y;
WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,
R4VERSCH,Y,GECONVERG,WURTEL2);
'IF'~GECONVERG'THEN'
'BEGIN'
DELTAR4:=R41-R42;R42:=R42+TEKEN2*WURTEL2;
VVERSCHIL:=R41-R42;RB1:=Y;
'IF'SIGN(DELTAR4)*SIGN(VVERSCHIL)<=0'THEN'
WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,
R4VERSCH,Y,GECONVERG,0)
'END'
'END'
'ELSE'
'BEGIN'
GEMINTERVALHALV(RB-DELTAR1,RB,Y,G1Y);
WURTEL1:=SQRT(G1Y);WURTEL2:=G2Y;
R42:=F2Y+TEKEN2*SQRT(WURTEL2);R41:=F1Y;
WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,
R4VERSCH,Y,GECONVERG,WURTEL1);
'IF'~GECONVERG'THEN'
'BEGIN'
DELTAR4:=R41-R42;R41:=R41+TEKEN1*WURTEL1;
VVERSCHIL:=R41-R42;RB1:=Y;
'IF'SIGN(DELTAR4)*SIGN(VVERSCHIL)<=0'THEN'
WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,
R4VERSCH,Y,GECONVERG,0)
'END'
'END'
'END';
CONTINUEOPL1:=CONTINUEOPL2:='TRUE'
'END';
'IF'J>=1'THEN'
'BEGIN'
'IF'SIGN(VVERSCHIL)*SIGN(VVERSCHIL)<=0'THEN'
'BEGIN'
'IF'~CONTIOPL'THEN'~CONTIOPL:='FALSE'
'ELSE'RB1:=RB-DELTAR1;
'IF'KLEINERINT'THEN'~NOGKLEINERINT:='TRUE' 'ELSE'
~NOGKLEINERINT:='FALSE';
INTERVALHALV(RB,RB1,Y,F1Y+TEKEN1*SQRT(G1Y),F2Y+

```

```

        TEKEN2*SQRT(G2Y),G1Y,G2Y,VERSCHIL,VVERSCHIL,
        CONV,KLEINERINT);
'IF 'KLEINERINT' THEN 'NOGKLEINERINT:='TRUE'
'ELSE' 'BEGIN'
        'IF 'NOGKLEINERINT' THEN'
        'BEGIN'NOGKLEINERINT:='FALSE';KLEINERINT:=
        'TRUE'
        'END'
        'END';
'IF 'KLEINERINT' THEN'
'BEGIN'J:=J-1;
'IF 'NOGKLEINERINT' THEN'
'BEGIN'
        DELTAR1:=(RB-RB1)/AANTALST2;WORTELNEG:=~GEMINTERVH;
        RK:=RB:=RB1;CONTINUEOPL1:=VCONTINUEOPL1;
        CONTINUEOPL2:=VCONTINUEOPL2;NOGKLEINERINT:='FALSE';
        T:=0
'END'
'END';
'IF 'CONV' THEN'
'BEGIN'
        R41:=F1Y+TEKEN1*SQRT(G1Y);R42:=F2Y+TEKEN2*SQRT(
        G2Y);
        WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,
        R4VERSCH,Y,GECONVERG,U)
'END'
'END';
'END';
'IF '~KLEINERINT' OR 'T' NEO '0' THEN 'VVERSCHIL:=VERSCHIL
'END';
'IF 'WORTEL1<0' AND 'J>=1' AND 'GEMINTERVH' THEN'
'BEGIN'
        CONTINUEOPL1:='FALSE';
        GEMINTERVALHALV(RB,RB-DELTAR1,Y>G1Y),
        WORTEL1:=SQRT(G1Y);WORTEL2:=G2Y;
        'IF 'WORTEL2>=0' THEN'
        'BEGIN'
                R42:=F2Y+TEKEN2*SQRT(WORTEL2);R41:=F1Y;
                WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,R4VERSCH,
                Y,GECONVERG,WORTEL1);
                'IF '~GECONVERG' THEN'
                'BEGIN'
                        DELTAR4:=R41-R42;R41:=R41+TEKEN1*WORTEL1;
                        VERSCHIL:=R41-R42;
                        'IF 'SIGN(DELTAR4)*SIGN(VERSCHIL)<=0' THEN'
                        WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,R4VERSCH
                        ,Y,GECONVERG,U);
                        'IF 'SIGN(VERSCHIL)*SIGN(VVERSCHIL)<=0' THEN'
                        'BEGIN'
                                RB1:=RB-DELTAR1;RA:=Y;
                                INTERVALHALV(RA,RB1,Y,F1Y+TEKEN1*SQRT(G1Y),
                                F2Y+TEKEN2*SQRT(G2Y),G1Y,G2Y,VERSCHIL,
                                VVERSCHIL,CONV,KLEINERINT);
                                'IF 'CONV' THEN'
                                'BEGIN'
                                        R41:=F1Y+TEKEN1*SQRT(G1Y);R42:=F2Y+TEKEN2*
                                        SQRT(G2Y);
                                        WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,
                                        R4VERSCH,Y,GECONVERG,U)
                                'END'
                        'END'
                'END'
        'END'
'END'

```

```

        'END'
    'END'
'END'
'END';
'IF' 'WURTEL2<0' 'AND' 'J>=1' 'AND' 'GEMINTERVH' 'THEN'
'BEGIN'
    CONTINUEOPL2:='FALSE';
    GEMINTERVALHALV(RB,RB-DELTAR1,Y,G2Y);
    WURTEL2:=SQRT(G2Y);WURTEL1:=G1Y;
    'IF' 'WURTEL1>=0' 'THEN'
    'BEGIN'
        R41:=F1Y+TEKEN1*SQRT(WURTEL1);R42:=F2Y;
        WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,R4VERSCH,
            Y,GECONVERG,WURTEL2);
        'IF' 'GECONVERG' 'THEN'
        'BEGIN'
            DELTAR4:=R41-R42;R42:=R42+TEKEN2*WURTEL2;
            VERSCHIL:=R41-R42;
            'IF' 'SIGN(DELTAR4)*SIGN(VERSCHIL)<=0' 'THEN'
            WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,
                R4VERSCH,Y,GECONVERG,0);
            'IF' 'SIGN(VERSCHIL)*SIGN(VVERSCHIL)<=0' 'THEN'
            'BEGIN'
                RM1:=RB-DELTAR1;RA:=Y;
                INTERVALHALV(RA,RB1,Y,F1Y+TEKEN1*SQRT(G1Y),
                    F2Y+TEKEN2*SQRT(G2Y),G1Y,G2Y,VERSCHIL,
                    VVERSCHIL,CONV,KLEINERINT);
                'IF' 'CONV' 'THEN'
                'BEGIN'
                    R41:=F1Y+TEKEN1*SQRT(G1Y);R42:=F2Y+TEKEN2*
                        SQRT(G2Y);
                    WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,
                        R4VERSCH,Y,GECONVERG,0)
                'END'
            'END'
        'END'
    'END'
'END'
'END'
'END'
'END';
'IF' 'I>1' 'AND' 'WURTELNEG' 'THEN'
'BEGIN'
    'IF' 'J=NEGWURTELS[1]' 'AND' 'NEGWURTELS[2]>=NEGWURTELS[1]'
        'THEN' 'J:=NEGWURTELS[2]';
'END'
'END';
'IF' 'KLEINERINT' 'AND' 'T=AANTALST2' 'THEN'
'BEGIN' 'KLEINERINT:=' 'FALSE';DELTAR1:=R1STAP
'END'
'END';
:= 'FALSE';
=1;
'FOR' 'I:=1' 'STEP' '1' 'UNTIL' '4' 'DO'
    'IF' 'GECONVERGEERD[I,J]' 'THEN' 'OPL:=' 'TRUE';
'IF' 'OPL' 'THEN'
'BEGIN'
    'FOR' 'I:=1' 'STEP' '1' 'UNTIL' '4' 'DO'
        'FOR' 'J:=1' 'STEP' '1' 'UNTIL' 'K' 'DO'
            LGCONVERGEERD[I,J]:=GECONVERGEERD[I,J]
'END'
'END'
'END'
'END'

```

