

Contents

1	Inleiding	3
2	Klassieke tijdsdiskrete structuren	7
2.1	Inleiding	7
2.2	Nietrekursieve tijdsdiskrete structuren	8
2.3	Rekursieve tijdsdiskrete structuren	10
2.4	4 rekursieve structuren	11
2.5	De kam (Comb) filters.	13
2.6	Transpositie	13
2.7	Een stabiliteitstest	14
2.8	Een realisatievorm voor tijdsdiskrete all-pass funkties	16
2.9	Een realisatievorm van rekursieve overdrachtsfunkties	18
2.10	Vraagstukken	20
3	Ontwerpmethoden voor tijdsdiskrete filters.	27
3.1	Introductie	27
3.2	Nietrekursieve filters met exact lineair faseverloop.	28
3.3	Ontwerp van nietrekursieve overdrachtsfunkties met exact lineair faseverloop door middel van de Fourier transformatie en gebruik van vensters	31
3.4	Methode van impuls-invariantie	34
3.5	De bilineaire transformatie methode	36
3.6	Verliesvrijheid in het tijdsdiskrete geval	41
3.7	Eenvoudige tijdsdiskrete laddersynthese.	44
3.8	Vraagstukken	46
4	Digitale golffilters	52
4.1	Inleiding	52
4.2	De vertaalprocedure	52
4.2.1	De w -vlak zelfinductie	53
4.2.2	De w -vlak capaciteit	53

4.2.3	De w-vlak bron	54
4.2.4	De w-vlak belastingsweerstand	54
4.2.5	De driepoort parallel verbinding	54
4.2.6	De driepoort serie verbinding	57
4.3	Twee voorbeelden	58
4.4	Opmerkingen	62
4.5	Een uitbreiding	63
4.5.1	Inleiding	63
4.5.2	De digitale vertaling	63
4.5.3	Enkele w vlak (Kuroda) identiteiten	64
4.5.4	Enkele voorbeelden	68
4.6	Een andere realisatiestructuur	70
4.7	Een filtervoorbeeld	74
5	Het implementeren van tijds-diskrete structuren in (programmeerbare) digitale hardware.	81
5.1	Introductie.	81
5.2	Parallelisme in tijds-diskrete structuren.	82
5.3	Verbetering van parallelisme door transformaties	89
5.4	Realisatie in programmeerbare hardware	95
5.5	Conclusies en opmerkingen.	103
6	Systemen met verschillende bemonsteringsfrequenties.	105
6.1	Inleiding	105
6.2	Verlaging van de bemonsteringsfrequentie.	107
6.2.1	Het gevaar van "aliasing" als een decimatie-filter gebruikt wordt.	107
6.2.2	De realisatie van niet-rekursieve decimatie filters.	107
6.2.3	Voorbeeld van het gebruik van een decimator.	109
6.3	Verhoging van de bemonsteringsfrequentie.	112
6.3.1	Voorbeeld van het gebruik van een interpolator.	114
6.4	Opgaven.	118
7	Eindige woordlengte effecten in tijdsdiskrete systemen.	121
7.1	Inleiding	121
7.2	Kwantisatie	122
7.3	Overflow:	123
7.4	Underflow.	124
7.5	Opgaven:	125
7.6	A/D conversie kwantisatie.	128

Chapter 1

Inleiding

De geschiedenis van de filtertechniek is analoog begonnen: analoge signalen, bewerkt met behulp van analoge netwerken.

Deze geschiedenis heeft filtereigenschappen van analoge filters tevoorschijn gebracht, die we met veel nut kunnen gebruiken in het tijdsdiskrete geval (verliesvrijheid, digitale golffilters).

De ontwikkeling van de analoge filtertheorie heeft zich (als onderdeel van de netwerktheorie) in het begin geconcentreerd op de situatie geschetst in figuur 1.1.

Figuur 1.1 schetst een niet-ideale spanningsbron (een spanningsbron met inwendige weerstand als model voor de signaalbron, denk b.v. aan een kabel) via een netwerk aangesloten op een belasting, met als model een weerstand (denk b.v. aan de karakteristieke weerstand van een kabel).

Het filternetwerk wat we zullen gebruiken is een 'geaarde tweepoort'. Dit als een onderdeel van de netwerktheorie waar ook meer algemene 'n-poort netwerken' worden bestudeerd.

De filterwerking houdt in dat we de energieverdeling aanwezig in het frekwentie-

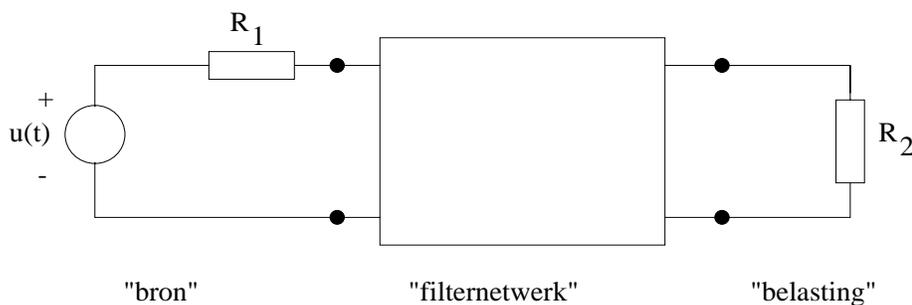


Figure 1.1: blokschema voor de filterwerking

spektrum van de signaalbron willen bewerken d.m.v. een lineair netwerk zodanig dat de energieverdeling die in de belasting terecht komt, een andere is. Bijvoorbeeld willen we bepaalde frekwenties aanwezig in de signaalbron, helemaal 'onderdrukken' (weghalen).

We zullen ons richten op de reeds bekende soorten filterkarakteristieken: laagdoorlaat, hoogdoorlaat, banddoorlaat, etc, zie hiervoor figuur 1.2.

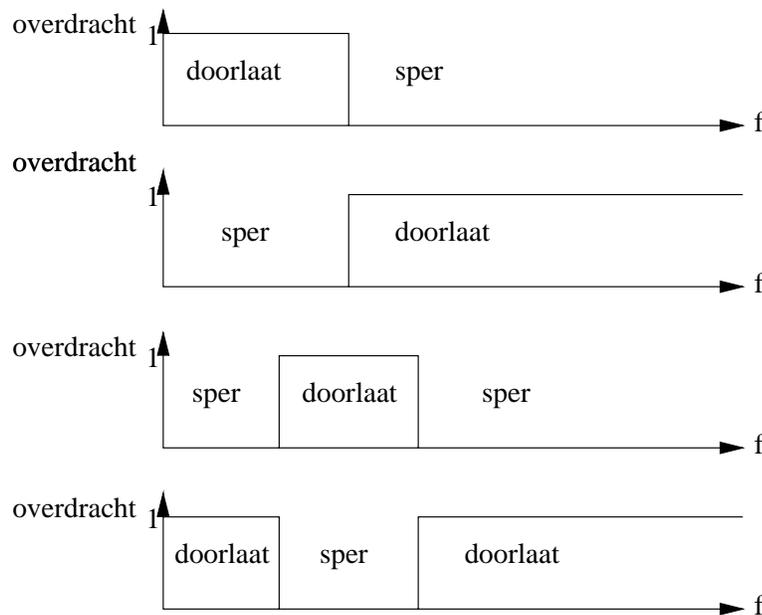


Figure 1.2: verschillende typen filters

De van ouds meest voor de hand liggende componenten om te gebruiken in het filternetwerk zijn:

- weerstand
- zelfinductie
- capaciteit

Dat betekent dus: Passieve netwerken.

Het gebruik van de weerstand heeft als nadeel dat daarin vermogen verloren gaat. Vandaar de grote aandacht voor:

Passieve, verliesvrije netwerken,

als filternetwerk.

In Deel 1 van dit diktaat zijn we begonnen de tijdscontinue filtertechniek te introduceren. De focus is vooral op de amplitude karakteristieken als benaderingen van de ideale genormeerde laagdoorlaatkarakteristiek.

Hiervoor hebben we eerst de standaard benaderingsmethoden besproken: De Butterworth, Chebyshev en Cauer benaderingen.

Daarna volgde meteen de methode om dan de overdrachtfuncties daaruit te vinden.

Tenslotte, in Deel 1, zijn een aantal realisatie methoden besproken om deze overdrachtfuncties ook werkelijk te implementeren.

Terugkomend op de historische ontwikkeling in de filtertechniek:

Heel snel bleek de LC-ladder afgesloten in weerstanden de beste structuur te zijn. Het bleek mogelijk om met elementen tot ongeveer 0.1% nauwkeurig, filterkarakteristieken te maken met een orde hoger dan 30. Dit betekent dat selectiviteit gemaakt kan worden die met een lagere orde niet mogelijk is en toch met relatief onnauwkeurige elementwaarden.

Even enkele (triviale) opmerkingen:

1. Een LC-netwerk (+weerstand) is nooit instabiel.
2. Dit netwerk heeft geen voedingsspanning nodig.

Toch, de ontwikkeling van filtertoepassingen bleek steeds kleinere afmetingen van de filters te vragen. De zelfinductie bleek het moeilijkst 'klein' te krijgen: dit element is vrijwel niet te integreren in chipvorm.

Het netwerkelement de gyrator verscheen. De combinatie gyrator + capaciteit levert een manier op een zelfinductie na te bootsen. De gyrator bleek d.m.v. electronica goed te realiseren. Dit leverde dus een elektronische realisatie van het netwerkelement de gyrator, met zijn beperkingen, b.v. een voedingsspanning die nodig is. Verliesvrije ladders werden nagebootst door gyrator-C filters.

In Deel 2 van dit diktaat is het begrip verliesvrijheid, vermogensoverdracht met behulp van verliesvrije netwerken van een bron naar een belasting, verder uitgewerkt. De verstrooiingsmatrix S is ingevoerd. Hiermee blijkt deze vermogensoverdracht bij uitstek te kunnen worden beschreven. Begrippen als:

maximale vermogensoverdracht, aanpassing, reflectie, zijn besproken, evenals de z.g. Belevitch polynomen.

Aandacht is ook besteed aan eenvoudige laddersynthese en het Jauman netwerk als verliesvrij alternatief voor de laddernetwerken van oneven orde.

Verdergaande eisen van verkleining leverde allerlei oplossingen: RC-actieve - en switched-capacitor filters. Volledige integratie en digitalisering leidt uiteindelijk tot digitale filters. Ook hier worden de beste filters gevonden als de LC-ladders worden nagebootst. In dit geval zijn het de optellers, vermenigvuldigers en delay-elementen die de basisbouwstenen vormen van de filterschakelingen en dus waard zijn nauwkeurig te worden bestudeerd.

De tijdsdiskrete filtertechniek is het onderwerp in Deel 3 van het diktaat.

Tijdsdiskrete structuren en ontwerpmethoden daarbij komen aan de orde. Hierbij komen nietrekursieve zowel als rekursieve structuren, lineair faseverloop, de bilineaire transformatie en digitale golffilters aan de orde.

Ook implementatie aspecten zoals gebruik van programmerbare hardware, verschillende bemonsteringsfrequenties en eindige woordlengte effecten, komen aan bod.

Chapter 2

Klassieke tijdsdiskrete structuren

2.1 Inleiding

We zullen het in dit hoofdstuk hebben over de lineaire, tijdsinvariante, diskrete-tijd structuren. De beschrijving ervan met differentie-vergelijkingen en z -transformatie, zullen we als bekend veronderstellen. Omdat deze structuren, met één ingang en uitgang, vaak als filter gebruikt wordt, zullen we deze benaming vaak gebruiken.

We kunnen deze structuren op 2 manieren klassificeren:

De eerste:

- nietrekursieve discrete-tijd structuren, dw.z. structuren zonder feedback lus.
- rekursieve discrete-tijd structuren, d.w.z. structuren met tenminste één feedback lus.

De tweede indeling:

- Finite impulse response filters,
- Infinite impulse response filters.

Deze indeling zegt dus iets over de lengte van de impulse response van de structuren.

Figuur 2.1 maakt duidelijk welke relatie bestaat tussen de beide indelingen: Een nietrekursieve structuur heeft altijd een eindige impulse response, maar het omgekeerde is niet waar. Een eindige impulse response blijkt ook mogelijk te zijn in bepaalde rekursieve structuren.

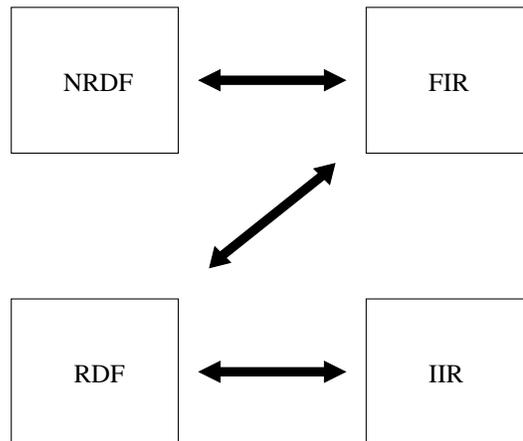


Figure 2.1: relatie tussen de beide indelingen

Een oneindig lange impulse response kunnen we alleen maken met een rekursieve structuur, maar het omgekeerde is niet altijd waar. Het is mogelijk met bepaalde rekursieve structuren een eindige responsie te realiseren.

2.2 Nietrekursieve tijdsdiskrete structuren

Twee voorbeelden van nietrekursieve structuren zijn gegeven in figuur 2.2.

Bepaal zelf de overdrachtsfunctie van deze structuren!

Als u dit doet, blijkt de overdrachtsfunctie in z , die u vindt bij figuur 2.2b, van de eerste orde in z te zijn. De overdrachtsfunctie die u vindt bij figuur 2.2a is van de tweede orde in z .

Definitie:

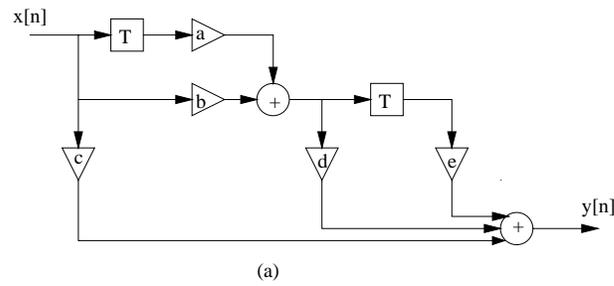
Een structuur die een overdrachtsfunctie van de N^e orde (zonder gemeenschappelijke factor in teller en noemer) realiseert, wordt kanoniek genoemd als er precies n delay elementen in de structuur voorkomen.

Figuur 2.2a is dus kanoniek, figuur 2.2b is niet kanoniek.

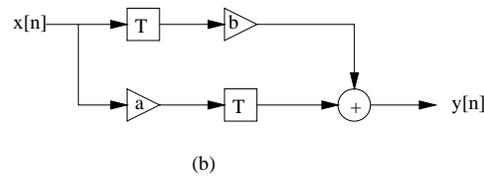
Een veel gebruikte nietrekursieve structuur is die van figuur 2.3.

Dit type filter wordt transversaal filter genoemd. Deze structuur heeft twee aantrekkelijke eigenschappen:

1. Hetingangssignaal $x[n]$ wordt slechts verschoven opgeslagen in de delay elementen. Het is een kanonieke structuur.



(a)



(b)

Figure 2.2: twee voorbeelden

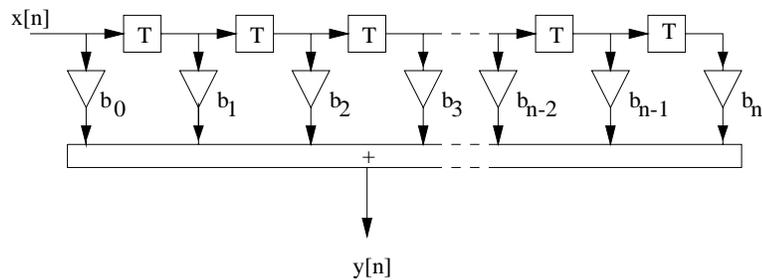


Figure 2.3: transversale structuur

2. De vermenigvuldigingscoëfficiënten in de structuur zijn rechtstreeks te vinden als coëfficiënten in de impulsresponsie $h[n]$ zoals uit het volgende blijkt. Bij een orde n zijn er $n+1$ coëfficiënten!