```

PROCEDURE MUINTERVALHALV(MU1,MU2,MU,R1,R4,GECONVERGEERD,LGECONVERGEERD,

```

```

Y,F1Y,F2Y,G1Y,G2Y);
REAL MU1,MU2,MU,Y,F1Y,F2Y,G1Y,G2Y;
REAL 'ARRAY'R1[*,*],R4[*,*];
BOOLEAN 'ARRAY'LGECONVERGEERD[*,*],LGECONVERGEERD[*,*];
BEGIN
  INTEGER I,J;
  BOOLEAN OPLUS,NGECONVERG;
  NGECONVERG:='TRUE';
  OPLUS:='NGECONVERG='DO'
  BEGIN
    MU:=(MU1+MU2)/2;
    IBEF(R1,R4,LGECONVERGEERD,GECUNVERGEERD,Y,F1Y,F2Y,G1Y,G2Y);
    OPLUS:='FALSE';
    O:=1;
    FOR I:=1 STEP 1 UNTIL 4 DO
      BEGIN
        IF GECUNVERGEERD[I,J] THEN
          OPLUS:='TRUE'
        END;
      IF OPLUS THEN
        MU1:=MU
      ELSE
        MU2:=MU;
        IF MU1#0 AND MU2#0 THEN
          BEGIN
            IF ABS(2*(MU1-MU2)/(MU1+MU2))<E-5 THEN NGECONVERG:='
            FALSE'
          END;
        ELSE
          BEGIN
            IF MU1=0 THEN
              BEGIN
                IF ABS(MU2)<E-6 THEN NGECONVERG:='FALSE'
              END;
            IF MU2=0 THEN
              BEGIN
                IF ABS(MU1)<E-6 THEN NGECONVERG:='FALSE'
              END;
            END;
          END;
        END;
      END;
    MU:=(MU1+MU2)/2
  ENDINTERVALHALV;
  PROCEDURE SCHRIFTERESULTATEN(R1,R4,R4VERSCH,LGECONVERGEERD,MU,EPS,Y,U1Y);
  MU,EPS,U1Y,Y;
  ARRAY R1[*,*],R4[*,*],R4VERSCH[*,*];
  AN ARRAY LGECUNVERGEERD[*,*];
  AL SIGMA,R1A,R4A,R2,R3,UMEGAKW,R4V;
  INTEGER I,J,G;
  SIGMA:=.5*MU/EPS;O:=1;
  UMEGAKW:=1/EPS-SIGMA*SIGMA;
  FOR I:=1 STEP 1 UNTIL 4 DO
    FOR J:=1 STEP 1 UNTIL 260 DO
      BEGIN
        IF LGECUNVERGEERD[I,J] THEN
          BEGIN
            R1A:=R1[I,J];Y:=R1A;IF U1Y#0 THEN
              BEGIN R4A:=R4[I,J]/U1Y;R4V:=R4VERSCH[I,J]/U1Y
            END;
            ELSE
              BEGIN R4A:=E60;R4V:=R4VERSCH[I,J]

```

1

VII

bijlage 2

BURROUGHS B7700 BEATHE COMPILER, VERSION 2.8.000, MONDAY, 06/27.

programma voor de berekening van grenskrommen voor het vierde orde filter

T O N L I A N N E / I N T E R V A L H A L V
 = = = = =

```
'BEGIN' 'FILE' INPUT, OUTPUT;
'REAL' MU, EPS, BETA, Y, MU1, MU2, MU3, VERSCHIL, VVERSCHIL, TM, T, WORTEL;
'INTEGER' K, I, J, N, L, M, Z;
'CAN' OPLOSSING, CONV, KLEINERINT, GECONVERG;
'CAN' 'ARRAY' LG, CONVERGEERD[1:4, 1:260], GECONVERGEERD[1:4, 1:260];
'ARRAY' R1[1:4, 1:260], R4[1:4, 1:260], R4VERSCH[1:4, 1:260];
'PROCEDURE' INTERVALHALV(X1, X2, Y, F1Y, F2Y, G1Y, G2Y, VERSCHIL, VVERSCHIL,
KLEINERINT);
  X1, X2, Y, F1Y, F2Y, VERSCHIL, VVERSCHIL, G1Y, G2Y;
  CONV, KLEINERINT;
  'REAL' A, B, AVERSCHIL, BVERSCHIL, F1, F2, CVERSCHIL, WORTELK1, WORTELK2;
  'INTEGER' TELLER;
  'BOOLEAN' NIETGECONV;
  A:=X1; B:=X2; AVERSCHIL:=VERSCHIL; BVERSCHIL:=VVERSCHIL; NIETGECONV:=
  'TRUE';
  Y:=(A+B)*.5; TELLER:=0; CVERSCHIL:=VERSCHIL; KLEINERINT:='FALSE';
  WORTELK1:=G1Y; WORTELK2:=G2Y; CONV:='TRUE';
  'IF' WORTELK1<0 'OR' WORTELK2<0 'THEN'
  'BEGIN' KLEINERINT:='TRUE'; CONV:='FALSE'; TELLER:=250
  'END';
  'IF' Y=0 'THEN' NIETGECONV:='FALSE';
  'IF' A=B 'THEN' NIETGECONV:='FALSE';
  'WHILE' TELLER<250 'AND' NIETGECONV 'DO'
  'BEGIN'
    Y:=(A+B)/2; WORTELK1:=G1Y; WORTELK2:=G2Y;
    'IF' WORTELK1<0 'OR' WORTELK2<0 'THEN'
    'BEGIN'
      TELLER:=250; KLEINERINT:='TRUE'; CONV:='FALSE'
    'END'
    'ELSE'
    'BEGIN'
      F1:=F1Y; F2:=F2Y; CVERSCHIL:=F1-F2; TELLER:=TELLER+1;
      'IF' SIGN(CVERSCHIL)*SIGN(AVERSCHIL)<=0 'THEN'
      'BEGIN'
        B:=Y; BVERSCHIL:=CVERSCHIL
      'END'
      'ELSE'
      'BEGIN'
        A:=Y; AVERSCHIL:=CVERSCHIL
      'END'
    'END';
    'IF' Y=0 'THEN' NIETGECONV:='FALSE'
    'ELSE' 'BEGIN'
      'IF' ABS((A-B)/Y)<=.10 'THEN' NIETGECONV:='FALSE'
    'END'
  'END';
  'IF' CONV 'THEN'
  'BEGIN'
    'IF' Y=0 'THEN' CONV:='FALSE'
  'END'
'END' INTERVALHALV;
```

I

```

REAL PROCEDURE GEMINTERVALHALV(X1,X2,Y,G1Y);
REAL X1,X2,Y,G1Y;
BEGIN
REAL A,B,C;
BOOLEAN GCONV;
INTEGER TELLER;
A:=X1;B:=X2;Y:=X2;C:=G1Y;Y:=.5*(A+B);TELLER:=0;GCONV:='FALSE';
IF A=B THEN GCONV:='TRUE';
WHILE NOT GCONV AND TELLER<250 DO
BEGIN
Y:=(A+B)/2;C:=G1Y;TELLER:=TELLER+1;
IF C>0 THEN
B:=Y
ELSE
A:=Y;
IF Y NEQ 0 THEN
BEGIN IF ABS((A-B)/Y)<@-10 THEN GCONV:='TRUE'
ELSE GCONV:='FALSE';
END;
IF Y=0 AND A=0 AND B=0 THEN GCONV:='TRUE';
END;
END GEMINTERVALHALV;

```

```

PROCEDURE CONTROLE(MU, EPS, BETA, X, Y, Z, T, OPLVOLD);
VALUE X, Y, Z, T;
REAL MU, EPS, BETA, X, Y, Z, T;
BOOLEAN OPLVOLD;
BEGIN
REAL A1, A2, A3, A4, a1, a2, a3, a4;
OPLVOLD:='TRUE';A4:=EPS;A3:=MU+BETA*EPS;A2:=1+EPS+BETA*MU;A1:=BETA+
MU;B4:=X*Y*Z*T;B3:=X*Y*Z+2*X*Y*T+2*X*Z*T;B2:=X*Y+2*X*Z+3*X*T+Z*T;
B1:=X+Z;
IF ABS(2*(A1-B1)/(A1+a1))>@-1 THEN OPLVOLD:='FALSE';
IF ABS(2*(A2-B2)/(A2+a2))>@-1 THEN OPLVOLD:='FALSE';
IF ABS(2*(A3-B3)/(A3+a3))>@-1 THEN OPLVOLD:='FALSE';
IF ABS(2*(A4-B4)/(A4+a4))>@-1 THEN OPLVOLD:='FALSE';
END;

```

```

PROCEDURE WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,R4VERSCH,Y,
ERG,WORTEL);
REAL R1[**],R4[**],R4VERSCH[**];
BOOLEAN GECONVERGEERD[**];
IN I,P;
REAL R42,Y,WORTEL;
BEGIN
REAL R4B,R3,R2B;
BOOLEAN OPLVOLD;
R4B:=(R41+R42)/2;GECONVERGEERD:=FALSE;R3:=BETA+MU-Y;
IF Y>0 AND R4B>0 AND R3>0 AND (R3#R3*Y)>0 THEN
BEGIN
IF ABS((R41-R42)/R4B)<(@-5+ABS(WORTEL/R4B)) THEN
BEGIN
R2B:=EPS*R3/R4B;
CONTROLE(MU, EPS, BETA, Y, R2B, R3, R4B/(R3*R3*Y), OPLVOLD);
IF OPLVOLD THEN
BEGIN R4[I,P]:=R4B;R1[I,P]:=Y;GECONVERGEERD[I,P]:='TRUE';
R4VERSCH[I,P]:=R41-R42;P:=P+1;GECONVERGEERD:=TRUE;
END;
END;
END;
END;

```

IV

```

'END' AARDETOEK;
PROCEDURE R1BEP(R1,R4,LGECONVERGEERD,GCONVERGEERD,Y,F1Y,F2Y,G1Y,G2Y);
  REAL ARRAY R1[*,*],R4[*,*];
  BOOLEAN ARRAY LGECONVERGEERD[*,*],GCONVERGEERD[*,*];
  REAL Y,F1Y,F2Y,G1Y,G2Y;
  BEGIN
    INTEGER I,TEKEN1,TEKEN2,J,T,P,AANTALST,AANTALST2;
    BOOLEAN CONTINUEOPL1,CONTINUEOPL2,GEMINTERVH,WURTELNEG,NCONTUPL,
      BEGNW,EINDENW,UPL,VCONTINUEOPL1,VCONTINUEOPL2,
      NOGKLEINERINT;
    REAL WURTEL1,WURTEL2,R3,R41,R42,DELTAR1,RB,RB1,RA,DELTAR4,RK,
      VERSCHILR4;
    INTEGER ARRAY NEGWURTELS[1:2];
    BEGNW:=TRUE;EINDENW:=TRUE;
    FOR I:=1 STEP 1 UNTIL 4 DO
      BEGIN
        IF I=1 THEN
          BEGIN
            TEKEN1:=TEKEN2:=1;
            FOR J:=1 STEP 1 UNTIL K DO
              GCONVERGEERD[I,J]:=FALSE;
          END;
        IF I=2 THEN
          BEGIN
            TEKEN1:=1;TEKEN2:=-1;
            FOR J:=1 STEP 1 UNTIL K DO
              GCONVERGEERD[I,J]:=FALSE;
          END;
        IF I=3 THEN
          BEGIN
            TEKEN1:=-1;TEKEN2:=1;
            FOR J:=1 STEP 1 UNTIL K DO
              GCONVERGEERD[I,J]:=FALSE;
          END;
        IF I=4 THEN
          BEGIN
            TEKEN1:=TEKEN2:=-1;
            FOR J:=1 STEP 1 UNTIL K DO
              GCONVERGEERD[I,J]:=FALSE;
          END;
        IF BETA=0 AND MU=0 THEN MU:=2-10;
        AANTALST:=100;VERSCHIL:=VVERSCHIL:=0;AANTALST2:=10;
        P:=1;DELTAR1:=(BETA+MU)/AANTALST;WURTELNEG:=FALSE;
        CONTINUEOPL1:=CONTINUEOPL2:=TRUE;NCONTUPL:=FALSE;
        KLEINERINT:=FALSE;
        FOR J:=0 STEP 1 UNTIL AANTALST DO
          BEGIN
            IF NOT KLEINERINT THEN RB:=J*DELTAR1
              ELSE BEGIN
                I:=T+1;RB:=RK+I*DELTAR1
              END;
            Y:=RB;WURTEL1:=G1Y;WURTEL2:=G2Y;
            IF WURTELNEG THEN
              GEMINTERVH:=FALSE;
            ELSE
              GEMINTERVH:=TRUE;
            IF WURTEL1<0 OR WURTEL2<0 THEN
              BEGIN
                WURTELNEG:=TRUE;
                IF I=1 AND J=AANTALST AND EINDENW AND NOT KLEINERINT THEN

```

V

```

      NEGWOEELS[2]:=AANTALST;
      'IF'~GEMINTERVH'AND'1=1'AND'BEGNW'AND'~KLEINERINT'THEN'
      'BEGIN'
        NEGWOEELS[1]:=J;BEGNW:='FALSE';
      'END'
    'END'
    'ELSE'
    'BEGIN'
      WOELNEG:='FALSE';
      'IF'~GEMINTERVH'AND'1=1'AND'EINDENW'AND'~KLEINERINT'THEN'
      'BEGIN'
        NEGWOEELS[2]:=J-1;EINDENW:='FALSE'
      'END'
    'END';
    'IF'WOEEL1>=0'AND'WOEEL2>=0'THEN'
    'BEGIN'
      Y:=RB;R41:=F1Y+TEKEN1*SQRT(WOEEL1);
      R42:=F2Y+TEKEN2*SQRT(WOEEL2);
      VERSCHIL:=R41-R42;VCONJINUEOPL1:=CONTINUEOPL1;
      VCONTINUEOPL2:=CONTINUEOPL2;
      'IF'~CONTINUEOPL1'OR'~CONTINUEOPL2'THEN'
      'BEGIN'
        NCONTOPL:='TRUE';VVERSCHIL:=VERSCHIL;
        'IF'~CONTINUEOPL1'THEN'
        'BEGIN'
          GEMINTERVALHALV(RB-DELTAR1,RB,Y,G1Y);
          WOEL1:=SQRT(G1Y);
          WOEL2:=G2Y;
          'IF'WOEL2>=0'THEN'
          'BEGIN'
            R42:=F2Y+TEKEN2*SQRT(WOEEL2);
            R41:=F1Y;
            WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,
              R4VERSCH,Y,GECUNVERG,WOEL1);
            'IF'~GECONVERG'THEN'
            'BEGIN'
              DELTAR4:=R41-R42;R41:=R41+TEKEN1*WOEL1;
              VVERSCHIL:=R41-R42;RB1:=Y;
              'IF'SIGN(DELTAR4)*SIGN(VVERSCHIL)<=0'THEN'
              'BEGIN'
                WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,
                  R42,R4VERSCH,Y,GECUNVERG,0)
              'END'
            'END'
          'END'
        'END'
      'ELSE'
      'BEGIN'
        GEMINTERVALHALV(RB-DELTAR1,RB,Y,G2Y);
        WOEL2:=SQRT(G2Y);WOEL1:=G1Y;
        R41:=F1Y+TEKEN1*SQRT(WOEEL1);R42:=F2Y;
        WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,
          R4VERSCH,Y,GECUNVERG,WOEL2);
        'IF'~GECONVERG'THEN'
        'BEGIN'
          DELTAR4:=R41-R42;R42:=R42+TEKEN2*WOEL2;
          VVERSCHIL:=R41-R42;RB1:=Y;
          'IF'SIGN(DELTAR4)*SIGN(VVERSCHIL)<=0'THEN'
          WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,
            R4VERSCH,Y,GECUNVERG,0)
        'END'
      'END'
    'END'
  'END'

```

```

        'END';
    'END';
    'IF' CONTINUEOPL2 THEN
    'BEGIN'
        GEMINTERVALHALV(RB=DELTAR1,RB*Y,G2Y);
        WORTEL2:=SQRT(G2Y);WURTEL1:=G1Y;
        'IF' WORTEL1>=0 THEN
        'BEGIN'
            R41:=F1Y+TEKEN1*SWRT(WORTEL1);R42:=F2Y;
            WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,
                R4VERSCH*Y,GECUNVERG,WORTEL2);
            'IF' GECONVERG THEN
            'BEGIN'
                DELTAR4:=R41-R42;R42:=R42+TEKEN2*WORTEL2;
                VVERSCHIL:=R41-R42;RB1:=Y;
                'IF' SIGN(DELTAR4)*SIGN(VVERSCHIL)<=0 THEN
                WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,
                    R4VERSCH*Y,GECUNVERG,0)
            'END';
        'END';
    'ELSE'
    'BEGIN'
        GEMINTERVALHALV(RB=DELTAR1,RB*Y,G1Y);
        WORTEL1:=SQRT(G1Y);WURTEL2:=G2Y;
        R42:=F2Y+TEKEN2*SWRT(WORTEL2);R41:=F1Y;
        WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,
            R4VERSCH*Y,GECUNVERG,WORTEL1);
        'IF' GECONVERG THEN
        'BEGIN'
            DELTAR4:=R41-R42;R41:=R41+TEKEN1*WORTEL1;
            VVERSCHIL:=R41-R42;RB1:=Y;
            'IF' SIGN(DELTAR4)*SIGN(VVERSCHIL)<=0 THEN
            WAARDETOEK(R1,R4,GECONVERGEERD,I,P,R41,R42,
                R4VERSCH*Y,GECUNVERG,0)
        'END';
    'END';
    'END';
    CONTINUEOPL1:=CONTINUEOPL2:=TRUE;
'END';
'IF' J>=1 THEN
'BEGIN'
    'IF' SIGN(VERSCHIL)*SIGN(VVERSCHIL)<=0 THEN
    'BEGIN'
        'IF' NCUNTOPL THEN NCUNTOPL:=FALSE;
        'ELSE' RB1:=RB-DELTAR4;
        'IF' KLEINERINT THEN NOGKLEINERINT:=TRUE 'ELSE'
        NOGKLEINERINT:=FALSE;
        INTERVALHALV(RB,RB1,Y,F1Y+TEKEN1*SQRT(G1Y),F2Y+
            TEKEN2*SQRT(G2Y),G1Y,G2Y,VERSCHIL,VVERSCHIL,
            CONV,KLEINERINT);
        'IF' KLEINERINT THEN NOGKLEINERINT:=TRUE;
        'ELSE' 'BEGIN'
            'IF' NOGKLEINERINT THEN
            'BEGIN' NOGKLEINERINT:=FALSE;KLEINERINT:=
                TRUE;
            'END';
        'END';
    'IF' KLEINERINT THEN
    'BEGIN' J:=J-1;
    'IF' NOGKLEINERINT THEN

```