Uit formule 2.1 volgt dat de overdrachtsfunctie alleen nulpunten heeft, behalve nog N polen in de oorsprong in het z -vlak.

De impulsresponsie is eindig, van lengte $N+1$.

Een belangrijke eigenschap van nietrekursieve filters die hier direkt mee samenhangt is die van stabiliteit:

1. Stabiliteit van een lineaire tijdsdiskrete overdrachtsfunctie $H(z)$ is gegarandeerd als alle polen van de functie binnen de eenheidscirkel liggen in het z -vlak. Dit is hier het geval, ze liggen allemaal in de oorsprong.

2. De impulse responsie is eindig, de waarden van de impulse responsie zijn ook eindig en dus de som van de absolute waarden van de impulse responsie is ook eindig. Dit is ook een uiting van stabiliteit, want bij instabiliteit zou de som van absolute waarden oneindig groot moeten kunnen worden.

$$H(z) = \frac{Y(z)}{X(z)} = b_0 + b_1 z^{-1} + \dots + b_N z^{-N} = \sum_{i=0}^N b_i z^{-i} = \frac{\sum_{i=0}^N b_i z^{N-i}}{z^N} \quad (2.1)$$

$$h[n] = b_0 \delta[n] + b_1 \delta[n-1] + \dots + b_N \delta[n-N] = \sum_{i=0}^N b_i \delta[n-i] \quad (2.2)$$

2.3 Rekursieve tijdsdiskrete structuren

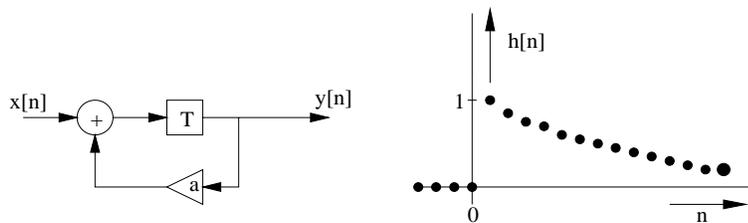


Figure 2.4: een eerste orde rekursieve structuur

Een eigenschap van rekursieve tijdsdiskrete structuren is dat er tenminste één feedback lus in zit. Een eenvoudig voorbeeld is te zien in figuur 2.4. Dit netwerk kan worden beschreven door de volgende differentie vergelijkingen:

$$\begin{aligned} \nu[n] &= x[n] + ay[n] \\ y[n] &= \nu[n-1] = x[n-1] + ay[n-1] \end{aligned} \quad (2.3)$$

met als impulse responsie:

$$h[n] = a^{n-1} u[n-1] \quad (2.4)$$

De overdrachtsfunctie $H(z)$ is eenvoudig te vinden:

$$H(z) = \frac{z^{-1}}{1 - az^{-1}} \quad (2.5)$$

2.4 4 rekursieve structuren

In het volgende zullen we 4 verschillende structuren kort bespreken. Dit zijn structuren die elk gebruikt kunnen worden om een willekeurige stabiele overdrachtsfunctie mee te realiseren.

Het zijn:

1. De direkte vorm no. 1,
2. De direkte vorm no. 2,
3. De cascade structuur,
4. De parallele structuur.

De direkte vorm no. 1:

De algemene differentie vergelijking voor een rekursief filter is:

$$y[n] = \sum_{i=0}^N b_i x[n-i] + \sum_{i=1}^M a_i y[n-i] \quad (2.6)$$

De overdrachtsfunctie die erbij hoort is:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^N b_i z^{-i}}{1 - \sum_{i=1}^M a_i z^{-i}} \quad (2.7)$$

Het is direct duidelijk dat deze structuur niet kanoniek is, en dat de coëfficiënten uit de differentie vergelijking zonder verandering voorkomen in de structuur als vermenigvuldigingscoëfficiënt.

De direkte vorm no. 2:

We kunnen de volgorde van berekeningen in figuur 2.5 veranderen: In figuur 2.5 wordt eerst een nietrekursief deel getekend en daarna een rekursief deel. In figuur 2.6 is dit omgekeerd. Nu ontstaan 2 rijen delay elementen waarin voor een deel dezelfde getallen worden opgeslagen. Dit is natuurlijk onnodig,

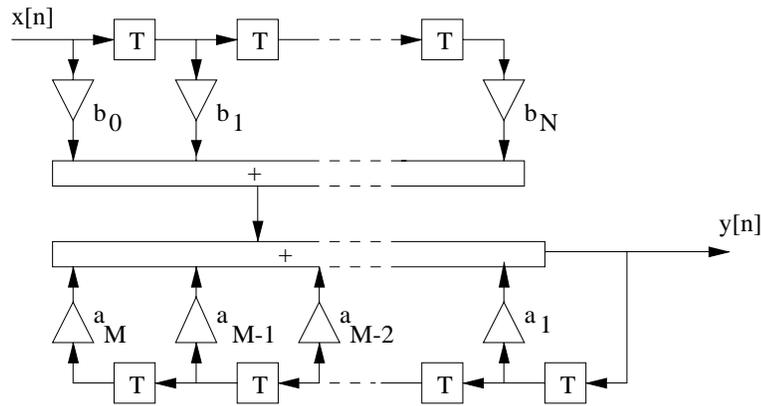


Figure 2.5: de direkte vorm no. 1

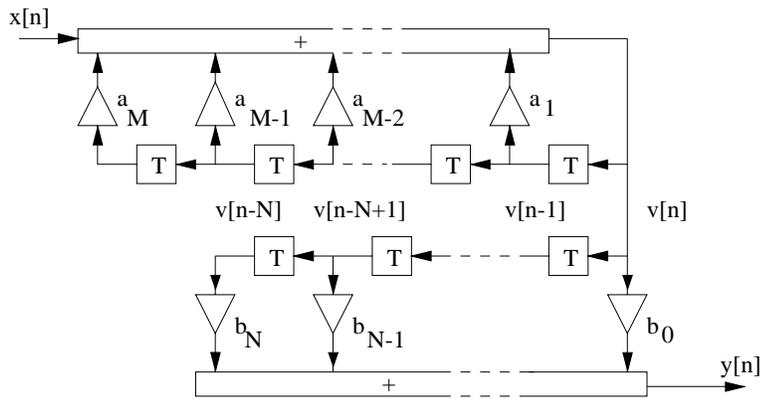


Figure 2.6: rekursieve- en nietrekursieve deel van plaats verwisseld

het resultaat is figuur 2.7. Het is niet moeilijk in te zien dat deze structuur wel kanoniek is.

De cascade structuur:

De overdrachtsfunctie $H(z)$ kan als volgt als een cascade van overdrachts functies van lagere orde worden geschreven (zie ook figuur 2.8):

$$H(z) = H_1(z) \dots H_i(z) \dots H_K(z) \tag{2.8}$$

De parallele structuur

We kunnen de overdrachtsfunctie $H(z)$ ook als een som schrijven, (zie ook figuur 2.9):

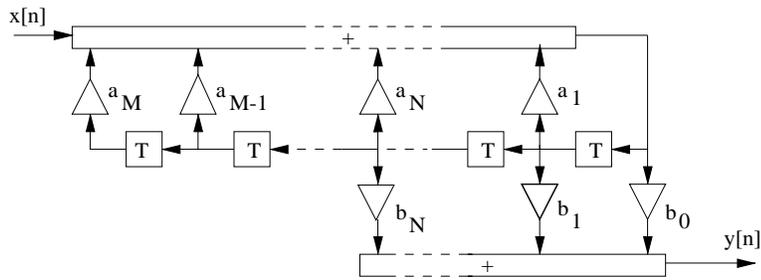


Figure 2.7: de direkte vorm no. 2

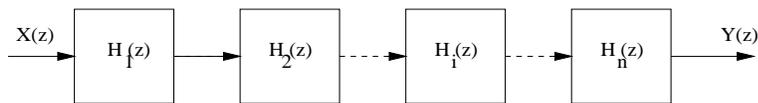


Figure 2.8: de cascade structuur

$$H(z) = H_0 + H_1(z) + \dots + H_i(z) + \dots H_K(z) \quad (2.9)$$

2.5 De kam (Comb) filters.

Een kam filterstructuur wordt gekregen door in een willekeurige tijdsdiskrete structuur elk afzonderlijk delay element T te vervangen door N gecascadeerde delay elementen T. Er ontstaat dan een structuur met als overdrachtsfunctie $G(z) = H(z^N)$. Dit wil zeggen dat de frequentie response in het fundamentele interval $(-\pi \leq \theta < \pi)$ van $H(z)$ N keer voorkomt bij $G(z)$. Neem als voorbeeld $H(z) = 1 - z^{-1}$. $G(z)$ wordt nu:

$$G(z) = \frac{Y(z)}{X(z)} = 1 - z^{-N} \quad (2.10)$$

2.6 Transpositie

Theorema:

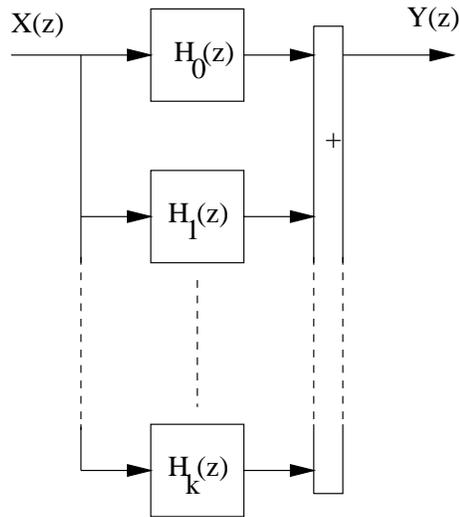


Figure 2.9: de parallele structuur

De overdrachtsfunctie van een lineair tijdsinvariant tijdsdiskreet systeem verandert niet als we transpositie toepassen op het netwerk, d.w.z.:

- De signalen in het netwerk veranderen van richting (dit houdt dus ook in dat de ingang uitgang wordt en omgekeerd).
- Optellers worden knooppunten en omgekeerd.

2.7 Een stabiliteitstest

Stel we hebben de rekursieve overdrachtsfunctie

$$H(z) = \frac{T(z)}{N(z)} = \frac{t_0 + \dots + t_n z^{-n}}{1 + n_0 z^{-1} + \dots + n_n z^{-n}} \quad (2.11)$$

We gaan van $H(z)$ de stabiliteit onderzoeken. Dit doen we aan de hand van $G(z)$, een all-pass functie afgeleid van $H(z)$ (hetzelfde noemerpolynoom).

$$G_n(z) = \frac{P(z)}{N(z)} = \frac{z^{-n} N(1/z)}{N(z)} \quad (2.12)$$

Als $G(z)$ stabiel is dan is ook $H(z)$ stabiel. Het is dus voldoende weer te geven wanneer $G(z)$ stabiel is. We nemen aan dat $P(z)$ en $G(z)$ géén gemeenschappelijke factoren hebben.

De coefficient n_n in (2.11) is het produkt van alle polen van $H(z)$ en $G(z)$.

Voor stabiliteit is dus tenminste nodig:

$$n_n^2 < 1 \quad (2.13)$$

Uit de literatuur is bekend dat nodig en voldoende is voor stabiliteit van $G_n(z)$:

1. formule (2.13) geldt,
2. $G_{n-1}(z)$ is stabiel.

Hierbij geldt:

$$z^{-1}G_{n-1}(z) = \frac{G_N(z) - n_n}{1 - n_n G_n(z)} \quad (2.14)$$

Hierin is $G_{n-1}(z)$ een all-pass van orde n-1.

Door deze test iteratief door te voeren tot aangekomen wordt bij $G_0(z)$, is aan te tonen dat een $G_n(z)$ en dus een bijbehorende $H(z)$ stabiel is.

Dit werkt natuurlijk alleen als de procedure niet voortijdig afbreekt door een gemeenschappelijke factor.

Voorbeeld 1:

$$H_3(z) = \frac{2.3 + 4.0z^{-1} + 3.6z^{-2} + 1.9z^{-3}}{1 + 0.9z^{-1} + 0.8z^{-2} + 0.3z^{-3}} \quad (2.15)$$

De bijbehorende all-pass $G_3(z)$ wordt:

$$G_3(z) = \frac{0.3 + 0.8z^{-1} + 0.9z^{-2} + 1.0z^{-3}}{1 + 0.9z^{-1} + 0.8z^{-2} + 0.3z^{-3}} \quad (\text{nodig: } n_3 = 0.3 < 1!)$$

$G_3(z)$ is dus stabiel als $G_2(z)$ stabiel is met:

$$G_2(z) = \frac{zG_3(z) - 0.3z}{1 - 0.3G_3(z)} = \frac{0.5824 + 0.7253z^{-1} + z^{-2}}{1 + 0.7253z^{-1} + 0.5824z^{-2}} \quad (\text{nodig: } n_2 = 0.5824 < 1!)$$

$G_2(z)$ is dus stabiel als $G_1(z)$ stabiel is met:

$$G_1(z) = \frac{zG_2(z) - 0.5824z}{1 - 0.5824G_2(z)} = \frac{0.45836 + z^{-1}}{1 + 0.45836z^{-1}} \quad (\text{nodig: } n_1 = 0.45836 < 1!)$$

$G_1(z)$ is dus stabiel als $G_0(z)$ stabiel is met:

$$G_0(z) = \frac{zG_1(z) - 0.4534z}{1 - 0.4534G_1(z)} = 1 \quad \Rightarrow \Rightarrow G_0(z) \text{ stabiel} \Rightarrow \Rightarrow H_3(z) \text{ stabiel.}$$

2.8 Een realisatievorm voor tijdsdiskrete all-pass funkties

Formule (2.14) uit de vorige paragraaf die de rekursieve stabiliteitstest beschrijft, levert direkt een realisatievorm voor de all-pass overdrachtsfunctie $G_n(z)$: figuur 2.10.

Dit is gemakkelijk na te gaan door in dit netwerk $G_n(z)$ te gaan uitdrukken in $G_{n-1}(z)$, of omgekeerd.

Er zijn nog twee andere realisatievormen te geven, zie de figuren 2.11 en 2.12.

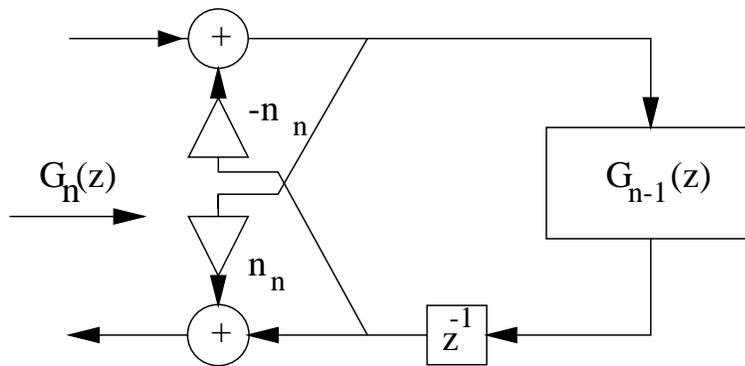


Figure 2.10: Iteratieve all-pass structuur

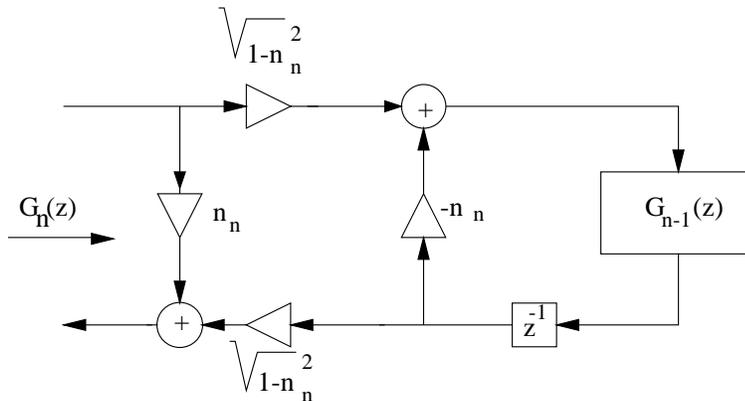


Figure 2.11: Alternatieve iteratieve all-pass

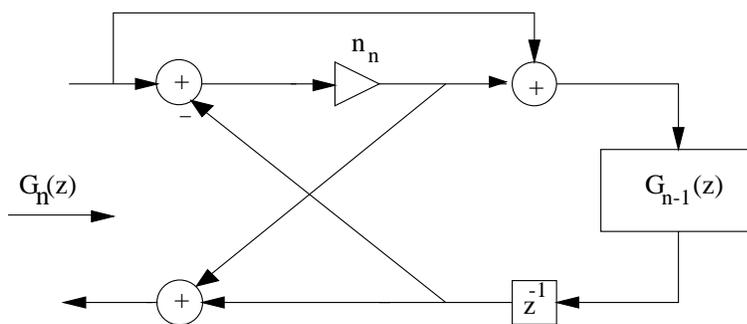


Figure 2.12: Iteratieve all-pass met slechts een vermenigvuldiger

Figuur 2.12 is een realisatievorm met slechts één vermenigvuldigings coefficient.