```

'BEGIN'
  'IF' DELTAR1 > RB * 2 - 10 'THEN'
  'BEGIN'
    DELTAR1 := (RB - RB1) / AANTALST2; WORTELNEG := -GEMINTERVH;
    RK := RB1 - RB1; CONTINUEOPL1 := VCONTINUEOPL1;
    CONTINUEOPL2 := VCONTINUEOPL2; NOGKLEINERINT := 'FALSE';
    T := 0;
  'END';
  'ELSE' 'BEGIN' KLEINERINT := 'FALSE'; J := J + 1; DELTAR1 :=
    (BETA + MU) / AANTALST; NOGKLEINERINT := 'FALSE';
  'END';
'END';
'END';
'IF' CONV 'THEN'
'BEGIN'
  R41 := F1Y + TEKEN1 * SQRT(G1Y); R42 := F2Y + TEKEN2 * SQRT(
    G2Y);
  WAARDETOEK(R1, R4, GECONVERGEERD, I, P, R41, R42,
    R4VERSCH, Y, GECONVERG, 0)
'END';
'END';
'END';
'IF' 'KLEINERINT' OR 'T' NEQ '0' THEN VVERSCHIL := VERSCHIL
'END';
'IF' WORTEL1 < 0 AND 'J' >= 1 AND GEMINTERVH 'THEN'
'BEGIN'
  CONTINUEOPL1 := 'FALSE';
  GEMINTERVALHALV(RB, RB - DELTAR1, Y, G1Y);
  WORTEL1 := SQRT(G1Y); WORTEL2 := G2Y;
  'IF' WORTEL2 >= 0 'THEN'
  'BEGIN'
    R42 := F2Y + TEKEN2 * SQRT(WORTEL2); R41 := F1Y;
    WAARDETOEK(R1, R4, GECONVERGEERD, I, P, R41, R42, R4VERSCH,
      Y, GECONVERG, WORTEL1);
    'IF' GECONVERG 'THEN'
    'BEGIN'
      DELTAR4 := R41 - R42; R41 := R41 + TEKEN1 * WORTEL1;
      VERSCHIL := R41 - R42;
      'IF' SIGN(DELTA4) * SIGN(VERSCHIL) <= 0 'THEN'
      WAARDETOEK(R1, R4, GECONVERGEERD, I, P, R41, R42, R4VERSCH,
        Y, GECONVERG, 0);
      'IF' SIGN(VERSCHIL) * SIGN(VVERSCHIL) <= 0 'THEN'
      'BEGIN'
        R01 := RB - DELTAR1; R01 := Y;
        'IF' KLEINERINT 'THEN' NOGKLEINERINT := 'TRUE' 'ELSE'
        NOGKLEINERINT := 'FALSE';
        INTERVALHALV(KA, RB1, Y, F1Y + TEKEN1 * SQRT(G1Y),
          F2Y + TEKEN2 * SQRT(G2Y), G1Y, G2Y, VERSCHIL,
          VVERSCHIL, CONV, KLEINERINT);
        'IF' NOGKLEINERINT 'THEN' KLEINERINT := 'TRUE' 'ELSE'
        KLEINERINT := 'FALSE';
        NOGKLEINERINT := 'FALSE';
        'IF' CONV 'THEN'
        'BEGIN'
          R41 := F1Y + TEKEN1 * SQRT(G1Y); R42 := F2Y + TEKEN2 *
            SQRT(G2Y);
          WAARDETOEK(R1, R4, GECONVERGEERD, I, P, R41, R42,
            R4VERSCH, Y, GECONVERG, 0)
        'END';
      'END';
    'END';
  'END';
'END';

```

Y

```

      'END'
    'END'
  'END'
  'IF' WORTEL2 < 0 'AND' J >= 1 'AND' GEMINTERVH 'THEN'
  'BEGIN'
    CONTINGEOPL2 := 'FALSE';
    GEMINTERVALHALV(RB, RB * DELTAR1, Y, G2Y);
    WORTEL2 := SQRT(G2Y); WORTEL1 := G1Y;
    'IF' WORTEL1 >= 0 'THEN'
    'BEGIN'
      R41 := F1Y + TEKEN1 * SQRT(WORTEL1); R42 := F2Y;
      WAARDETOEK(R1, R4, GECONVERGEERD, I, P, R41, R42, RAVERSCH,
        Y, GECONVERG, WORTEL2);
      'IF' GECONVERG 'THEN'
      'BEGIN'
        DELTAR4 := R41 - R42; R42 := R42 + TEKEN2 * WORTEL2;
        VERSCHIL := R41 - R42;
        'IF' SIGN(DELTA4) * SIGN(VERSCHIL) <= 0 'THEN'
        WAARDETOEK(R1, R4, GECONVERGEERD, I, P, R41, R42,
          R4VERSCH, Y, GECONVERG, 0);
        'IF' SIGN(VERSCHIL) * SIGN(VVERSCHIL) <= 0 'THEN'
        'BEGIN'
          RB1 := RB - DELTAR1; RA := Y;
          'IF' KLEINERINT 'THEN' NOGKLEINERINT := 'TRUE' 'ELSE'
          NOGKLEINERINT := 'FALSE';
          INTERVALHALV(RA, RB1, Y, F1Y + TEKEN1 * SQRT(G1Y),
            F2Y + TEKEN2 * SQRT(G2Y), G1Y, G2Y, VERSCHIL,
            VVERSCHIL, CONV, KLEINERINT);
          'IF' NOGKLEINERINT 'THEN' KLEINERINT := 'TRUE' 'ELSE'
          KLEINERINT := 'FALSE';
          NOGKLEINERINT := 'FALSE';
          'IF' CONV 'THEN'
          'BEGIN'
            R41 := F1Y + TEKEN1 * SQRT(G1Y); R42 := F2Y + TEKEN2 *
              SQRT(G2Y);
            WAARDETOEK(R1, R4, GECONVERGEERD, I, P, R41, R42,
              R4VERSCH, Y, GECONVERG, 0)
          'END'
        'END'
      'END'
    'END'
  'END'
  'END'
  'IF' I > 1 'AND' WORTELNEG 'THEN'
  'BEGIN'
    'IF' J = NEGWORTELS[1] 'AND' NEGWORTELS[2] >= NEGWORTELS[1]
    'THEN' J := NEGWORTELS[2]
  'END'
'END'
'IF' KLEINERINT 'AND' T = AANTALST2 'THEN'
'BEGIN' KLEINERINT := 'FALSE'; DELTAR1 := (BETA + MU) / AANTALST
'END'
';
'FALSE';
:
I := 1 'STEP' 1 'UNTIL' 4 'DO'
  'IF' GECONVERGEERD(I, J) 'THEN' OPL := 'TRUE';
'IF' OPL 'THEN'
'BEGIN'
  'FOR' I := 1 'STEP' 1 'UNTIL' 4 'DO'
    'FOR' J := 1 'STEP' 1 'UNTIL' K 'DO'

```

V

LGECONVERGEERDI,J := GECONVERGEERDI,J

'END'

END'R1BEP;

PROCEDURE MUINTERVALHALV(MU1,MU2,MU,M1,R4,GECONVERGEERD,LGECONVERGEERD,

Y,F1Y,F2Y,G1Y,G2Y);

REAL'MU1,MU2,MU,Y,F1Y,F2Y,G1Y,G2Y;

REAL'ARRAY'R1[**,**],R4[**,**];

BOOLEAN'ARRAY'GECONVERGEERD[**,**],LGECONVERGEERD[**,**];

BEGIN

INTEGER'I,J;

BOOLEAN'OPLOSS,NGECONVERG;

REAL'MUA,MUB;

NGECONVERG := 'TRUE';

MUA := MU1; MUB := MU2;

WHILE'NGECONVERG'DO

BEGIN

MU := (MUA + MUB) / 2;

OPLOSS := 'FALSE';

IF'(MU + BETA) > 0 THEN

BEGIN

R1BEP(R1,R4,LGECONVERGEERD,GECONVERGEERD,Y,F1Y,F2Y,G1Y,G2Y);

J := 1;

FOR'I := 1 STEP 1 UNTIL 4 DO

BEGIN

IF'GECONVERGEERD[I,J] THEN

OPLOSS := 'TRUE';

END

END;

IF'OPLOSS THEN

MUA := MU

ELSE

MUB := MU;

IF'MUA # 0 AND MUB # 0 THEN

BEGIN

IF'ABS(2 * (MUA - MUB) / (MUA + MUB)) < e-5 THEN'NGECONVERG :=

'FALSE';

END

ELSE

BEGIN

IF'MUA = 0 THEN

BEGIN

IF'ABS(MUB) < e-6 THEN'NGECONVERG := 'FALSE';

END;

IF'MUB = 0 THEN

BEGIN

IF'ABS(MUA) < e-6 THEN'NGECONVERG := 'FALSE';

END

END

END;

MU := (MUA + MUB) / 2

MUINTERVALHALV;

PF

RE'SCHRYFRESULTATEN(R1,R4,R4VERSCH,LGECONVERGEERD,MU,EPS,Y,U1Y);

MU, EPS, U1Y, Y;

REAL'ARRAY'LGECONVERGEERD[**,**];

ARRAY'R1[**,**],R4[**,**],R4VERSCH[**,**];

V

VI

REAL'SIGMA,R1A,R4A,R2,R3,OMEGAKW,R4V;

INTEGER'I,J,Q;

SIGMA := .5 * ML / EPS; Q := 1;

OMEGAKW := 1 / EPS - SIGMA * SIGMA;