Hierdoor kan een n^e orde all-pass gerealiseerd worden met slechts n vermenigvuldigers en n delay elementen!

Het rekenvoorbeeld (voorbeeld 1) ziet er nu uit als figuur 2.13 als voor de rekensekties de structuur van figuur 2.10 wordt gekozen.

In figuur 2.13 geldt: $\frac{Y_3(z)}{X(z)} = G_3(z) = \frac{0.3 + 0.8z^{-1} + 0.9z^{-2} + z^{-3}}{1 + 0.9z^{-1} + 0.8z^{-2} + 0.3z^{-3}}$

Bij berekening in figuur 2.13 van $Y_2(z)$, $Y_1(z)$ en $Y_0(z)$ blijkt:

$$H_2(z) = \frac{Y_2(z)}{X(z)} = \frac{0.5824 + 0.72527z^{-1} + z^{-2}}{1 + 0.9z^{-1} + 0.8z^{-2} + 0.3z^{-3}}$$

$$H_1(z) = \frac{Y_1(z)}{X(z)} = \frac{0.4583 + z^{-1}}{1 + 0.9z^{-1} + 0.8z^{-2} + 0.3z^{-3}}$$

$$H_0(z) = \frac{Y_0(z)}{X(z)} = \frac{1}{1 + 0.9z^{-1} + 0.8z^{-2} + 0.3z^{-3}}$$

Het is eenvoudig in te zien dat we door middel van een lineaire combinatie van $G_3(z)$, $H_2(z)$, $H_1(z)$ en $H_0(z)$ elke 3^e graads overdrachtsfunctie kunnen realiseren. Dit wordt in de volgende paragraaf verkort (en zonder bewijs) opgeschreven.

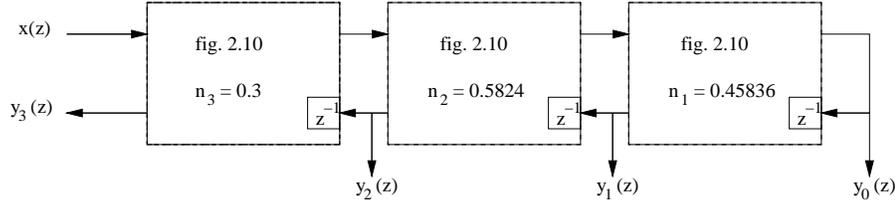


Figure 2.13:

2.9 Een realisatievorm van rekursieve overdrachtsfuncties

Met:
$$H_n(z) = \frac{T_n(z)}{N_n(z)} = \frac{t_0 + \dots + t_n z^{-n}}{1 + n_0 z^{-1} + \dots + n_n z^{-n}} = \frac{Y(z)}{X(z)}$$

We weten dat: $\frac{N_n(1/z)z^{-n}}{N_n(z)}$ een n^e orde all-pass is.

De rekursieve berekening voor de all-pass zoals die in het vorige is beschreven, kan als volgt opgeschreven worden voor de vermenigv. coefficienten n_m (voor $m=n, n-1, \dots, 0$) en het polynoom N_m (van N_n, N_{n-1}, \dots, N_0):

$$\begin{aligned} B_m &= N_m(1/z)z^{-m} \\ n_m &= N_{m,m} \\ N_{m-1}(z) &= [N_m(z) - n_m B_m(z)] / (1 - n_m^2) \end{aligned}$$

Met $\nu_m =$ een vermenigv. coefficient, en:

$$\begin{aligned} \nu_m &= T_{m,m} \\ T_{m-1}(z) &= T_m(z) - B_m(z) \cdot \nu_m \end{aligned}$$

Levert dit het netwerk van figuur 2.14.

Voor voorbeeld 1 worden de uitgewerkte polynomen zoals hierna aangegeven.

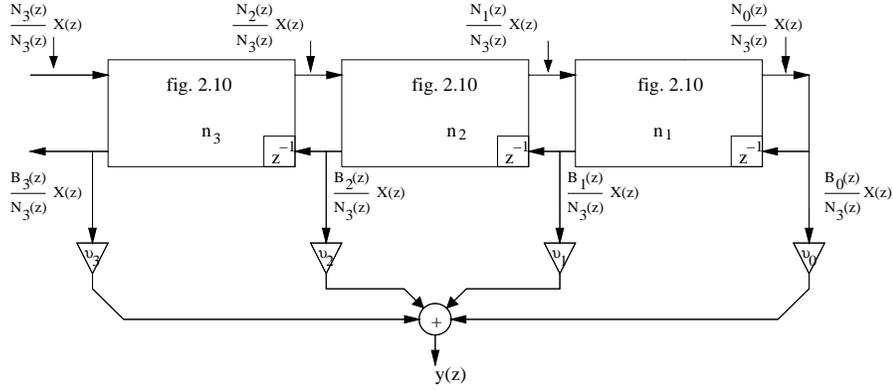


Figure 2.14:

$$N_3(z) = 1 + 0.9z^{-1} + 0.8z^{-2} + 0.3z^{-3} \quad n_3 = 0.3$$

$$B_3(z) = 0.3 + 0.8z^{-1} + 0.9z^{-2} + z^{-3}$$

$$T_3(z) = 2.3 + 4.0z^{-1} + 3.6z^{-2} + 1.9z^{-3} \quad \nu = 1.9$$

$$N_2(z) = 1 + 0.72527z^{-1} + 0.5824z^{-2} \quad n_2 = 0.5824$$

$$B_2(z) = 0.5824 + 0.72527z^{-1} + z^{-2}$$

$$T_2(z) = 1.73 + 2.48z^{-1} + 1.89z^{-2} \quad \nu_2 = 1.89$$

$$N_1(z) = 1 + 0.4583z^{-1} \quad n_1 = 0.4583$$

$$B_1(z) = 0.4583 + z^{-1}$$

$$T_1(z) = 0.6292 + 1.10924z^{-1} \quad \nu_1 = 1.10924$$

$$N_0(z) = 1$$

$$B_0(z) = 1$$

$$T_0(z) = 0.12083 \quad \nu_0 = 0.12083$$

2.10 Vraagstukken

1. Gegeven is de tijdsdiskrete overdrachtsfunctie $H(z)$:

$$H(z) = 1 + 2z^{-1} + az^{-2}$$

Voor welke waarden van a is $H(z)$ stabiel?

$H(z)$ is nietrekursief, dus $H(z)$ is stabiel voor elke reële a .

2. Teken een transversale filterstructuur die onderstaande $H(z)$ realiseert:

$$H(z) = a_0 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3} + a_4z^{-4}$$

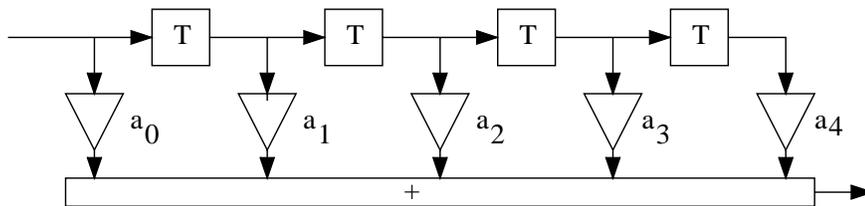


Figure 2.15: figuur hoort bij vraag 2

3. Gegeven is de onderstaande overdrachtsfunctie $H(z)$:

$$H(z) = \frac{1}{(z - a)^2 + 0.25}$$

Voor welke waarden van a is $H(z)$ strikt stabiel?

$$H(z) = \frac{1}{(z - a - 0.5)(z - a + 0.5)} \quad |a| < 0.5$$

4. Gegeven is $H(z)$:

$$H(z) = \frac{1 + 2z^{-1} + 3z^{-2}}{1 + 4z^{-1} + 5z^{-2}}$$

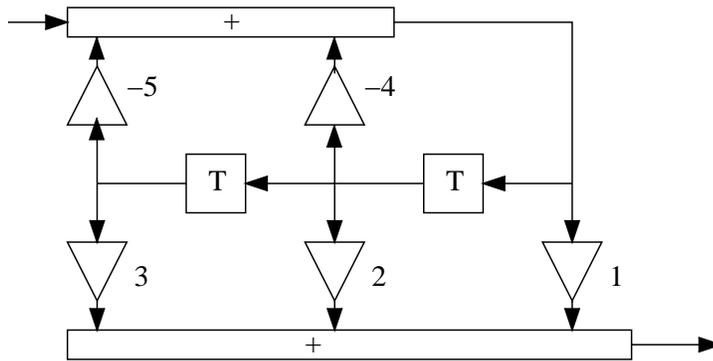


Figure 2.16: figuur hoort bij vraag 4

Geef een kanonieke structuur voor $H(z)$ waarin de coëfficiënten van $H(z)$ (eventueel met een - teken) direkt voor komen als vermenigvuldigungscoëfficiënten.

5. Gegeven is het tijdsdiskreet netwerk volgens bijgaande figuur:

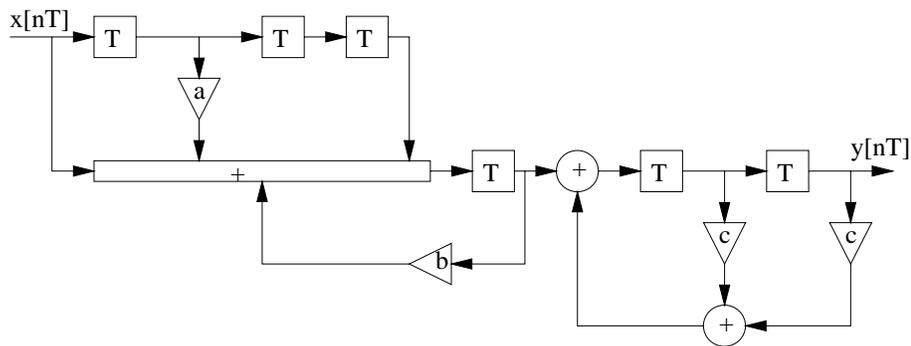


Figure 2.17: figuur hoort bij vraag 5

Gevraagd:

- Bepaal de overdrachtsfunctie $H(z) = \frac{Y(z)}{X(z)}$.

$$H(z) = (1 + az^{-1} + z^{-3}) \cdot \frac{z^{-1}}{1 - bz^{-1}} \cdot \frac{z^{-2}}{1 - cz^{-1} - cz^{-2}}$$

- Voor welke waarden van a is $H(z)$ strikt stabiel? Voor elke reële a .

- En voor welke waarden van b ?

$$|b| < 1.$$

- Is $H(z)$ stabiel voor $c = 1$? nee.

6. Geef de overdrachtsfunctie $H(z)$ van bijgaand netwerk:

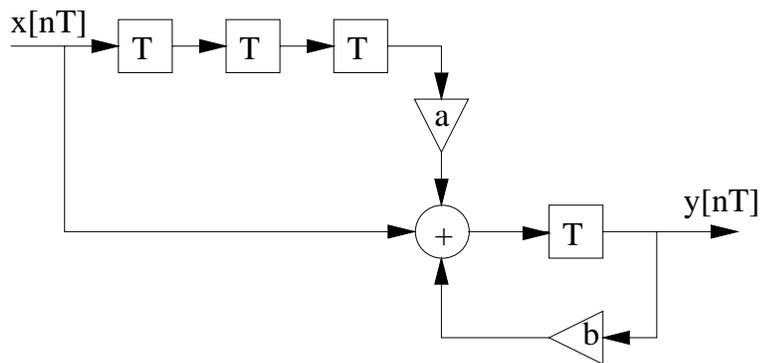


Figure 2.18: figuur hoort bij vraag 6

$$H(z) = \frac{z^{-1} + az^{-4}}{1 - bz^{-1}}$$

7. Teken bijgaand netwerk na transpositie:

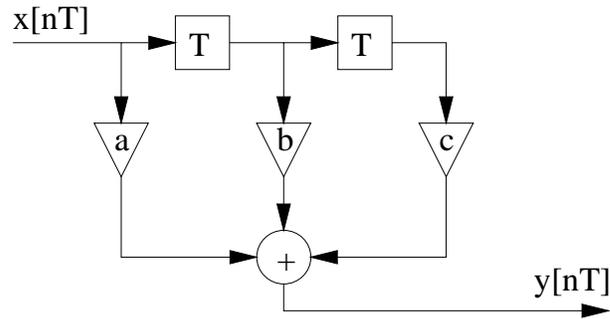


Figure 2.19: figuur hoort bij vraag 7

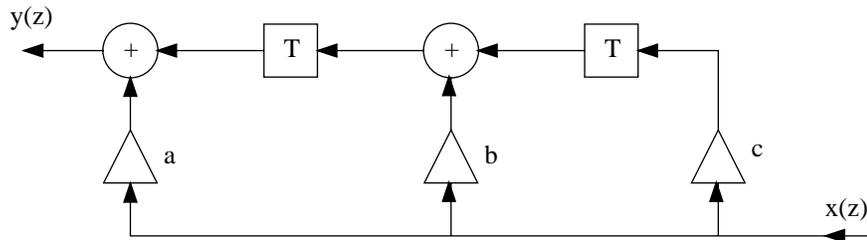


Figure 2.20: figuur hoort bij vraag 7

8. • Kan een NRDF instabiel zijn? nee.

• Waar mogen de polen van een stabiel IIR - filter liggen?

binnen de eenheidscirkel in het z -vlak.

• Welk type filter heeft slechts polen in de oorsprong van het z -vlak?

een nietrekursief filter.

9. Welke van de onderstaande structuren zijn altijd stabiel?

- Een NRDF structuur. wel
- Een IIR structuur. niet
- Een RDF structuur met eindige impulsresponsie. wel
- Een FIR structuur. wel

10. Gegeven is:

$$H(z) = \frac{z + a}{(z + b)(z^2 + cz + d)} = \frac{1 + az^{-1}}{1 + bz^{-1}} \cdot \frac{z^{-2}}{1 + cz^{-1} + dz^{-2}} = \frac{z^{-1}}{1 + bz^{-1}} \cdot \frac{z^{-1} + az^{-2}}{1 + cz^{-1} + dz^{-2}}$$

Teken twee verschillende kanonieke kaskade realisaties van $H(z)$, elk bestaande uit een kaskadering van een eerste- en een tweede orde sectie.

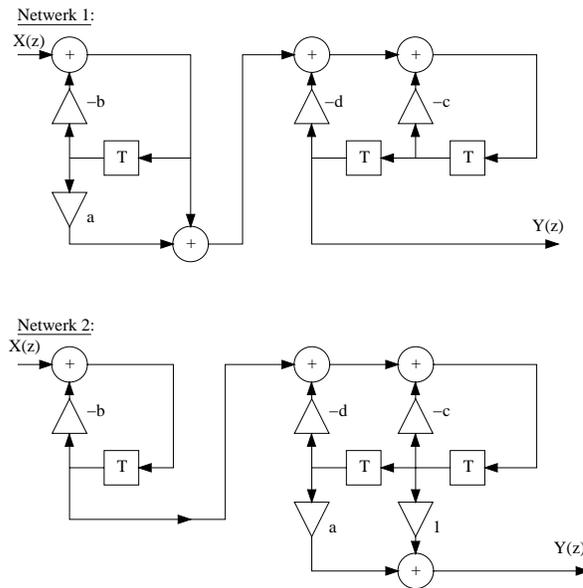


Figure 2.21: figuur hoort bij vraag 10

11. Gegeven is de overdrachtsfunctie:

$$H(z) = \frac{T(z)}{N(z)} = \frac{T(z)}{1 + 0.5z^{-1} + 0.25z^{-2}}$$

Gevraagd: Bewijs de stabiliteit van $H(z)$ door de stabiliteit te onderzoeken van de all-pass:

$$G(z) = \frac{z^{-2}N(1/z)}{N(z)}$$

Gebruik de rekursieve formule:

$$z^{-1}G_{n-1}(z) = \frac{G_n(z) - n_n}{1 - n_n G_n(z)}$$

Geef de resultaten waaruit u de stabiliteit concludeert.

Geef het netwerk dat u gevonden heeft voor $G_2(z)$.

Uitwerking:

$$N(z) = 1 + 0.5z^{-1} + 0.25z^{-2} \quad z^{-2}N\left(\frac{1}{z}\right) = z^{-2} + 0.5z^{-1} + 0.25.$$

$$G_2(z) = \frac{z^{-2} + 0.5z^{-1} + 0.25}{1 + 0.5z^{-1} + 0.25z^{-2}} = \frac{T_1}{N_1} \quad n_2 = 0.25 < 1.$$

$$z^{-1}G_1(z) = \frac{\frac{T_1}{N_1} - 0.25}{1 - 0.25\frac{T_1}{N_1}} \quad G_1(z) = \frac{z^{-1} + \frac{6}{15}}{1 + \frac{6}{15}z^{-1}}$$

$$G_0(z) = 1. \quad n_1 = \frac{6}{15} < 1. \quad G_1(z) = \textit{stabiel}.$$

Hieruit volgt dus uiteindelijk: $H(z)$ is stabiel.

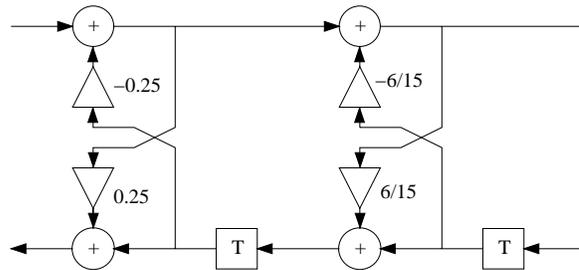
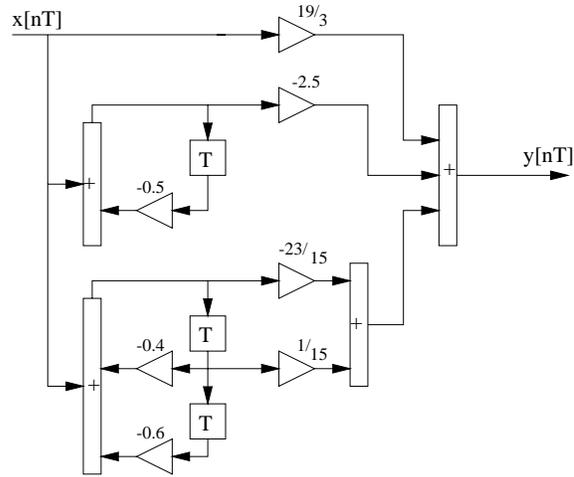
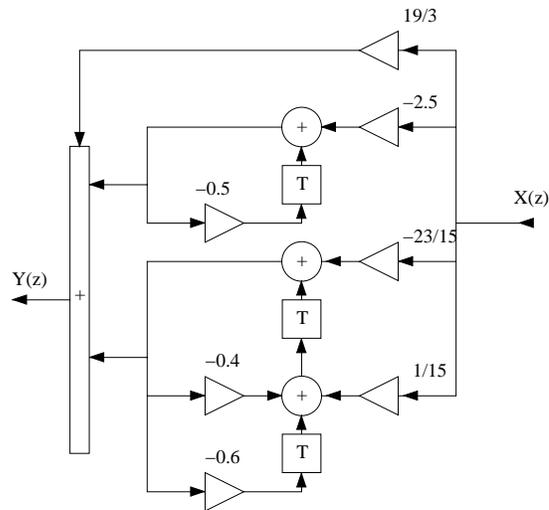


Figure 2.22: figuur hoort bij vraag 11

12. Bereken $H(z)$ van bijgaande schakeling en teken ook de schakeling die ontstaat na transpositie.



$$H(z) = \frac{19}{3} - \frac{5}{2 + z^{-1}} - \frac{23 - z^{-1}}{3(5 + 2z^{-1} + 3z^{-2})} = \frac{23 + 40z^{-1} + 36z^{-2} + 19z^{-3}}{(2 + z^{-1})(5 + 2z^{-1} + 3z^{-2})}$$



Chapter 3

Ontwerpmethoden voor tijdsdiskrete filters.

3.1 Introductie

Een filter, en dus ook een tijdsdiskreet filter, heeft tot doel de frekwentie inhoud van het signaal aan de ingang van het filter te veranderen. De nadruk valt meestal slechts op één van de twee mogelijkheden: óf amplitude- óf faseverloop verandering.

Een filter wordt vaak gespecificeerd aan de hand van het verloop van de amplitude van het uitgangssignaal dat gewenst wordt als aan de ingang een voor elke frekwentie gelijke amplitude wordt aangesloten.

Soms valt de nadruk op het faseverloop: in het geval van nietrekursieve tijdsdiskrete filters is het namelijk mogelijk exact lineair faseverloop te realiseren.

In dit hoofdstuk zullen we achtereenvolgens enkele methoden van ontwerp bespreken.

In paragraaf 3.2 komt lineair faseverloop in nietrekursieve tijdsdiskrete filters aan de orde.

In paragraaf 3.3 bespreken we een methode die een amplitudekarakteristiek oplevert bij lineair faseverloop en dus geschikt is voor nietrekursieve tijdsdiskrete realisatie.

In paragraaf 3.4 bespreken we de methode van impuls response invariantie: De bepaling van een tijdsdiskrete overdrachtsfunctie in z als de gewenste impuls response de bemonsterde versie is van een gegeven tijdscontinue impuls response.

Tenslotte bespreken we in paragraaf 3.5 de bilineaire transformatie. Met behulp van deze transformatie kunnen we ontwerpmethoden die ontwikkeld zijn

voor amplitude karakteristieken in het tijdscontinue domein, direkt toepassen in het tijdsdiskrete domein.

3.2 Nietrekursieve filters met exact lineair faseverloop.

Het blijkt dat we hier onderscheid dienen te maken in 4 gevallen:

1. Het aantal coefficienten van de impuls response $h[nT]$ is even en symmetrisch t.o.v. een punt op de tijdsas.
2. $h[nT]$ is even en antisymmetrisch.
3. $h[nT]$ is oneven en symmetrisch.
4. $h[nT]$ is oneven en antisymmetrisch.

In bijgaande figuur 3.1 zijn deze 4 mogelijkheden aangegeven.

1. **Geval 1:** impulsresponse symmetrisch, aantal coefficienten even:

We kunnen de impulsresponse schrijven als:

$$h[nT] = a_4\delta[nT] + a_3\delta[nT - T] + a_2\delta[nT - 2T] + a_1\delta[nT - 3T] + a_1\delta[nT - 4T] + a_2\delta[nT - 5T] + a_3\delta[nT - 6T] + a_4\delta[nT - 7T].$$

$$H(e^{j\theta}) = a_4 + a_3e^{-j\theta} + a_2e^{-j.2\theta} + a_1e^{-j.3\theta} + a_1e^{-j.4\theta} + a_2e^{-j.5\theta} + a_3e^{-j.6\theta} + a_4e^{-j.7\theta}.$$

$$H(e^{j\theta}) = a_1e^{-j.\frac{7}{2}\theta}(e^{j.\frac{1}{2}\theta} + e^{j.-\frac{1}{2}\theta}) + a_2e^{-j.\frac{7}{2}\theta}(e^{j.\frac{3}{2}\theta} + e^{j.-\frac{3}{2}\theta}) + a_3e^{-j.\frac{7}{2}\theta}(e^{j.\frac{5}{2}\theta} + e^{j.-\frac{5}{2}\theta}) + a_4e^{-j.\frac{7}{2}\theta}(e^{j.\frac{7}{2}\theta} + e^{j.-\frac{7}{2}\theta}).$$

$$H(e^{j\theta}) = e^{-j.\frac{7}{2}\theta}.2.(a_1\cos\frac{1}{2}\theta + a_2\cos\frac{3}{2}\theta + a_3\cos\frac{5}{2}\theta + a_4\cos\frac{7}{2}\theta).$$

2. **Geval 2:** aantal coefficienten even en antisymmetrie:

Op dezelfde manier als bij geval 1 krijgen we nu:

$$H(e^{j\theta}) = 2j.e^{-j.\frac{7}{2}\theta}(b_1\sin\frac{1}{2}\theta + b_2\sin\frac{3}{2}\theta + b_3\sin\frac{5}{2}\theta + b_4\sin\frac{7}{2}\theta).$$

3. **Geval 3:** aantal coefficienten oneven en symmetrie:

$$H(e^{j\theta}) = e^{-j.\frac{(9-1)}{2}\theta}[c_0 + 2(c_1\cos\theta + c_2\cos2\theta + c_3\cos3\theta + c_4\cos4\theta)].$$

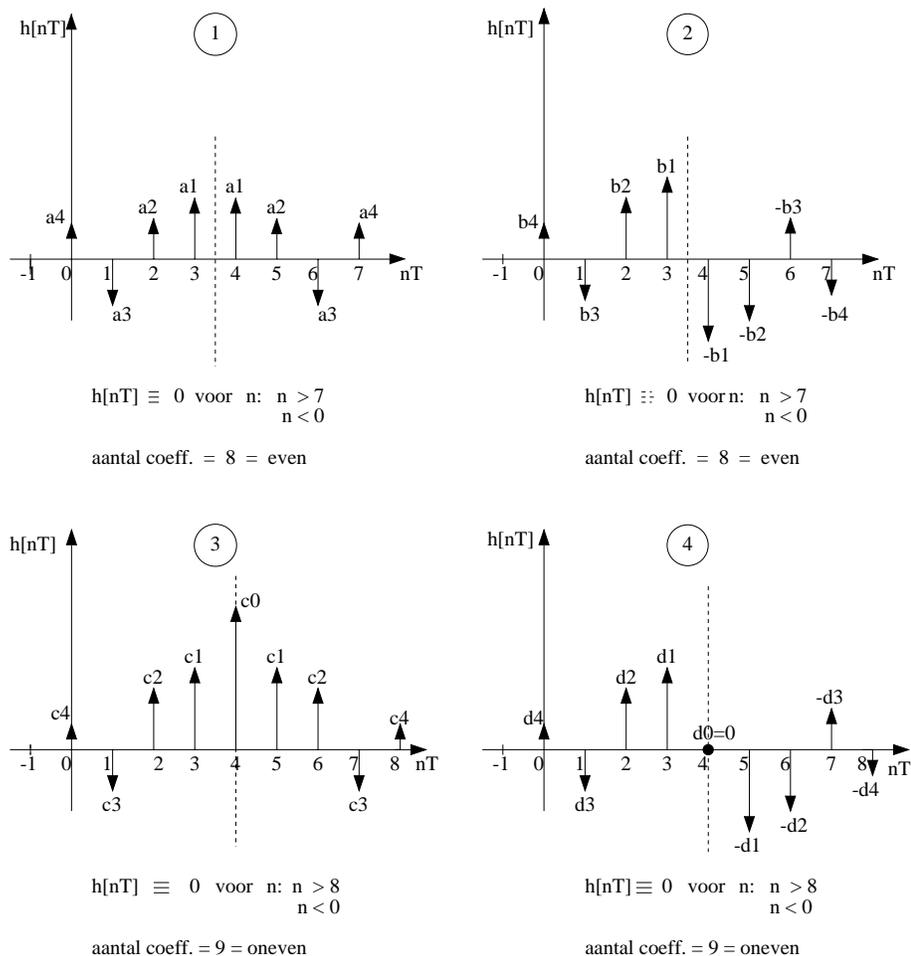


Figure 3.1:

4. Geval 4: aantal coëfficiënten oneven en antisymmetrie:

$$H(e^{j\theta}) = e^{-j \cdot \frac{(\theta-1)}{2} \theta} \cdot 2j(d_1 \sin\theta + d_2 \sin 2\theta + d_3 \sin 3\theta + d_4 \sin 4\theta).$$

Voor bovenstaande gevallen geldt:

1. Als $\theta = 0 \Rightarrow \sin.. \theta = 0, \cos.. \theta = 1$.

Dus: geval 1 + 3 levert een amplitude met laagdoorlaat karakter, geval 2 + 4 levert géén laagdoorlaat karakter.

2. We hebben in alle 4 gevallen exact lineair faseverloop.

3. Een generalisatie naar de impulsresponse van willekeurige, maar eindige, lengte is eenvoudig te maken.

3.3 Ontwerp van nietrekursieve overdrachtsfuncties met exact lineair faseverloop door middel van de Fourier transformatie en gebruik van vensters

De gang van zaken is geschetst in figuur 3.2. We willen, als voorbeeld, een FIR filter vinden dat het laagdoorlaat karakter dat bovenaan in de figuur getekend is (in het fundamenteel interval tussen $-\pi < \theta < \pi$).

De Inverse Fourier Transformatie voor tijdsdiskrete signalen levert de impulsresponse $h_d[n]$ op.

Deze heeft 2 vervelende eigenschappen:

1. Is meestal oneindig lang.
2. Is niet-causaal (begint vóór $n < 0$).

Dit maakt het onmogelijk deze impulsresponse exact te realiseren met een FIR filter.

We nemen daarom genoegen met 2 benaderingen:

1. We maken de impulsresponse eindig van lengte. Dit levert $h[n]$. Deze is wél symmetrisch.
2. We passen een "shift" toe. We zorgen er voordat $h[n]$ pas $\neq 0$ wordt bij $n \geq 0$.

We hebben nu een impulsresponse die eindig is van lengte en bovendien een symmetrie vertoont.

Een FIR filter met exact lineair faseverloop is dus mogelijk!

We hebben nu echter slechts een benadering gemaakt van het laagdoorlaat karakter waar we van uit gingen!

Wat er gebeurd is, is verduidelijkt in figuur 3.3.

Vermenigvuldiging van de impulsresponse met een rechthoekig venster (dit deden we toen de impulsresponse eindig van lengte werd gemaakt) heeft in het frekwentie domein een convolutie tot gevolg. Op deze wijze krijgen we inzicht in het gevolg van de benadering die we toepasten.

Het gebruik van een rechthoekig venster is niet noodzakelijk de beste methode om een benadering te krijgen met eindige impulsresponse.

In de praktijk zijn dan ook verschillende andere vensters in gebruik. Ze leveren alle een iets andere benadering van de ideale laagdoorlaat van een bepaalde orde.