```

'FOR' I:=1 'STEP' 1 'UNTIL' 4 'DO'
  'FOR' J:=1 'STEP' 1 'UNTIL' 260 'DO'
    'BEGIN'
    'IF' LGECONVERGEERD[I,J] 'AND' R1[I,J] 'NEQ' 0 'THEN'
      'BEGIN'
        R1A:=R1[I,J]; Y:=R1A; 'IF' U1Y 'NEQ' 0 'THEN'
          'BEGIN' R4A:=R4[I,J]/U1Y; R4V:=R4VERSCH[I,J]/U1Y
          'END' 'ELSE'
            'BEGIN' R4A:=#60; R4V:=R4VERSCH[I,J]
            'END'
          'IF' R1A>0 'AND' R4A>0 'AND' R1A<(BETA+MU) 'THEN'
            'BEGIN'
              R3:=BETA+MU-R1A; R2:=EPS/(R1A+R3+R4A);
              'IF' OMEGAKW>0 'THEN'
                WRITE(OUTPUT, </> E11.4,X3, E11.4,X3, E11.4,X3, E11.4,X3,
                  E11.4,X3, E11.4,X3, E11.4,X5,
                  E9.2,X5, E9.2,X3, I2>> R1A, R2, R3, R4A, R4V
                  MU, EPS
                  SIGMA, SQRT(OMEGAKW), Q);
              'IF' OMEGAKW<=0 'THEN'
                WRITE(OUTPUT, </> E11.4,X3, E11.4,X3, E11.4,X3, E11.4,X3,
                  E11.4,X3, E11.4,X3, E11.4,"---*
                  "-E9.2,"---*"-E9.2,X3, I2>> R1A, R2, R3, R4A, R4V, MU, EPS,
                  SIGMA+SQRT(-OMEGAKW), SIGMA-SQRT(-OMEGAKW), Q);
              Q:=Q+1
            'END'
          'ELSE'
            'BEGIN'
              R3:=BETA+MU-R1A;
              'IF' OMEGAKW>0 'THEN'
                WRITE(OUTPUT, </> E11.4,X3, "ONEINDIG", X4, E11.4,X3, E11.4,
                  X3, E11.4,X3, E11.4,X3, E11.4,
                  X5, E9.2,X5, E9.2,X3, I2>> R1A, R3, R4A, R4V,
                  MU, EPS, SIGMA, SQRT(OMEGAKW), Q);
              'IF' OMEGAKW<=0 'THEN'
                WRITE(OUTPUT, </> E11.4,X3, "ONEINDIG", X4, E11.4,X3, E11.4,
                  X3, E11.4,X3, E11.4,X3, E11.4,
                  "---*"-E9.2,"---*"-E9.2,X3, I2>> R1A, R3, R4A, R4V, MU, EPS
                  SIGMA+SQRT(-OMEGAKW), SIGMA-SQRT(-OMEGAKW), Q);
              Q:=Q+1
            'END'
          'END'
        'END'
      'END'

```

SCHRYFRESULTATEN

```

RE MUSTAP(R1, R4, R4VERSCH, GECONVERGEERD, LGECONVERGEERD, Y, MU, MU1,
  IY, F2Y, G1Y, G2Y, U1Y);

```

```

Y, MU, MU1, MU2, F1Y, F2Y, G1Y, G2Y, U1Y;
'ARRAY' R1[*,*], R4[*,*], R4VERSCH[*,*];

```

```

'BOOLEAN' 'ARRAY' GECONVERGEERD[*,*], LGECONVERGEERD[*,*];

```

```

'BEGIN' 'REAL' MUMAX, MUSJAP, MUB;

```

```

'INTEGER' I, J;

```

```

'BOOLEAN' OPL, GECONV1, GECONV2, TEKENW;

```

```

MU1:=MUB:=MU;

```

```

OPL:='FALSE';

```

```

'IF' (MU+BETA)>0 'THEN'

```

```

'BEGIN'

```

```

R1BEP(R1, R4, LGECONVERGEERD, GECONVERGEERD, Y, F1Y, F2Y, G1Y, G2Y);

```

```

'FOR' J:=1 'STEP' 1 'UNTIL' 4 'DO'

```

```

'BEGIN' 'IF' GECONVERGEERD[J,1] 'THEN' OPL:='TRUE'

```

```

'END';

```

VII

PR

VIII

```

'IF' OPL 'THEN' WRITE(OUTPUT, </> "DUBBELE-NEG->REELE-POOL-VOLDOET.")
'ELSE' WRITE(OUTPUT, </> "DUBBELE-NEG->REELE-POOL-VOLDOET-NIET.")
'END';
GECONV1:=OPL;
'IF' BETA>MU 'THEN' MUMAX:=-MU; 'ELSE' MUMAX:=-BETA;
MUSTAP:=(MUMAX-MU1)/10;
TEKENW:='FALSE';
'FOR' I:=1 'STEP' 1 'UNTIL' 10 'DO';
'BEGIN' MU2:=MUB+I*MLSTAP; MU:=MU2;
OPL:='FALSE';
'IF' (MU+BETA)>0 'THEN';
'BEGIN'
R1BEP(R1,R4,LGECONVERGEERD,GECONVERGEERD,Y,F1Y,F2Y,G1Y,G2Y);
'FOR' J:=1 'STEP' 1 'UNTIL' 4 'DO';
'BEGIN' 'IF' GECONVERGEERD(J,1) 'THEN' OPL:='TRUE';
'END';
'END';
GECONV2:=OPL;
'IF' (GECONV1 'AND' GECONV2) 'OR' (~GECONV1 'AND' GECONV2) 'THEN';
'BEGIN' TEKENW:='TRUE';
'IF' GECONV1 'THEN';
MUIINTERVALHALV(MU1,MU2,MU,R1,R4,GECONVERGEERD,LGECONVER
GEERD,Y,F1Y,F2Y,G1Y,G2Y)
'ELSE';
MUIINTERVALHALV(MU2,MU1,MU,R1,R4,GECONVERGEERD,LGECONVER
GEERD,Y,F1Y,F2Y,G1Y,G2Y);
SCHRYFRESULTA(IEN(R1,R4,R4VERSCH,LGECONVERGEERD,MU,EPS,Y,U1Y)
'END';
GECONV1:=GECONV2; MU1:=MU2;
'END';
'IF' ~TEKENW 'THEN';
'BEGIN' 'IF' GECONV1 'THEN';
'BEGIN' WRITE(OUTPUT, </> "VOOR-DEZE-STRAAL-(R=">E11.4)>)-ZYN-
ALLE-OPL-MOGELYK">>SQRT(1/EPS));
SCHRYFRESULTA(IEN(R1,R4,R4VERSCH,LGECONVERGEERD,MU,EPS,Y,U1Y)
WRITE(OUTPUT, <"***">);
'END';
'ELSE';
WRITE(OUTPUT, </> "VOOR-DEZE-STRAAL-(R=">E11.4)>)-IS-GEEN-
OPL-MOGELYK.">>SQRT(1/EPS));
'END';
'END' MUSTAP;
READ(INPUT, /, L);
'FOR' Z:=0 'STEP' 1 'UNTIL' 20 'DO';
'BEGIN';
BETA:=.1*Z;
K:=250;
WRITE(OUTPUT(SKIP(1)));
WRITE(OUTPUT, </> "BETA=">F6.3>>BETA);
WRITE(OUTPUT, </> " *-VOR-EEN-OPLOSSING-VAN-SIGMA-EN-OMEGA-BETEKEN-
-DE-POLEN-REEL->YN">);
WRITE(OUTPUT, </> "X9,">R1">,">X12,">R2">,">X12,">R3">,">X12,">R4">,">X4,">R4VERSCHIL",
">MU">,">X11,">EPS">,">X9,">SIGMA">,">X9,"
">OMEGA">,">X3,">OPL.>);
'FOR' I:=0 'STEP' 1 'UNTIL' L 'DO';
'BEGIN';
TM:=10**N;
'FOR' M:=1 'STEP' 1 'UNTIL' 9 'DO';
'BEGIN';
'FOR' I:=1 'STEP' 1 'UNTIL' 4 'DO';

```

VIII

```
'FOR' J:=1 'STEP' 1 'UNTIL' 260 'DO'  
  LGCONVERGEERDI, J:= 'FALSE'  
  T:=.1*M;
```

```
EPS:=1/(.1*T*TM)/MU:=2*EPS; EPS:=EPS*EPS;  
MUSTAP(R1,R4,R4VERSCH,GECONVERGEERD,LGECONVERGEERD,Y,MU,MU1,  
MU2,((MU+BETA*EPS)*(MU+BETA-Y)-2*EPS)/4,  
(Y*((MU+BETA-Y)**2)/(2*(MU+BETA+2*Y)))*(-2*Y*(MU+BETA-Y)+1+  
EPS+BETA*MU),((MU+BETA*EPS)*(MU+BETA-Y)-2*EPS)**2-  
8*Y*((MU+BETA-Y)**3)*EPS)/16,((Y*(MU+BETA-Y))**2)  
*(MU+BETA*Y)*((MU+BETA-Y)*((1+EPS+BETA*MU-2*Y*(MU+BETA-Y))  
**2)/(4*(MU+BETA+2*Y)**2)-EPS/(MU+BETA+2*Y)),((MU+BETA-Y)  
**2)*Y)
```

```
'END'
```

```
'END'
```

```
'END'
```

```
'END'
```

```
=====  
ERRORS DETECTED = 0.
```

```
SEGMENTS = 14. TOTAL SEGMENT SIZE = 1535 WORDS. CORE ESTIMATE = 3236 WORDS:
```

```
= 608 CARDS, 4221 SYNTACTIC ITEMS, 61 DISK SEGMENTS:
```

```
NAME: (U2086S140)TONLIANNE/INTERVALHALV: 87700 CODE GENERATED:
```

```
TIME = 19.805 SECONDS ELAPSED; 3.471 SECONDS PROCESSING; 10.672 SECONDS I/O  
=====
```

bijlage

T O N L I A N N E / I N T E R V A L H A L V
 = = = = =

programma voor de berekening van nomogrammen voor het vierde orde filter.