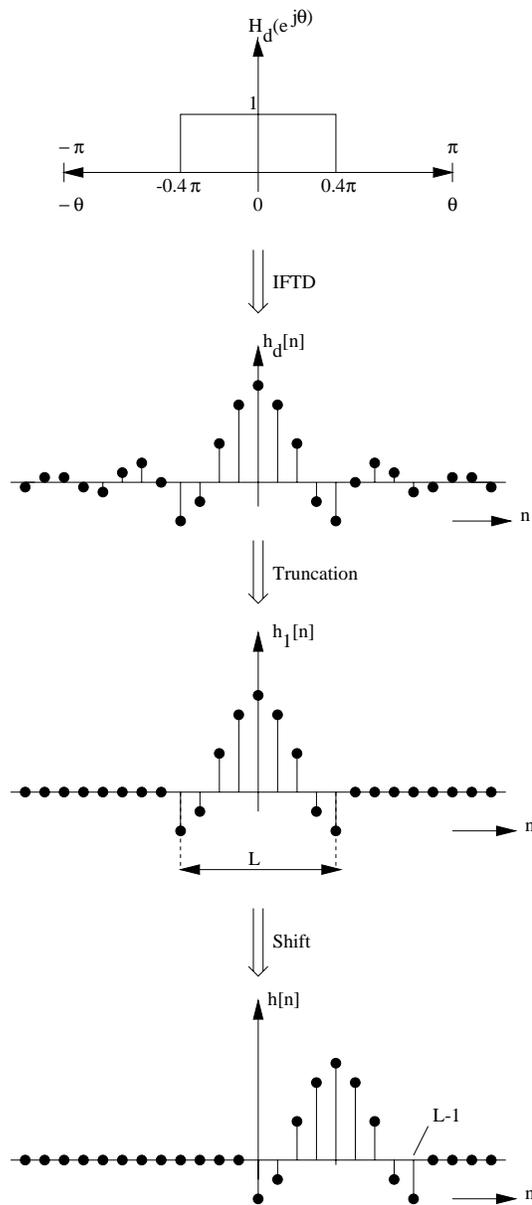


Figure 3.2:

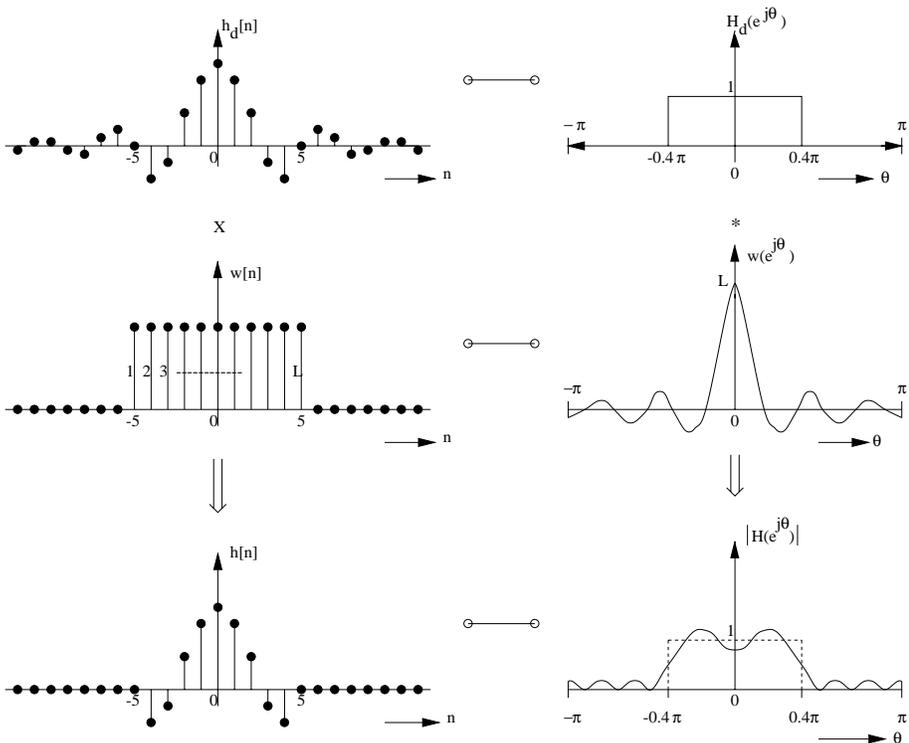


Figure 3.3:

Enkele venster funkties worden hieronder gegeven:

1. Het rechthoekige venster:

$$w[n] = \begin{cases} 1 & \text{voor } 0 \leq n \leq L-1 \\ 0 & \text{in de andere gevallen} \end{cases}$$

2. Hanning venster:

$$w[n] = \begin{cases} \frac{1}{2} \{1 - \cos(\frac{2\pi n}{L-1})\} & \text{voor } 0 \leq n \leq L-1 \\ 0 & \text{in de andere gevallen} \end{cases}$$

3. Bartlett venster:

$$w[n] = \begin{cases} \frac{2n}{L-1} & \text{voor } 0 \leq n \leq \frac{L-1}{2} \\ 2 - \frac{2n}{L-1} & \text{voor } \frac{L-1}{2} \leq n \leq L-1 \\ 0 & \text{in de andere gevallen} \end{cases}$$

3.4 Methode van impuls-invariantie

Hierbij gebruiken we een analoog filter met frekwentie karakteristiek $H_a(\omega)$, overdrachtsfunctie $H_a(s)$ en impuls response $h_a(t)$.

De bedoeling is om een IIR filter te vinden met een impuls response die de bemonstering is van $h_a(t)$:

$$h_d[n] = h_a(nT)$$

De gevolgen in het frekwentie domein zijn geschetst in figuur 3.4.

De frekwentie karakteristiek die hoort bij $h_d[n]$ is periodiek en kan dus nooit gelijk zijn aan $H_a(\omega)$.

De vervorming van $H_a(\omega)$ heet aliasing.

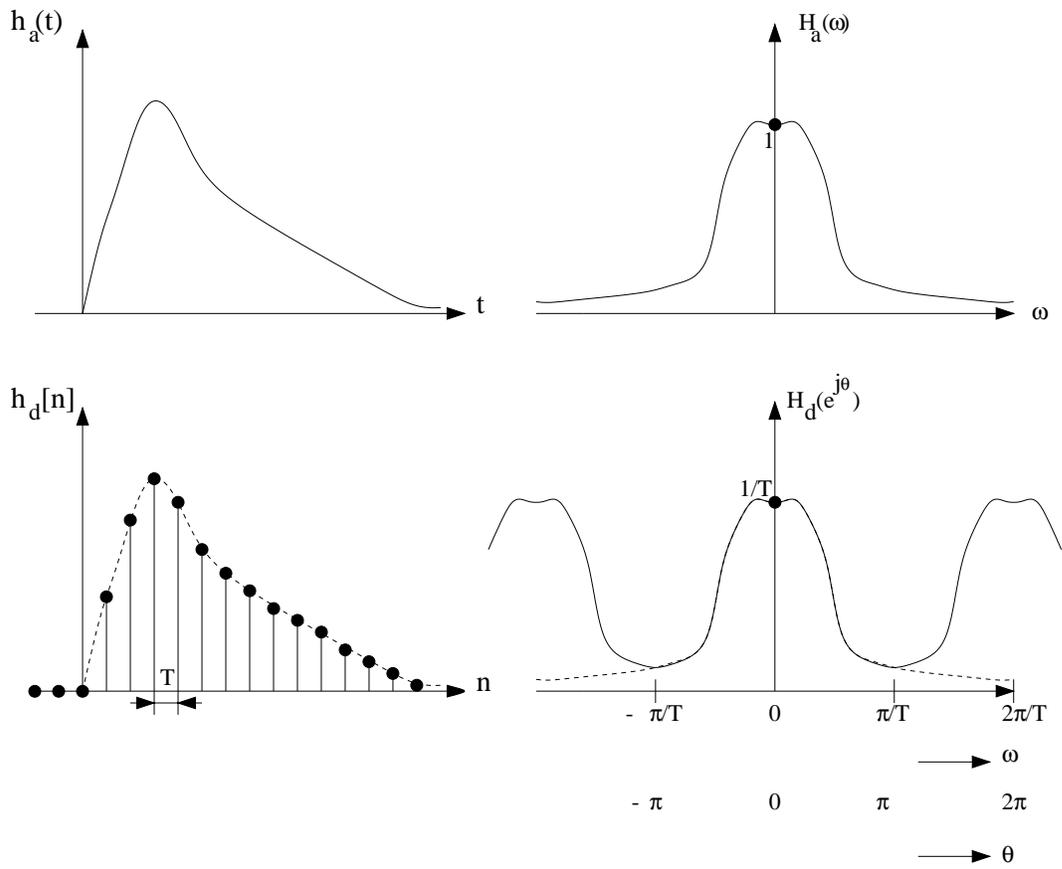


Figure 3.4:

Voorbeeld:

$$H_a(s) = \frac{A}{s - B} \Rightarrow H_a(\omega) = \frac{A}{j\omega - B}$$

Hieruit volgt: $h_a(t) = \begin{cases} Ae^{Bt} & \text{voor } t \geq 0 \\ 0 & \text{voor } t < 0 \end{cases}$

Dus volgt:

$$h_d[n] = h_a(nT) = Ae^{BnT}u[n]$$

Dit naar het z-vlak getransformeerd levert:

$$H_d(z) = \sum_{n=-\infty}^{\infty} h_d[n]z^{-n} = A \sum_{n=0}^{\infty} (e^{BT}z^{-1})^n = \frac{A}{1 - e^{BT}z^{-1}}$$

Dit levert de realisatie van figuur 3.5.

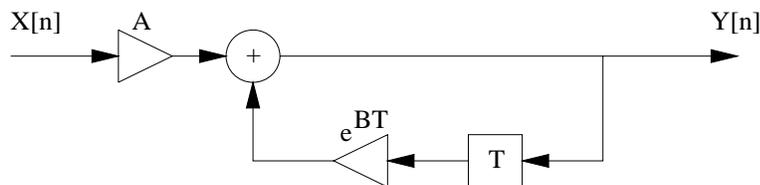


Figure 3.5:

Dus: als $H_a(s)$ een pool heeft bij $s=B$, heeft $H_d(z)$ een pool bij $z = e^{BT}$.

Dit kunnen we generaliseren als:

$$H_a(p) = \sum_{k=1}^N \frac{A_k}{s - B_k} \Rightarrow H_d(z) = \sum_{k=1}^N \frac{A_k}{1 - e^{B_k T} z^{-1}}$$

Dit geldt ook voor reële polen B_k , evenals voor toegevoegd complexe polen.

3.5 De bilineaire transformatie methode

Dit is de meest bekende, en gebruikte, methode om vanuit een tijdscontinue overdrachtsfunctie van een filter, de overdrachtsfunctie van een tijdsdiskreet filter te vinden met een corresponderende frekwentie karakteristiek.

We hebben de transformatie:

$$w = \frac{z-1}{z+1} = \frac{1-z^{-1}}{1+z^{-1}} = \frac{1-e^{-sT}}{1+e^{-sT}} = \tanh\left(\frac{sT}{2}\right)$$

($w = \frac{z-1}{z+1}$ heet de bilineaire transformatie).

De inverse transformatie:

$$z = -\frac{w+1}{w-1}$$

We hebben hier het verband gelegd tussen drie (3) complexe vlakken:

Het s-vlak, het z-vlak en het w-vlak.

Met de relaties:

$$\omega_s = 2\pi f_s = \frac{2\pi}{T}$$

$$\omega T = \omega \cdot \frac{1}{f_s} = \frac{\omega}{\omega_s} \cdot 2\pi$$

kunnen we de belangrijkste verbanden snel inzien.

De 3 complexe vlakken zijn getekend in figuur 3.6.

We leren uit de figuur 3.6 dat het linkerdeel van een periode uit het s-vlak met langs de $j\omega$ -as het fundamentele frekwentie-interval eenduidig wordt afgebeeld op de ruimte binnen de eenheidskring in het z-vlak en dit wordt weer eenduidig afgebeeld op het linkerdeel van het w-vlak.

Als we schrijven:

$$w = u + jv \quad \text{en} \quad T = \frac{2\pi}{\omega_s}$$

dan volgt langs de imaginaire as in het w-vlak ($u=0$):

$$v = \tan\left(\frac{\omega T}{2}\right) = \tan\left(\frac{\omega}{\omega_s} \cdot \pi\right)$$

Dit geeft een tan - verband van figuur 3.7.

Dus de gehele imaginaire as in het complexe w-vlak wordt afgebeeld op het fundamentele frekwentie-interval langs de imaginaire as in het complexe s-vlak waar voor geldt:

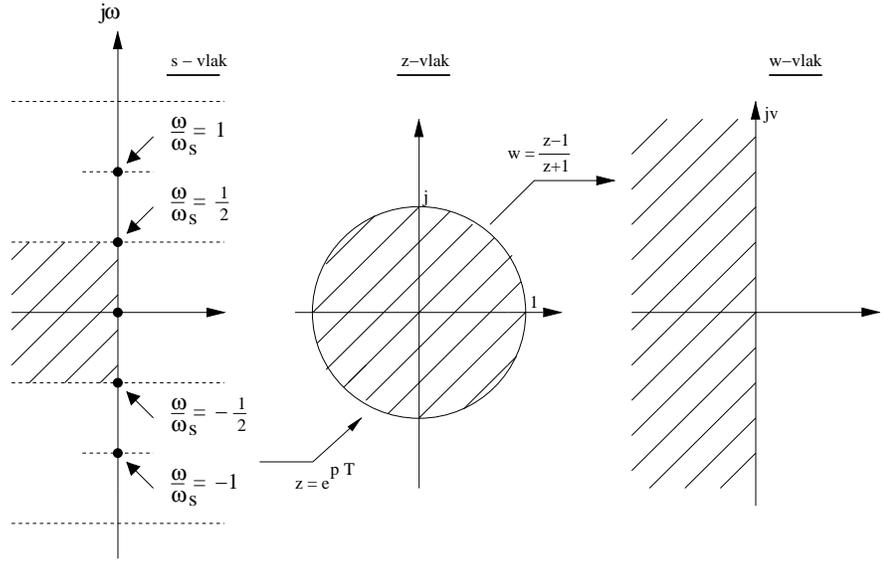


Figure 3.6:

$$\left| \frac{\omega}{\omega_s} \right| \leq \frac{1}{2}$$

Als we dus een tijdscontinue overdrachtsfunctie $H(s)$ hebben met een bepaald gewenst gedrag langs de imaginaire as (frequentie karakteristiek) dan kunnen we formeel s vervangen door w en dan de bilineaire transformatie toepassen. We hebben dan een $H(z)$ die langs de eenheidscirkel een gedrag vertoont dat gelijkvormig is aan dat van het tijdscontinue filter, maar dan periodiek geworden.

Voorbeeld:

$$H(w) = \frac{1}{w^3 + w^2 + 2w + 1} = \frac{1}{(w + 1)(w^2 + w + 1)}$$

Van $H(w)$ heeft $|H(v)|^2$ de vorm van figuur 3.8.

De bilineaire transformatie levert $H(z)$ met:

$$\begin{aligned} H(z) &= \frac{z^3 + 3z^2 + 3z + 1}{6z^3 + 2z} = \frac{z^{-3} + 3z^{-2} + 3z^{-1} + 1}{2z^{-2} + 6} \\ &= \frac{(z + 1)(z^2 + 2z + 1)}{2z(3z^2 + 1)} = \frac{(1 + z^{-1})(z^{-2} + 2z^{-1} + 1)}{2(z^{-2} + 3)} \end{aligned}$$

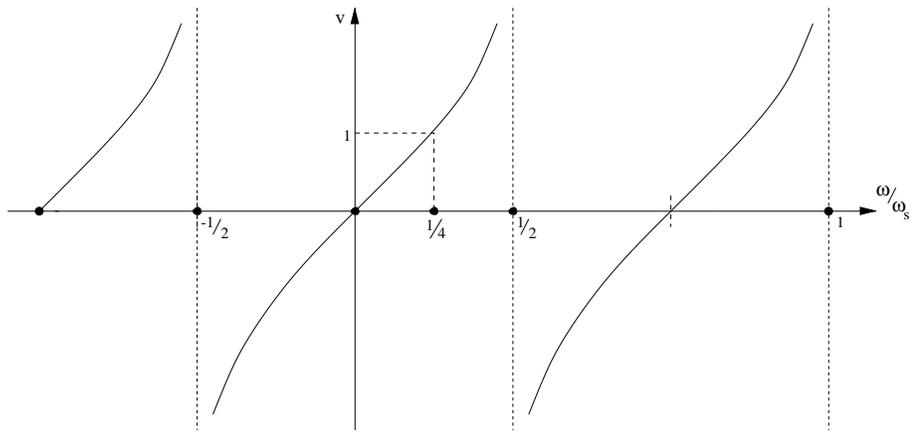


Figure 3.7:

De frekwentie karakteristiek wordt gegeven in figuur 3.9.