```

REGIN 'FILE' INPUT, OUTPUT;
  'REAL' R1, R2, R3, B, C, R3STAP, DELTA R2, REEN, RTWEE, RDRIE, RVIER, K, R20, ALPHA,
    BETA, BCONTR, CCONTR, A, ACONTR;
  'BOOLEAN' OPL, GECONV;
  'INTEGER' I, J, N;
REAL 'PROCEDURE' RIBEP (X2, X3, B, C, R1A, R1B, COMPLEXER1, A);
REAL X2, X3, B, C, R1A, R1B, A;
BOOLEAN COMPLEXER1;
REGIN
  'REAL' WORTEL, AA, BB, CC;
  COMPLEXER1 := 'FALSE';
  AA := -X2*X2*X3*X3*(X2+X3);
  BB := -2*A*X2*X2+X3*(-2*A*X3+X2*(-5*A+(X2+X3)*(B+A*C)));
  CC := A*(2*(B+A*C)-4*A*(1/X2+1/X3)+X2*X3*(X3-B-C));
  'IF' (BB*BB-4*AA*CC) < 0 'THEN' COMPLEXER1 := 'TRUE'
  'ELSE'
  'BEGIN'
    WORTEL := SQRT (BB*BB-4*AA*CC);
    R1A := (-BB+WORTEL)/(2*AA);
    R1B := (-BB-WORTEL)/(2*AA)
  'END'
END RIBEP;
REAL 'PROCEDURE' BETAB (R1C, X2, X3):
REAL R1C, X2, X3;
REGIN
  BETAB := (A/X2+2*A/X3+R1C*X2*X3)*(2*A*X2-X2*X3*C*(A-1)+X3*(2*A+X2*
    (-R1C+X3*(X2*R1C-1))))/
    (2*A+X2*X3*(-A+R1C*(X2+X3)))+(A/R1C+3*A/X3)/X2+R1C*(2*X3+X2)-1-A
END BETAB;
REAL 'PROCEDURE' VOLGENDER1 (R11, R12, X1, R1C, NEGR1R2);
REAL R11, R12, X1, R1C;
BOOLEAN NEGR1R2;
REGIN
  'BOOLEAN' POSR1, POSR2;
  'BEGIN'
    POSR1 := POSR2 := NEGR1R2 := 'FALSE';
    'IF' R11 >= 0 'THEN' POSR1 := 'TRUE';
    'IF' R12 >= 0 'THEN' POSR2 := 'TRUE';
    'IF' POSR1 'AND' POSR2 'THEN'
    'BEGIN'
      'IF' ABS (R11-X1) <= ABS (R12-X1) 'THEN' R1C := R11
      'ELSE' R1C := R12
    'END'
    'ELSE'
    'BEGIN'
      'IF' POSR1 'THEN' R1C := R11;
      'IF' POSR2 'THEN' R1C := R12
    'END';
    'IF' ~POSR1 'AND' ~POSR2 'THEN'
    'BEGIN'
      WRITE (OUTPUT, </, "ER_IS_GEEN_POS._R1:R1A=", E11.4, /, X19,

```

```

        "R18=",E11.4>,R11,R12);
    NTEGR1R2:='TRUE'
III  'END'
    'END'
'END'VOLGENDER1;
REAL'PROCEDURE'R3INTERVALHALV(X3,RSTAPR3,X2,GECONVERGEERD,VERSCHIL,
VVERSCHIL,R1C,X1);
REAL'X2,X3,RSTAPR3,VERSCHIL,VVERSCHIL,R1C,X1;
BOOLEAN'GECONVERGEERD;
BEGIN
    REAL'R31,R32,R3STAP,R3B,R1A,R1B;
    BOOLEAN'COMPLR1,NEGR1R2;
    BEGIN
        R31:=X3;R32:=X3+RSTAPR3;R3STAP:=RSTAPR3/2;
        GECONVERGEERD:=COMPLR1:=NEGR1R2:='FALSE';
        WHILE'~GECONVERGEERD'AND'~COMPLR1'AND'~NEGR1R2'DO'
        BEGIN
            R3B:=R31+R3STAP;
            R1BEP(X2,R3B,B,C,R1A,R1B,COMPLR1,A);
            IF'COMPLR1'THEN'
                WRITE(OUTPUT,</,"VOOR_DEZE_R3(=",E11.4,")_EN_R2(=",
                    "*)_IS_GEEN_OPL._MOGELYK(COMPLEXE_R1)">,R3B,X2)
            ELSE'
                BEGIN
                    VOLGENDER1(R1A,R1B,X1,R1C,NEGR1R2);
                    VERSCHIL:=BETAB(R1C,X2,R3B)-B*C;
                    IF'VERSCHIL*VVERSCHIL<0'THEN'
                        R32:=R3B
                    ELSE'R31:=R3B;
                    R3STAP:=R3STAP/2;
                    IF'ABS((R31-R32)*2/(R31+R32))<=@-10
                        THEN'GECONVERGEERD:='TRUE';
                    VVERSCHIL:=VERSCHIL
                END'
            END'
        END';
        X3:=R3B
    END'
'END'R3INTERVALHALV;
REAL'PROCEDURE'CPLEBEP(Y1,Y2,Y3,B,C,REEN,RTWEE,RDRIE,RVIER,K,GECONV,
R3STAP,A);
REAL'Y1,Y2,Y3,B,C,R3STAP,REEN,RTWEE,RDRIE,RVIER,K,A;
BOOLEAN'GECONV;
BEGIN
    REAL'VERSCHIL,VVERSCHIL,R3A,R1C,R1A,R1B;
    BOOLEAN'NOGGEENTEKENW,GECONVERGEERD,POSR3,POSR1,COMPLEXER1,NEGR1R2;
    INTEGER'I,TELLER;
    BEGIN
        R1BEP(Y2,Y3,B,C,R1A,R1B,COMPLEXER1,A);GECONV:='FALSE';
        IF'COMPLEXER1'THEN'
            WRITE(OUTPUT,</,"VOOR_DEZE_R3(=",E11.4,")_EN_R2(=",E11.4,
                "*)_IS_GEEN_OPL._MOGELYK(COMPLEXE_R1).">,Y3,Y2)
            ELSE'
                BEGIN
                    VOLGENDER1(R1A,R1B,Y1,R1C,NEGR1R2);
                    POSR1:=~NEGR1R2;
                    VVERSCHIL:=BETAB(R1C,Y2,Y3)-B*C;
                    NOGGEENTEKENW:='TRUE';R3A:=Y3;COMPLEXER1:='FALSE';
                    POSR3:='TRUE';TELLER:=0;
                    WHILE'NOGGEENTEKENW'AND'~COMPLEXER1'AND'POSR1'AND'POSR3'AND'

```

```

    TELLER<250'DO'
  'BEGIN'
    R3A:=R3A-R3STAP;
    'IF'R3A<0'THEN'POSR3:='FALSE';
    TELLER:=TELLER+1;
    R1BEP(Y2,R3A,B,C,R1A,R1B,COMPLEXER1,A);
    'IF'COMPLEXER1'THEN'
      WRITE(OUTPUT,</,"VOOR_DEZE_R3(=",E11.4,")_EN_R2(=",E11.4,
        ")_IS_GEEN_OPL._MOGELYK(COMPLEXE_R1.).">,R3A,Y2)
    'ELSE'
      'BEGIN'
        VOLGENDER1(R1A,R1B,Y1,R1C,NEGR1R2);
        POSR1:=~NEGR1R2;
        VERSCHIL:=BETAB(R1C,Y2,R3A)-B*C;
        'IF'VERSCHIL*VVERSCHIL<0'THEN'NOGGEENTEKENW:='F/
        'ELSE''IF'ABS(VERSCHIL)>ABS(VVERSCHIL)'THEN'R3S'
          -R3STAP;
        VVERSCHIL:=VERSCHIL
      'END'
    'END':
    'IF'~NOGGEENTEKENW'THEN'
      'BEGIN'
        R3INTERVALHALV(R3A,R3STAP,Y2,GECONVERGEERD,VERSCHIL,
          VVERSCHIL,R1C,Y1);
        'IF'GECONVERGEERD'THEN'
          'BEGIN'
            REEN:=R1C;RDRIE:=R3A;RTWEE:=Y2;
            RVIER:=A/(Y2*R3A*R1C);
            K:=(2-R1C*(2*A*R3A+Y2*(2*A+R3A*(C*(1-A)-R3A+R1C*
              (Y2*R3A-1)))))/(2*A+Y2*R3A*(-A+R1C*(Y2+R3A)));
            GECONV:='TRUE'
          'END'
        'END'
      'END'
    'END'
  'END'
  'ENC'OPLBEP;
  'BEGIN'
  'FOR'I:=1'STEP'1'UNTIL'7'DO'
  'BEGIN'
    READ(INPUT,</,A,B,C,R1,R2,R3,DELTAR2,R3STAP,N);
    R20:=R2;
    WRITE(OUTPUT,</,"VIERDE_ORDE_NOMOGRAM_VOOR_(A*P*P+B*P+1)*
      (P*P+C*P+1)",</,</,"DE_INGELEZEN_WAARDEN_ZYN:",</,"A=",E11.4,</,
      "B=",E11.4,</,"C
      =",E11.4,</,"R1=",E11.4,</,"R2=",E11.4,</,"R3=",E11.4,</,"R3STAP=",
      E11.4,</,"R2STAP=",E11.4,</,"N=",I4,</,</,</>,A,B,C,R1,R2,R3,R3STAP,
      DELTAR2,N);
    WRITE(OUTPUT,</,X5,"R1",X12,"R2",X12,"R3",X12,"R4",X14,"K",X15,
      "ACONTROLE",X5,"BCONTROLE",X5,"CCONTROLE",</,</,</>);
    'FOR'J:=1'STEP'1'UNTIL'N'DO'
    'BEGIN'
      R2:=R20+J*DELTAR2;
      OPLBEP(R1,R2,R3,B,C,REEN,RTWEE,RDRIE,RVIER,K,GECONV,R3STAP,A);
      OPL:='FALSE';
      'BEGIN'
        'IF'GECONV'THEN'
          'BEGIN'
            ALPHA:=(2-K)*(2*RVIER+RTWEE+RDRIE)+REEN+RDRIE;
            BETA:=(RVIER*(RDRIE+2*RTWEE)+RTWEE*RDRIE)*(2-K)+

```