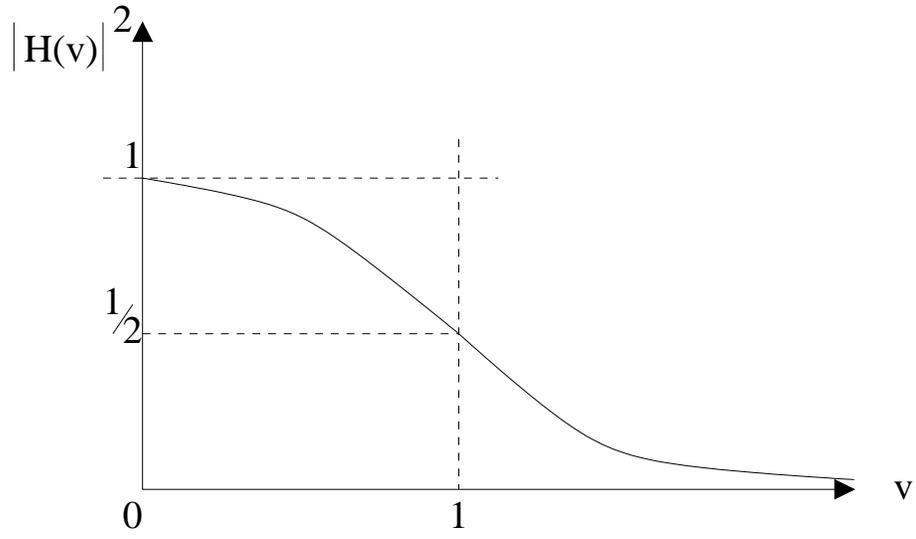


Figure 3.8:

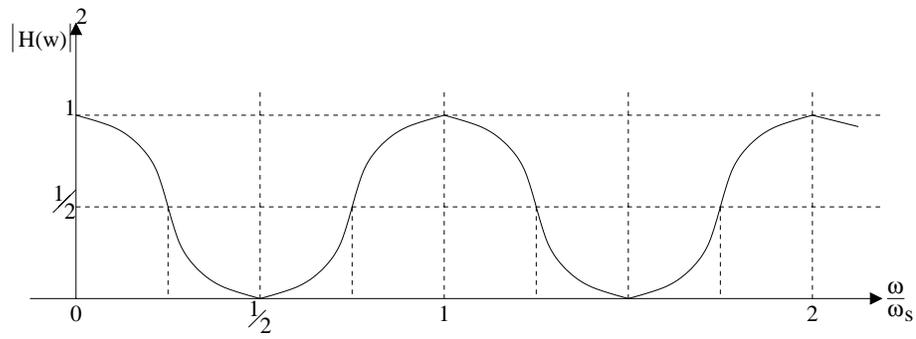


Figure 3.9:

3.6 Verliesvrijheid in het tijdsdiskrete geval

Om het eenvoudig te houden: we kunnen in het voorgaande de complexe variabele s vervangen door w . (met $w = \frac{z-1}{z+1}$, de bilineaire transformatie)

Een n bij n matrix S in z heet passief als geldt:

$$S^T(z) \cdot S(z^{-1}) \leq 1$$

Een n bij n matrix S in z heet orthogonaal(verliesvrij) als geldt:

$$S^T(z) \cdot S(z^{-1}) = 1$$

Voorbeeld:

Een voorbeeld van een eerste orde, orthogonale, tijdsdiskrete tweepoort is hieronder gegeven in figuur 3.10.

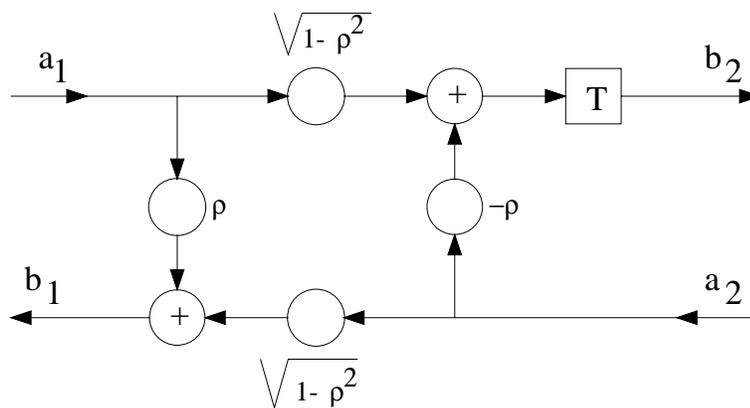


Figure 3.10: een tijdsdiskrete tweepoort

Voor figuur 3.10 hebben we:

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} \rho & \sqrt{1-\rho^2} \\ z^{-1}\sqrt{1-\rho^2} & -\rho z^{-1} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

Voor de tweepoortmatrix $H(z)$ geldt:

$$H^T(z) \cdot H(z^{-1}) = \begin{bmatrix} \rho & z^{-1}\sqrt{1-\rho^2} \\ \sqrt{1-\rho^2} & -\rho z^{-1} \end{bmatrix} \begin{bmatrix} \rho & \sqrt{1-\rho^2} \\ z\sqrt{1-\rho^2} & -\rho z \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Stelling:

Een scalaire, verliesvrije overdrachtsfunctie $H(z)$ van graad n heeft de vorm:

$$H(z) = \eta \frac{a_0 z^n + a_1 z^{n-1} + \dots + 1}{z^n + a_{n-1} z^{n-1} + \dots + a_0} \quad \text{met } \eta = \pm 1$$

de nulpunten van het noemerpolynoom

$a(z) = z^n + a_{n-1} z^{n-1} + \dots + a_0$ liggen alle binnen de eenheidscirkel.

Eind.

Bovenstaande volgt vrijwel direkt uit de verliesvrijheidseis. $H(z)$ wordt een digitale all-pass functie genoemd.

De stelling levert ons dus niet direkt de mogelijkheid om een willekeurige overdrachtsfunctie $H(z)$ te realiseren als een orthogonaal (verliesvrij) filter.

Hiertoe kunnen we terug naar de analoge, verliesvrije tweepoortsituatie.

We zouden $H(z)$ kunnen beschouwen als een onderdeel (b.v. S_{11} of S_{21}) van een tweepoortmatrix S die zelf wel orthogonaal is.

($H(z)$ moet wel voldoen aan de eisen die worden gesteld aan S_{11} of S_{21}).

Dit wordt in de literatuur genoemd: embedding.

Gebruik makend van de Belevitch-polynomen notatie kunnen we vanuit de onvolledige specificatie van S d.m.v. $H(z)$ de volledige beschrijving van S proberen te verkrijgen.

Een realisatie van $H(z)$ verkrijgen we dan door in de realisatie van $S(z)$: $a_2 = 0$ te stellen en de output b_2 te negeren (we hebben dan $H(z)$ gerealiseerd als een reflectie S_{11}).

Blijft over een realisatie van $S(z)$ te vinden. Hiervoor zijn kaskadesynthesemethoden ontwikkeld (direkt in z) die we hier niet zullen behandelen.

Een voorbeeld van het ontstaan van een verliesvrije overdrachtsfunctie $H(z)$:

Recept:

1. Bereken de reflektiefunctie $\rho(s)$ van een verliesvrije impedantie $Z(s)$.
2. Transformeer deze $\rho(s)$ met behulp van de bilineaire transformatie naar het z -vlak: $\rho(z)$.

Voorbeeld:

Kies:

$$Z(s) = \frac{k}{s} \quad (\text{een s-vlak capaciteit}).$$

dan is:

$$\rho(s) = \frac{Z(s) - 1}{Z(s) + 1} = \frac{k - s}{k + s}.$$

$s = \frac{z-1}{z+1}$ ingevuld levert:

$$\rho(z) = \frac{z(k-1) + (k+1)}{z(k+1) + (k-1)}.$$

Deze functie voldoet aan de gestelde eis van verliesvrijheid.
De functie heet ook een all-pass functie omdat:

$$|\rho(z)| = 1$$

voor alle z op de eenheidscirkel.

Bij Deel 2, par. 2.6, Uitgewerkte vraagstukken, vraagstuk 1:

Als we invullen: $s = \frac{z-1}{z+1}$ dan wordt:

$$S(z) = S_{(z)}^T = \frac{1}{3z+1} \begin{bmatrix} z-1 & 2(z+1) \\ 2(z+1) & z-1 \end{bmatrix} \quad \text{en:} \quad S_{(z^{-1})} = \frac{1}{3+z} \begin{bmatrix} 1-z & 2(z+1) \\ 2(z+1) & 1-z \end{bmatrix}$$

Controleer zelf dat geldt: $S_{(z)}^T \cdot S_{(z^{-1})} = 1$

We hebben hier dus weer een orthogonale(verliesvrije) 2 x 2 matrix waarin verschillende overdrachtsfuncties voorkomen.

Blijkbaar hebben we hier ook een voorbeeld van embedding van een scalaire overdrachtsfunctie in een orthogonale 2 x 2 matrix.

3.7 Eenvoudige tijdsdiskrete laddersynthese.

We zullen ons hier beperken tot tijdsdiskrete all-pass functies. Dit is een erg belangrijke soort tijdsdiskrete functies omdat later zal blijken dat we met behulp van 2 all-pass netwerken tijdsdiskrete filters kunnen bouwen door deze 2 netwerken parallel te schakelen.

Een all-pass overdrachtsfunctie $H(z)$ ziet er als volgt uit:

$$H(z) = \eta \frac{a_0 z^n + a_1 z^{n-1} + \dots + 1}{z^n + a_{n-1} z^{n-1} + \dots + a_0} \quad \text{met} \quad \eta = \pm 1, a_0 \dots a_{n-1} \text{ is re \u00e9el}$$

waarvoor:

$$\left| H(e^{j\omega}) \right|^2 = 1 \quad \text{voor alle } \omega.$$

en:

$$H(z) \cdot H(z^{-1}) = 1 \quad (\text{verliesvrijheidseis}).$$

Een all-pass functie $H(z)$ kan gerealiseerd worden m.b.v. de klassieke tijdsdiskrete structuren.

Die realisaties waarin de co\u00e9fficienten a_0 t/m a_{n-1} rechtstreeks voorkomen als vermenigvuldigingsco\u00e9fficienten hebben een voordeel boven andere: als om welke reden dan ook de co\u00e9fficienten slechts een beperkte nauwkeurigheid kunnen hebben, blijft de all-pass functie toch intact. Deze is n.l. niet afhankelijk van de co\u00e9fficient zelf, maar van de co\u00e9fficientenstructuur van $H(z)$.

Sinds ≈ 1973 bestaat er nog een andere structuur voor een all-pass functie: De synthese procedure is gegeven door onderstaande rekursieve relatie:

$$z^{-1} H_{m-1}(z) = \frac{H_m(z) - k_m}{1 - k_m H_m(z)} \quad \begin{array}{l} m = 1, 2, \dots, n \\ k_m = H_m(\infty) \end{array}$$

Drie realisatiestructuren die hierbij horen zijn gegeven in figuur 3.11.

De algemene structuur (lattice structure) voor een all-pass functie, op deze manier gerealiseerd, wordt dan zoals gegeven in figuur 3.12.

Voorbeeld 1:

$$H_{1(z)} = \frac{0.3z + 1}{z + 0.3}$$

Hier wordt $k_1 = 0.3$, dan:

$$z^{-1} H_{0(z)} = \frac{\frac{0.3z+1}{z+0.3} - 0.3}{1 - 0.3 \frac{0.3z+1}{z+0.3}} = \frac{0.3z + 1 - 0.3(z + 0.3)}{z + 0.3 - 0.3(0.3z + 1)} = \frac{1}{z} \Rightarrow H_{0(z)} = 1$$

Een mogelijke realisatie wordt figuur 3.13.

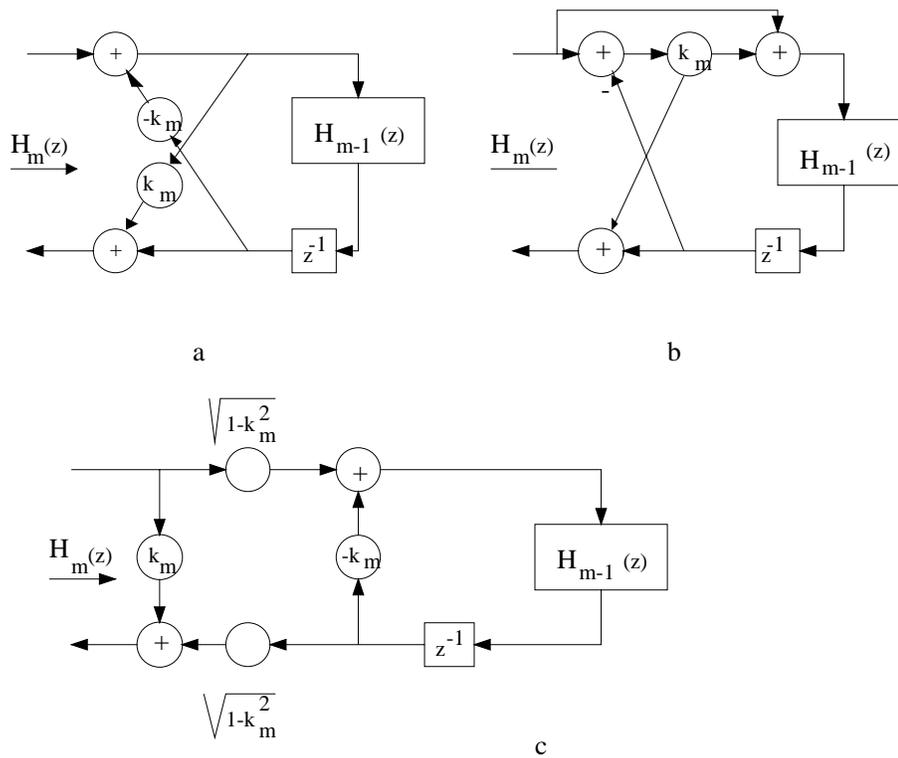


Figure 3.11:

Algemeen:

De structuur van figuur 3.11b levert realisaties met slechts n vermenigvuldigingscoëfficiënten voor een n -de orde all-pass functie.

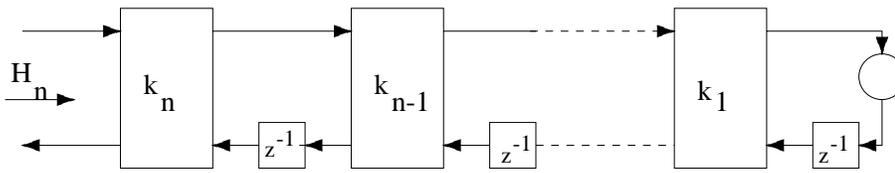


Figure 3.12:

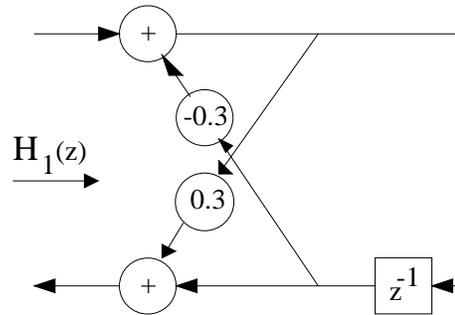


Figure 3.13:

3.8 Vraagstukken

1. Maak de impulsresponsies van figuur 3.14 af tot responsies van een lineair fase filter:

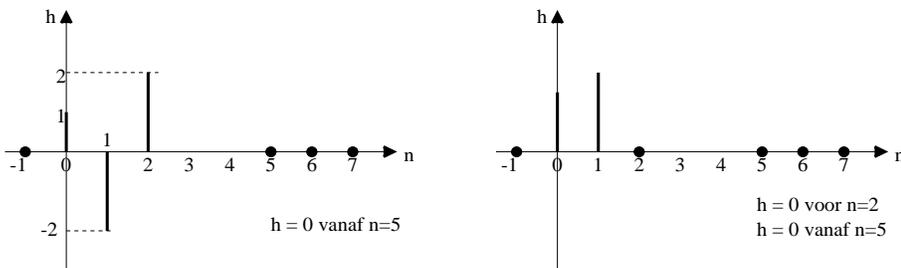


Figure 3.14:

2. Doe hetzelfde met figuur 3.15:

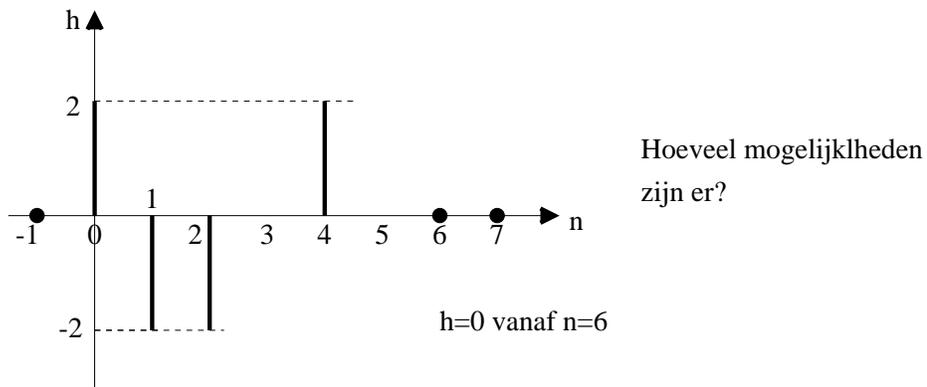


Figure 3.15:

3. Van een FIR filter hebben we:

$$H(z) = -1 + 2z^{-1} + az^{-2} + bz^{-3} - z^{-4} \quad (\text{a en b reëel})$$

- Wat is de waarde van b als $a=0$ en we lineair faseverloop hebben.

- Wat is de waarde van b als $a=1$ en we lineair faseverloop hebben.

4. De frekwentie karakteristiek $H(e^{j\theta})$ is gegeven als:

$$H(e^{j\theta}) = \begin{cases} 1 & \text{voor } |\theta| < \pi/3 \\ 0 & \text{voor } \pi/3 < |\theta| < \pi \end{cases}$$

- Gebruik de *IFTD* voor de bepaling van $h[n]$.

$$\left(h[n] = \frac{1}{2\pi} \int_{-\pi}^{+\pi} H(e^{j\theta}) e^{j\theta n} d\theta \right)$$

- Bepaal uit $h[n]$ een $h_1[n]$ van lengte 9 van een causaal filter met lineaire fase.

5. Bepaal $h[n]$, $H(z)$ en $H(e^{j\theta})$ van de netwerken van figuur 3.16. Teken de amplitude- en fase karakteristiek.

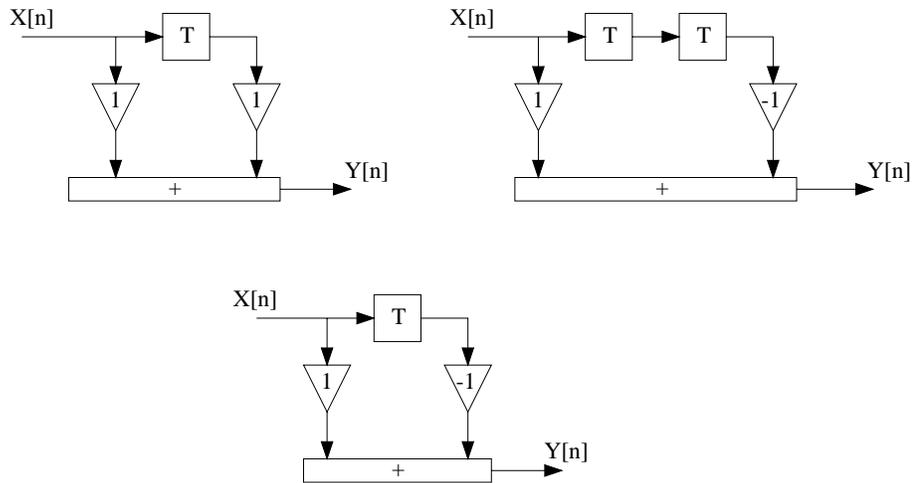


Figure 3.16:

6. Gegeven is $H(s)$ van een tijdscontinue filter:

$$H(s) = \frac{5s + 7}{(s + 1)(s + 2)}$$

- Bereken $H_1(z)$ volgens de methode van impulsinvariantie ($T = \frac{1}{4}$).

- Wat kunt u zeggen over de overeenkomsten en verschillen tussen:

(a) de impulsresponsies van $H_1(z)$ en $H(s)$.

(b) de frekwentiekarakteristieken van $H_1(z)$ en $H(s)$.

- Geef een realisatie van $H_1(z)$.

7. Bereken vanuit onderstaande overdrachtsfuncties $H_1(s)$ en $H_2(s)$ m.b.v. de impulsinvariantie ($T = \frac{1}{4}$) de overdrachtsfuncties $H_1(z)$ en $H_2(z)$.

$$H_1(s) = \frac{1}{s+1} - \frac{2}{s+2} + \frac{1}{s+3}$$

$$H_2(s) = \frac{2}{(s+1)(s+2)(s+3)}$$

Chapter 4

Digitale golffilters

4.1 Inleiding

In de delen 1 en 2 van dit diktaat hebben we een aantal begrippen en methoden besproken die gezamenlijk verantwoordelijk zijn voor het succes van de verliesvrije laddernetwerken afgesloten in weerstanden.

Hoe kunnen we dit gebruiken in het digitale geval?

De belangrijkste gedachtensprong houdt in dat we de complexe variabele s vervangen door de complexe variabele w (de bilineaire variabele $w = \frac{z-1}{z+1}$).

Dat houdt in dat we de benaderingen gevonden in deel 1 natuurlijk ook kunnen gebruiken langs de imaginaire as in het w -vlak. Dit levert dan overdrachtsfuncties eerst in w en dan in z .

Deel 2 leverde laddernetwerken. Ook hier kunnen we stellen dat dit evengoed ladder netwerken zijn bestaande uit w -vlak netwerkelementen. Ook de S -matrix theorie uit deel 2 geldt voor deze w -vlak netwerken.

In dit hoofdstuk zullen we een methode bespreken die deze netwerken vertaalt in berekenbare digitale structuren, uitgaande dus van w -vlak laddernetwerken afgesloten in w -vlak weerstanden. De digitale uitwerking van deze netwerken noemen we digitale golffilters.

4.2 De vertaalprocedure

In plaats van de golfvariabelen, zullen we hier gebruik maken van de zogenaamde spanningsgolfvariabelen. Dit levert uiteindelijk digitale netwerken op met eigenschappen die veel lijken op die van de analoge ladders.

De definitie van de spanningsgolfvariabelen is:

$$A_k = U_k + R_k I_k \quad (4.1)$$

$$B_k = U_k - R_k I_k \quad (4.2)$$

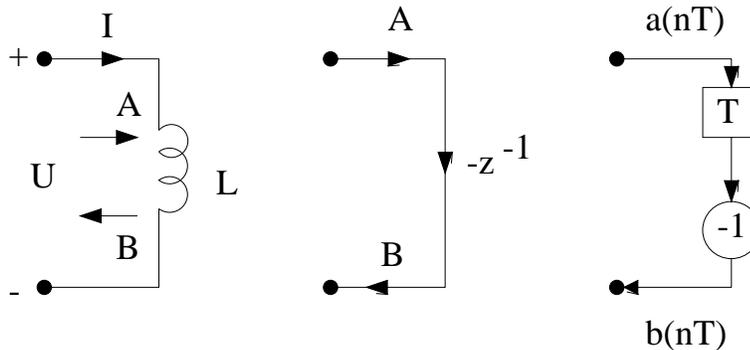


Figure 4.1: vertaling van de zelfinductie

Hierin is R_k de referentieweerstand, een positieve konstante voor elke poort k .
(vergelijk dit met de definities van de vermogensgolfvariabelen).

4.2.1 De w -vlak zelfinductie

We hebben de relatie volgens figuur 4.1:

$$U = w.L.I \quad (4.3)$$

Via de bilineaire transformatie wordt dat:

$$U = \frac{z-1}{z+1}.L.I \quad (4.4)$$

Introductie van de spanningsgolfvariabelen A en B met de keus: $R = L$ levert:

$$B = -\frac{1}{z}.A \quad (4.5)$$

In (4.5) lijkt de zelfinductie L te zijn verdwenen, deze zit echter in de gebruikte definitie van A en B .

4.2.2 De w -vlak capaciteit

Hier resulteert figuur 4.2.

Uit $U = \frac{1}{wC}.I$ en gebruik van de variabelen A en B met daarin $R = \frac{1}{C}$ volgt:

$$B = \frac{1}{z}.A \quad (4.6)$$

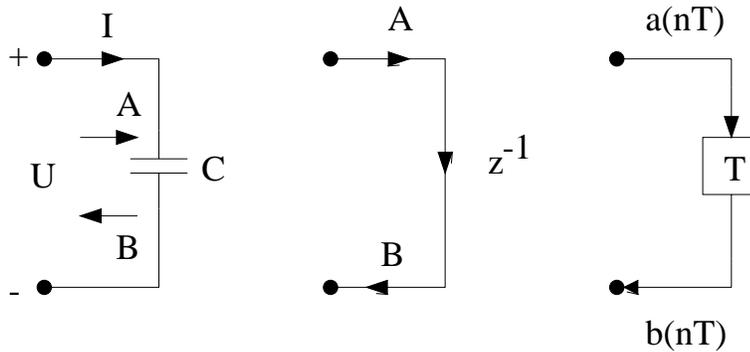


Figure 4.2: vertaling van de capaciteit

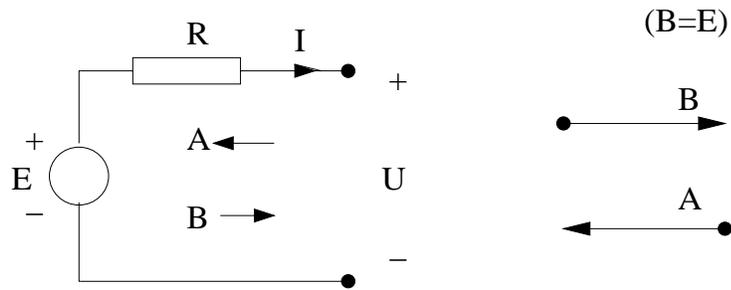


Figure 4.3: vertaling van de bron met weerstand

4.2.3 De w-vlak bron

Hier resulteert figuur 4.3.

4.2.4 De w-vlak belastingsweerstand

Hier resulteert figuur 4.4.

De vergelijking $U + RI = 0$ levert (met referentie R):

$$B = 0 \quad (4.7)$$

4.2.5 De driepoort parallel verbinding

In de voorgaande secties hebben we de belangrijkste elementen 'vertaald', blijft over het verbindingsnetwerk. Met behulp van de driepoort parallel - en

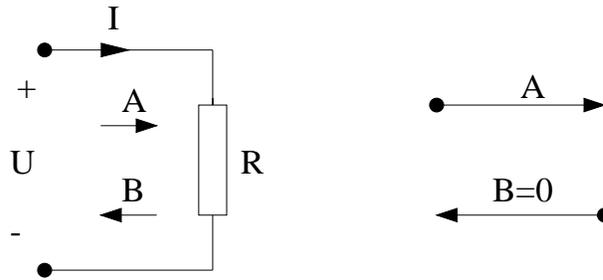


Figure 4.4: vertaling van de belastingsweerstand

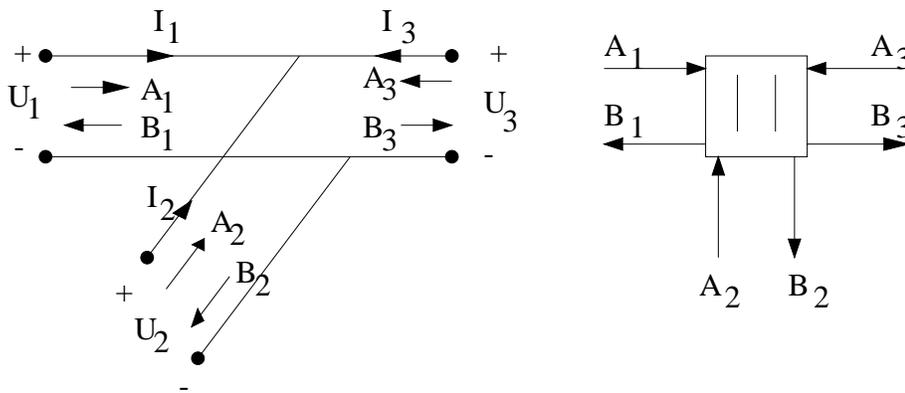


Figure 4.5: vertaling van de parallelschakeling

serie verbinding kunnen we het verbindingsnetwerk van alle ladders opbouwen, vandaar de behandeling van beide.

In figuur 4.5 is de parallelle verbinding getekend met zijn digitale symbool de 'driepoort parallel adapter'.

We hebben:

$$U_1 = U_2 = U_3 \quad \text{en} \quad I_1 + I_2 + I_3 = 0 \quad (4.8)$$

Uit de voor elke poort gedefinieerde **A** en **B** volgt:

$$\begin{aligned} B_i &= (\alpha_1 A_1 + \alpha_2 A_2 + \alpha_3 A_3) - A_i \\ \alpha_i &= \frac{2g_i}{g_1 + g_2 + g_3} \\ \alpha_1 + \alpha_2 + \alpha_3 &= 2 \\ g_i &= \frac{1}{R_i}, \quad i = 1, 2, 3, \quad 0 < \alpha_i < 2 \end{aligned} \quad (4.9)$$

In (4.9) zijn de coëfficiënten α_i niet onafhankelijk, zodat we één kunnen elim-

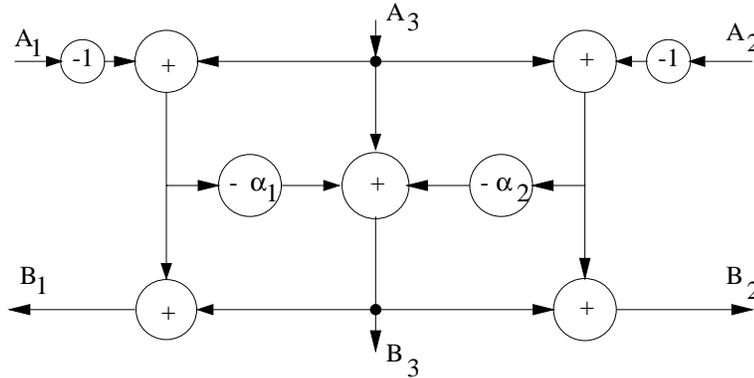


Figure 4.6: rekenschema voor de driepoort-paralleladapter

ineren. De poort waarvan we de coefficient elimineren noemen we de afhankelijke poort.

Kiezen we poort 3 afhankelijk dan krijgen we:

$$\begin{aligned}
 B_3 &= A_3 - \sum_{i=1}^2 \alpha_i (A_3 - A_i) \\
 B_i &= B_3 + (A_3 - A_i), \quad i = 1, 2
 \end{aligned}
 \tag{4.10}$$

Deze matrixrelatie \hat{S} is gegeven in figuur 4.6 met een stroomschema dat 2 vermenigvuldigers bevat:

$$\hat{S} = \begin{bmatrix} \alpha_1 - 1 & \alpha_2 & 2 - \alpha_1 - \alpha_2 \\ \alpha_1 & \alpha_2 - 1 & 2 - \alpha_1 - \alpha_2 \\ \alpha_1 & \alpha_2 & 1 - \alpha_1 - \alpha_2 \end{bmatrix}
 \tag{4.11}$$

Een probleem doet zich voor bij het direkt verbinden van adapters met elkaar: er ontstaat ter plaatse een niet-berekenbare rekenlus!

Dit kunnen we oplossen door één van de 2 poorten die verbonden worden met elkaar, 'aangepast' te maken. Voor die aangepaste poort geldt:

$$\alpha_i = 1
 \tag{4.12}$$

Kiezen we poort 3 aangepast in (4.10) dan, met $\alpha_3 = 1$ wordt B_3 onafhankelijk van A_3 ! en daarmee is de rekenlus verbroken.

We hebben nu echter een conditie ($\alpha_3 = 1$) toegevoegd.