```

RVIER*(RDRIE+3*REEN)+REEN*(RTWEE+2*RDRIE)-1-A;
BCCNTR:=(ALPHA+SQRT(ALPHA*ALPHA-4*BETA))/2;
CCCNTR:=ALPHA-BCCNTR;
ACCNTR:=REEN*RTWEE*RDRIE*RVIER;
WRITE(OUTPUT,</,E11.4,X3,E11.4,X3,F11.4,X3,E11.4,X3,
      E11.4,X3,I4,X3,E11.4,X3,E11.4,X3,E11.4>,REEN,RTWEE,
      RDRIE,RVIER,
      K,J,ACCNTR,BCCNTR,CCCNTR);
OPL:='TRUE';R3:=RDRIE;R1:=REEN

```

```

'END';
'IF'~OPL'THEN'
'BEGIN'
  WRITE(OUTPUT,</,"VOOR_DEZE_R2(=",E11.4,")_IS_
        OPL._MOGELYK.">,R2);J:=N

```

```
'END'
```

```
'END'
```

```
'END'
```

```
'END'
```

```
'END'
```

```
'END'
```

```

=====
RS DETECTED = 0.
ENTS = 9. TOTAL SEGMENT SIZE = 602 WORDS. CORE ESTIMATE = 1989 WORDS. STAC
  189 CARDS. 1334 SYNTACTIC ITEMS. 27 DISK SEGMENTS.
AME: (U2086S140)TONLIANNE/INTERVALHALV. 87700 CODE GENERATED.
ME = 29.214 SECONDS ELAPSED; 1.404 SECONDS PROCESSING; 3.694 SECONDS I/O.
=====

```

BIJLAGE 4 Enige Numerieke Beschouwingen.

Als we numeriek snijpunten van twee functies en de bijbehorende functiewaarden willen bepalen, zullen we deze waarden nooit exact kunnen vinden door de fouten die we maken, doordat we niet alle reële getallen tot onze beschikking hebben om mee te werken, maar slechts een bepaald aantal getallen uit deze oneindige verzameling.

zullen nu formules af gaan leiden, waaruit we een bovengrens voor de grootte van de gemaakte fout kunnen berekenen.

Wel dat we het punt $X = X_0$ willen bepalen waar de twee functies $f(X)$ en $g(X)$ aan elkaar gelijk zijn:

$$Y_0 = f(X_0) = g(X_0)$$

dan kunnen we, bij numerieke bepaling van dit punt, als volgt de fout in de gevonden waarden voor X_0 en bijbehorende $Y_0 = f(X_0) = g(X_0)$ berekenen. (zie ook fig. 1)

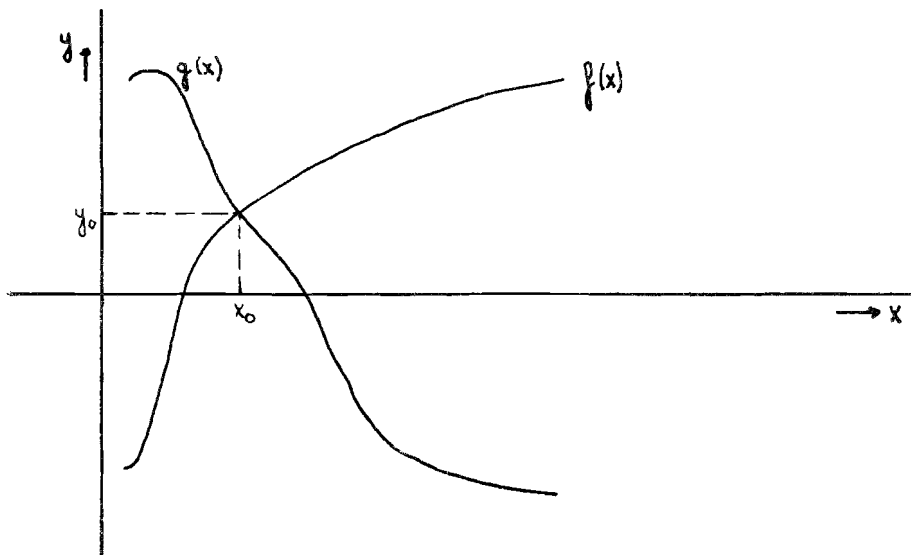


fig. 1: Snijpunt van $f(X)$ en $g(X)$: $Y_0 = f(X_0) = g(X_0)$.

Voor de gevonden numerieke benadering geldt:

$$y_n = f_n(x_0 + \Delta x) = g_n(x_0 + \Delta x) \quad (1)$$

Hierin is:

$$f_n(x) = f(x) + \epsilon_1 \quad (2a)$$

$$g_n(x) = g(x) + \epsilon_2 \quad (2b)$$

ϵ_1 en ϵ_2 zijn enige malen de "machinefout" (de fout die ontstaat doordat de rekenmachine voor een willekeurig getal het dichtstbijliggende machinegetal neemt) door de bewerkingen die met de niet exact juiste getallen moeten worden uitgevoerd volgens het voorschrift $f(X)$, respectievelijk $g(X)$; X is van de orde van grootte van de "machinefout".

Substitutie van (2a) en (2b) in (1) geeft:

$$f(x_0 + \Delta x) + \epsilon_1 = g(x_0 + \Delta x) + \epsilon_2 \quad (3)$$

ontwikkelen we $f(X_0 + \Delta X)$ en $g(X_0 + \Delta X)$ in een Taylorreeks en verwaarlozen we de hogere afgeleiden dan levert dit het volgende resultaat:

$$f(x_0) + \Delta x \cdot f'(x_0) + \epsilon_1 = g(x_0) + \Delta x \cdot g'(x_0) + \epsilon_2 \quad (4)$$

ofwel:

$$\Delta x \{ f'(x_0) - g'(x_0) \} = \epsilon_2 - \epsilon_1 \quad (5)$$

Stellen we:

$$|\varepsilon_1| + |\varepsilon_2| \leq 2|\varepsilon| \quad (6)$$

Man vinden we hieruit voor de fout in de berekende waarde voor X_0 :

$$|\Delta x| \leq \left| \frac{2\varepsilon}{f'(x_0) - g'(x_0)} \right| \quad (7)$$

Men hieruit dat de fout erg groot kan worden als de afgeleiden van de twee functies nagenoeg aan elkaar gelijk zijn.

De fout in de berekende waarde voor Y_0 kunnen we als volgt uitdrukking afleiden.

Als Y_0 geldt:

$$y_0 = \frac{f(x_0) + g(x_0)}{2} \quad (8)$$

De berekende waarde, die we vinden, wordt echter:

$$y_n = y_0 + \Delta y = \frac{f_n(x_0 + \Delta x) + g_n(x_0 + \Delta x)}{2} \quad (9)$$

Substitutie van (2a) en (2b) in deze uitdrukking, toepassing van de Taylorontwikkeling en verwaarlozen van de hogere afgeleiden geeft:

$$\begin{aligned}
 y_0 + \Delta y &= \frac{f(x_0) + g(x_0)}{2} + \Delta x \cdot \frac{f'(x_0) + g'(x_0)}{2} + \frac{\epsilon_1 + \epsilon_2}{2} \\
 &= y_0 + \Delta x \cdot \left\{ \frac{f'(x_0) + g'(x_0)}{2} \right\} + \frac{\epsilon_1 + \epsilon_2}{2} \quad (10)
 \end{aligned}$$

Absolute fout in de berekende waarde voor Y_0 wordt hiermee:

$$|\Delta y| \leq |\Delta x| \cdot \left| \frac{f'(x_0) + g'(x_0)}{2} \right| + |\epsilon| \quad (11)$$

De relatieve fout in de berekende waarde voor Y_0 vinden we:

$$\begin{aligned}
 \frac{\Delta y}{y_0} &= \frac{\Delta x}{x_0} \left\{ [f'(x_0) + g'(x_0)] \cdot \frac{x_0}{2y_0} \right\} + \frac{\epsilon_1 + \epsilon_2}{2y_0} \\
 &= \frac{\Delta x}{x_0} \left\{ [f'(x_0) + g'(x_0)] \cdot \frac{x_0}{f(x_0) + g(x_0)} \right\} + \frac{\epsilon_1 + \epsilon_2}{f(x_0) + g(x_0)} \\
 &= \frac{\Delta x}{x_0} \left\{ \frac{dy}{dx} \Big|_{x=x_0} \cdot \frac{x_0}{y_0} \right\} + \frac{\epsilon_1 + \epsilon_2}{f(x_0) + g(x_0)} \quad (12)
 \end{aligned}$$

EQUAL-VALUED-CAPACITOR ACTIVE-RC-NETWORK REALISATION OF A 3RD-ORDER LOWPASS BUTTERWORTH CHARACTERISTIC

Indexing terms Active filters, Capacitors

The letter describes an active RC network which realises a 3rd-order lowpass Butterworth characteristic. It requires one active element and has the practical advantage that all capacitors have the same value. A design chart is included.

In the practical design of active RC circuits, it is highly desirable, from a fabrication viewpoint, to be able to constrain the design so that all the capacitors have the same value. In this letter, we discuss such a constrained active RC network which realises a 3rd-order lowpass Butterworth characteristic. The network configuration is shown in Fig. 1. The triangle represents a voltage-controlled voltage source with a gain of K . The open-circuit voltage transfer function for this network is

$$\frac{V_2}{V_1} = K [R_1 R_2 R_3 p^3 + \{2R_1 R_3 + R_2 R_3 + R_1 R_2(2-K)\} p^2 + \{R_1 + R_3 + (R_1 + R_2)(2-K)\} p + 1]^{-1} \quad (1)$$

The frequency-normalised lowpass Butterworth transfer function to be realised is

$$\frac{V_2}{V_1} = \frac{H}{p^3 + 2p^2 + 2p + 1} \quad (2)$$

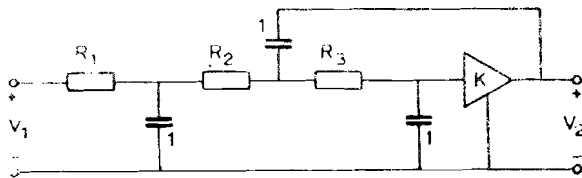


Fig. 1 Network configuration

The solution for the values of the resistors (as a function of the gain of the voltage-controlled voltage source) which realise this network function is readily found by solving the set of nonlinear equations

$$\left. \begin{aligned} 1 &= R_1 R_2 R_3 \\ 2 &= 2R_1 R_3 + R_2 R_3 + R_1 R_2(2-K) \\ 2 &= R_1 + R_3 + (R_1 + R_2)(2-K) \end{aligned} \right\} \quad (3)$$

for a sequence of values of the parameter K . The results are shown in Fig. 2. It should be noted that solutions exist for a range of K from approximately 1.86 to 2.99, although the spread of resistances is somewhat extreme at the limits of this range. Of special interest is the solution for $K = 2$, since this gain is easily produced by a high-gain differential-input amplifier using two equal-valued feedback resistors. For such a K , the resistors of the network shown in Fig. 1 have the values

$$R_1 = 1.565 \Omega$$

$$R_2 = 1.469 \Omega$$

$$R_3 = 0.435 \Omega$$

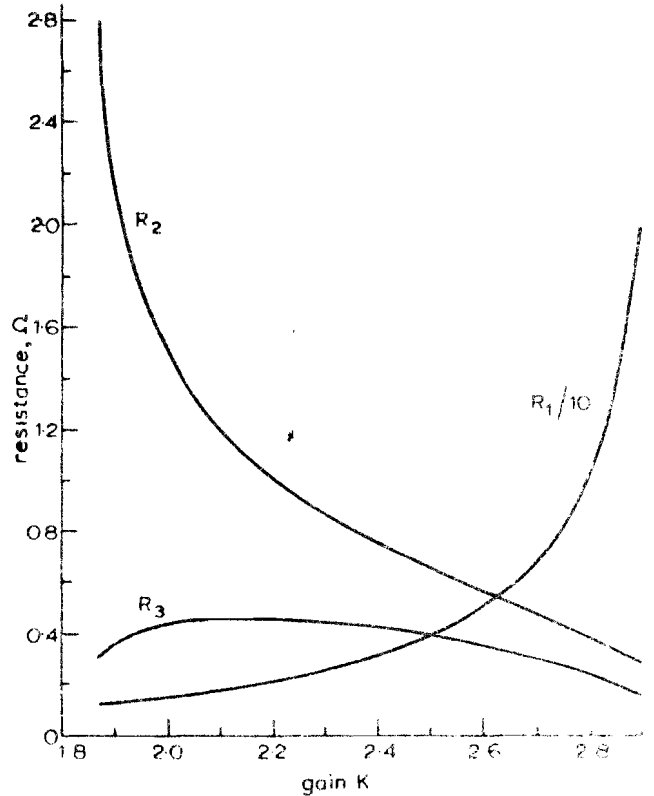


Fig. 2 Resistance/gain results

Acknowledgment The author would like to acknowledge the efforts of R. S. Aikens and M. Al-Rahmani in performing the computer programming used to obtain the results given in this letter. The research described was supported in part by the Instrumentation Division of the Ames Research Center of the US National Aeronautics & Space Administration under grant NGL-03-002-136.

I. P. HUFELSMAN

29th April 1971

Department of Electrical Engineering
University of Arizona
Tucson, Ariz. 85721, USA

An Equal-Capacitor Active RC Network Realization of Fourth-Order Low-Pass Butterworth Characteristic

L. P. HUELSMAN

Abstract—This letter describes an active RC network which realizes a fourth-order low-pass Butterworth characteristic. It requires one active element and has the practical advantage that all capacitors have the same value. A design chart is included.

In the practical design of active RC circuits it is highly desirable, from a fabrication viewpoint, to be able to constrain the design so that all capacitors have the same value. In this paper we discuss such a constrained active RC network which realizes a fourth-order low-pass Butterworth characteristic. The network configuration is shown in Fig. 1. In this figure the triangle represents a voltage-controlled voltage source with a gain of K . The open-circuit voltage transfer function for this network is¹

$$\frac{V_2(s)}{V_1(s)} = \frac{K}{a_4s^4 + a_3s^3 + a_2s^2 + a_1s + 1} \quad (1)$$

where

$$a_4 = \left(\frac{R_3}{R_2} \right) + R_4 + \frac{R_1R_3}{R_2} \left\{ \left(\frac{R_4}{R_2} + 1 \right) + R_2(1-K) + \frac{R_4}{R_2} \right\}$$

$$a_3 = \left(\frac{R_3}{R_2} \right) + R_3F \left\{ \left(\frac{R_4}{R_3} + 1 \right) + R_4 \left[F \left(1 + \frac{R_3}{R_2} \right) + \frac{K R_1}{R_2} \right] \right\}$$

$$a_2 = R_4 + R_4 \left\{ R_1R_2 \left(1 + \frac{R_3}{R_2} \right) + R_2F \right\}$$

$$a_1 = R_1 + R_3R_4 \quad (2)$$

and

$$F = (2-K)R_2 + R_1 \quad (3)$$

for the Butterworth case we find that $a_1 = a_3 = 2.6131$, $a_2 = 3.4142$, and $a_4 = 1$. Solving the set of simultaneous nonlinear equations given in (2) for these values, and for a range of values of K , we obtain the design chart shown in Fig. 2. Of special interest is the solution for $K = 2$, for which we obtain

$$R_1 = 0.1339$$

$$R_2 = 3.8931$$

$$R_3 = 2.4792$$

and

$$R_4 = 0.7736.$$

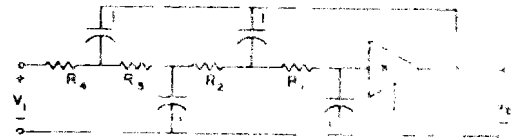


Fig. 1.

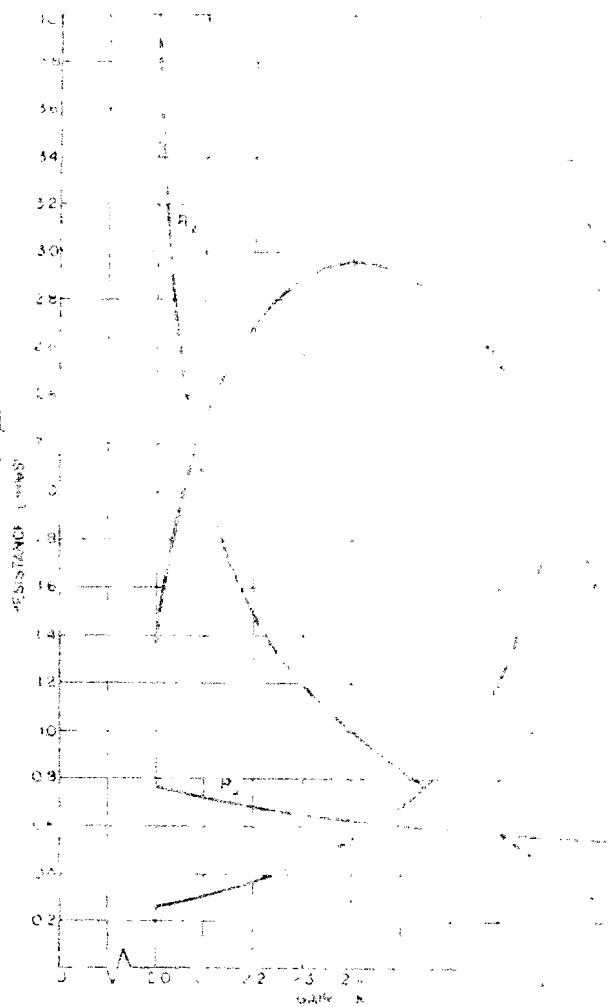


Fig. 2.

Manuscript received May 28, 1974. This work was supported by the Instrumentation Division of the Office of Naval Research, National Aeronautics and Space Administration, Grant N00019-73-002-139.

The author is with the Department of Electrical Engineering, University of Arizona, Tucson, Arizona 85724.
These equations were provided by the author upon request for reproduction.