Hierdoor verandert (4.10) in (4.13):

$$\begin{aligned}
 \alpha_3 &= 1, & g_3 &= g_1 + g_2 \\
 \alpha_1 + \alpha_2 &= 1 & \alpha_i &= \frac{g_i}{g_3} \quad (i = 1, 2)
 \end{aligned}
 \tag{4.13}$$

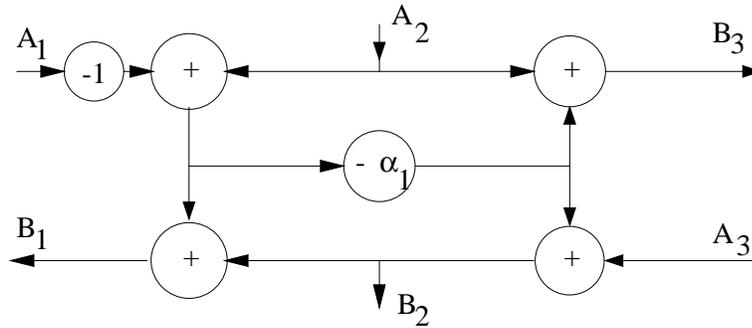


Figure 4.7: rekenschema voor de driepoort-paralleladapter met aangepaste poort 3

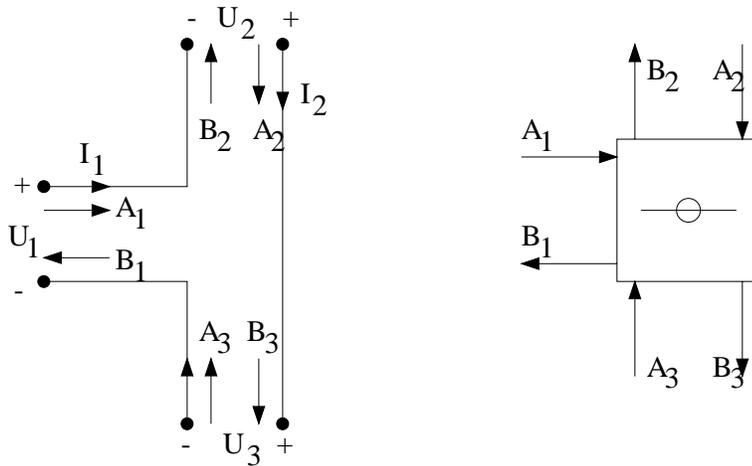


Figure 4.8: vertaling van de serieverbinding

We kunnen opnieuw een poort (behalve poort 3) afhankelijk kiezen. Doen we dit voor poort 2 dan resulteert figuur 4.7 en:

$$\hat{S} = \begin{bmatrix} \alpha_1 - 1 & -\alpha_1 + 1 & 1 \\ \alpha_1 & -\alpha_1 & 1 \\ \alpha_1 & -\alpha_1 + 1 & 0 \end{bmatrix} \quad (4.14)$$

4.2.6 De driepoort serie verbinding

We hebben de situatie van figuur 4.8.

Hieruit volgt:

$$U_1 + U_2 + U_3 = 0, \quad I_1 = I_2 = I_3 \quad (4.15)$$

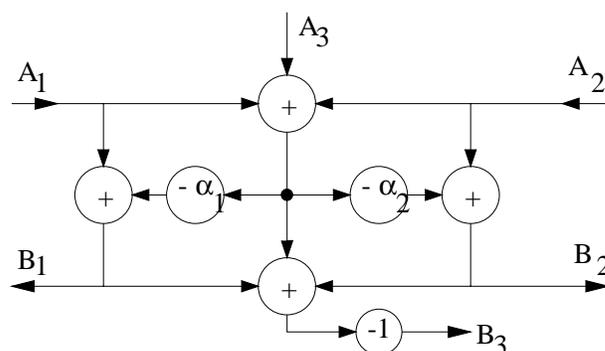


Figure 4.9: rekenschema van de driepoort-serieverbinding

Met de spanningsgolfvariabelen levert dit:

$$B_i = A_i - \alpha_i(A_1 + A_2 + A_3) \quad \alpha_i = \frac{2R_i}{R_1 + R_2 + R_3} \quad (4.16)$$

$$\alpha_1 + \alpha_2 + \alpha_3 = 2 \quad i = 1, 2, 3$$

Kiezen we opnieuw poort 3 aangepast dan wordt dit:

$$\alpha_3 = 1, \quad R_3 = R_1 + R_2 \quad (4.17)$$

$$\alpha_1 + \alpha_2 = 1, \quad \alpha_i = \frac{R_i}{R_3}, \quad i = 1, 2$$

Uit (4.16) met poort 3 afhankelijk ontstaat figuur 4.9 en:

$$\hat{S} = \begin{bmatrix} 1 - \alpha_1 & -\alpha_1 & -\alpha_1 \\ -\alpha_2 & 1 - \alpha_2 & -\alpha_2 \\ \alpha_1 + \alpha_2 - 2 & \alpha_1 + \alpha_2 - 2 & \alpha_1 + \alpha_2 - 1 \end{bmatrix} \quad (4.18)$$

Uit (4.17) met poort 2 afhankelijk volgt figuur 4.10 en:

$$\hat{S} = \begin{bmatrix} 1 - \alpha_1 & -\alpha_1 & -\alpha_1 \\ \alpha_1 - 1 & \alpha_1 & \alpha_1 - 1 \\ -1 & -1 & 0 \end{bmatrix} \quad (4.19)$$

In figuur 4.11 wordt aangegeven hoe in een adapter wordt meegegeven welke poort aangepast is.

4.3 Twee voorbeelden

Voorbeeld 1:

Gegeven is het laddernetwerk van figuur 4.12.

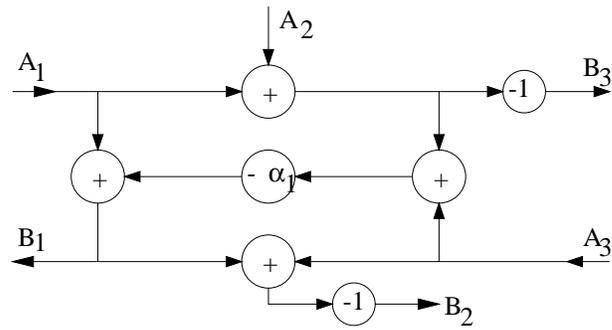


Figure 4.10: rekenschema van de driepoort-serieverbinding met aangepaste poort 3

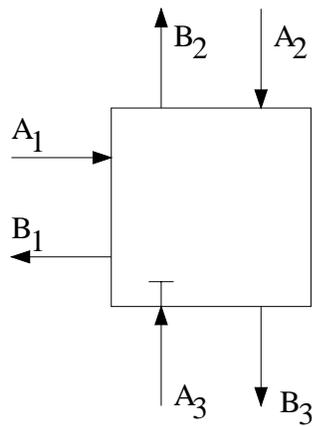


Figure 4.11: adapter met aangepaste poort 3

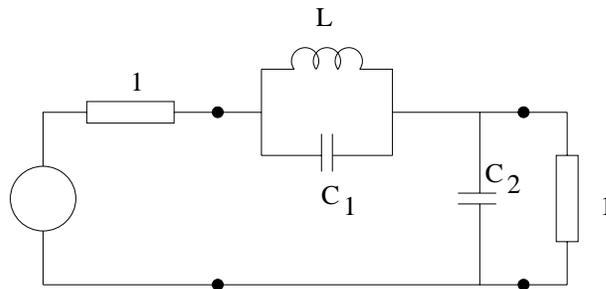


Figure 4.12: een verliesvrij laddernetwerk als voorbeeld

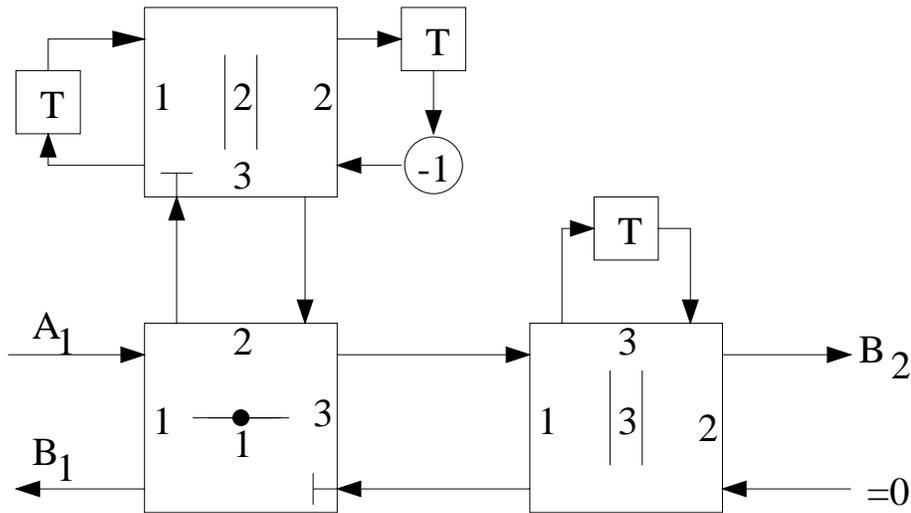


Figure 4.13: vertaling in adaptervorm van figuur 12.12

Omdat de referentieweerstanden beide 1Ω zijn, zijn voor de in- en uitgangspoort van het verliesvrije netwerk de vermogensgolfvariabelen numeriek gelijk aan de spanningsgolfvariabelen.

De vertaling in adapterstructuur is gegeven in figuur 4.13.

In figuur 4.13 zijn tweemaal adapterpoorten direct verbonden, er moeten dus 2 poorten (elk van een andere adapter) aangepast worden gekozen.

De gekozen poortnummering is in figuur 4.13 aangegeven. In adapter 1 zit figuur 4.10, in adapter 2 zit figuur 4.7 en in adapter 3 zit figuur 4.6.

Ga zelf na hoe de volgorde van de berekeningen verloopt in dit rekenschema.

Voor adapter 2 geldt:

$$\begin{aligned}
 g_1 &= C_1, \\
 g_2 &= \frac{1}{L}, \\
 g_3 &= g_1 + g_2 \quad (\text{volgens (4.13)}) \Rightarrow \alpha_1 = \frac{g_1}{g_3}
 \end{aligned}
 \tag{4.20}$$

Voor adapter 1 geldt:

$$\begin{aligned}
 R_1 &= 1 \\
 R_2 &= \frac{1}{g_3} \quad (\text{van adapter 2}) \\
 R_3 &= R_1 + R_2 \quad (\text{volgens (4.16)}) \Rightarrow \alpha_1 = \frac{R_1}{R_3}
 \end{aligned}
 \tag{4.21}$$

Voor adapter 3 geldt:

$$\begin{aligned}
 g_1 &= \frac{1}{R_3} \quad (\text{van adapter 1}) & \alpha_1 &= \frac{2g_1}{g_1+g_2+g_3} \\
 g_3 &= C_2 & \Rightarrow & \\
 g_2 &= 1 & \alpha_2 &= \frac{2g_2}{g_1+g_2+g_3}
 \end{aligned} \tag{4.22}$$

Dit voorbeeld bevat dus 4 vermenigvuldigingscoëfficiënten!
 Kleine afwijkingen in de ideale waarden (b.v. door afronding) van deze 4 coëfficiënten zijn steeds terug te rekenen naar kleine afwijkingen van de ideale waarden van L , C_1 , C_2 en de belastingsweerstand!

Voorbeeld 2:

Aan de hand van een numeriek voorbeeld laten we zien wat orthogonaal zijn betekent voor een driepoort-parallel adapter.

Vroeger hebben we gezien dat voor orthogonale netwerken geldt:

$$S_{(z)}^T \cdot S_{(z^{-1})} = 1$$

Dit werd afgeleid uitgaande van de vermogensparameters a_i en b_i .
 De golffilters zijn echter gedefinieerd op basis van spanningsparameters a_i en b_i

Dit leidt tot onderstaande relatie tussen spannings- en vermogens S -matrices:
 (\hat{S}, S)

$$S = G^{\frac{1}{2}} \hat{S} G^{-\frac{1}{2}}$$

Hierin is G een diagonaalmatrix met poortgeleidingen g_i .
 De orthogonaliteitsrelatie wordt nu:

$$\begin{aligned}
 (G^{\frac{1}{2}} \hat{S}_{(z)} G^{-\frac{1}{2}})^T (G^{\frac{1}{2}} \hat{S}_{(z^{-1})} G^{-\frac{1}{2}}) &= 1 \\
 G^{-\frac{1}{2}} \hat{S}_{(z)}^T G^{\frac{1}{2}} G^{\frac{1}{2}} \hat{S}_{(z^{-1})} G^{-\frac{1}{2}} &= 1 \\
 \hat{S}_{(z)}^T G \hat{S}_{(z^{-1})} &= G
 \end{aligned}$$

Toegepast op een numeriek voorbeeld van een driepoort parallel adapter waarvoor geldt:

$$\left. \begin{array}{l} g_1 = 1 \\ g_2 = 2 \\ g_3 = 1 \end{array} \right\} \Rightarrow \left. \begin{array}{l} \alpha_1 = \frac{2g_1}{g_1+g_2+g_3} = 0.5 \\ \alpha_2 = \frac{2g_2}{g_1+g_2+g_3} = 1 \end{array} \right\} \Rightarrow \hat{S} = \begin{bmatrix} -0.5 & 1 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0.5 & 1 & -0.5 \end{bmatrix}$$

De orthogonaliteitsrelatie wordt nu:

$$\begin{bmatrix} -0.5 & 0.5 & 0.5 \\ 1 & 0 & 1 \\ 0.5 & 0.5 & -0.5 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -0.5 & 1 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0.5 & 1 & -0.5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Eenvoudig rekenwerk leert dat dit klopt.

4.4 Opmerkingen

1. Omdat kleine afwijkingen in de coëfficiënten altijd terug te rekenen zijn naar elementwaarden in het oorspronkelijke netwerk (met positieve waarden), is de gerealiseerde functie altijd stabiel.
(Vergelijk dit met de traditionele overdrachtsfunctie realisaties).
2. Uit de adapterstructuur blijkt dat we twee functies hebben gerealiseerd: S_{21} en S_{11} . (Overdracht + reflectie!).
3. Deze golffiltermethode levert ons dus een manier om selectieve digitale filters te vinden met eigenschappen die direct samenhangen met het analoge, verliesvrije prototype wat we daarvoor eerst moeten ontwerpen. Dit laatste is een nadeel. Vandaar dat veel aandacht besteed is (en wordt) aan directe synthese van functies van z met soortgelijke eigenschappen (orthogonale filters).

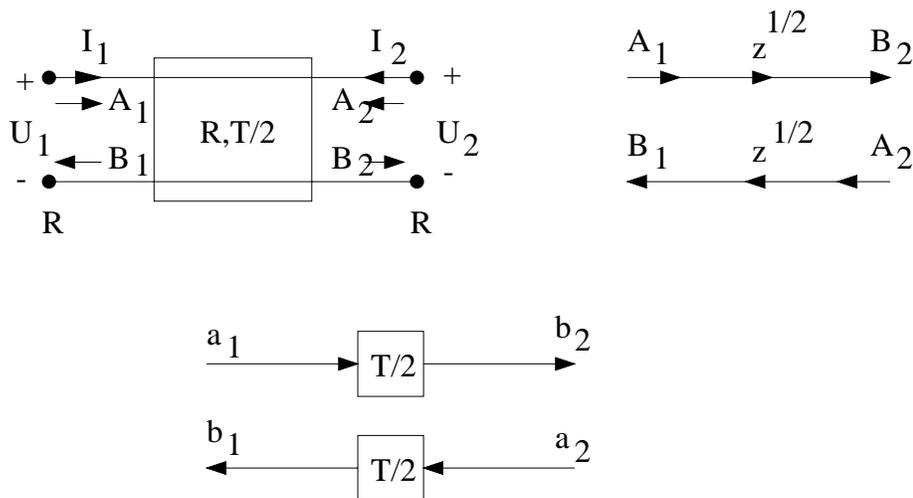


Figure 4.14: het eenheidselement en zijn vertaling

4.5 Een uitbreiding

4.5.1 Inleiding

De golffiltermethodiek die in het voorgaande is gegeven is een vereenvoudigde versie.

Een meer uitgebreide versie maakt ook gebruik van de microgolffiltertheorie. De componenten die in deze microgolffilters worden gebruikt zijn stukken ideale transmissielijn van allemaal dezelfde lengte en verschillende karakteristieke weerstanden zoals reeds beschreven in deel 2.

4.5.2 De digitale vertaling

De digitale vertaling van het eenheidselement volgt direkt uit formule (4.7). De impedantie matrix vergelijkingen worden:

$$\left. \begin{aligned} U_1 &= \frac{R(z+1)}{z-1} I_1 + \frac{2R\sqrt{z}}{z-1} I_2 \\ U_2 &= \frac{2R\sqrt{z}}{z-1} I_1 + \frac{R(z+1)}{z-1} I_2 \end{aligned} \right\} \quad (4.23)$$

Getransformeerd m.b.v. de spanningsgolfvariabelen wordt dit:

$$\left. \begin{aligned} B_1 &= z^{\frac{1}{2}} A_2 \\ B_2 &= z^{\frac{1}{2}} A_1 \end{aligned} \right\} \quad (4.24)$$

Een en ander is te zien in figuur 4.14.

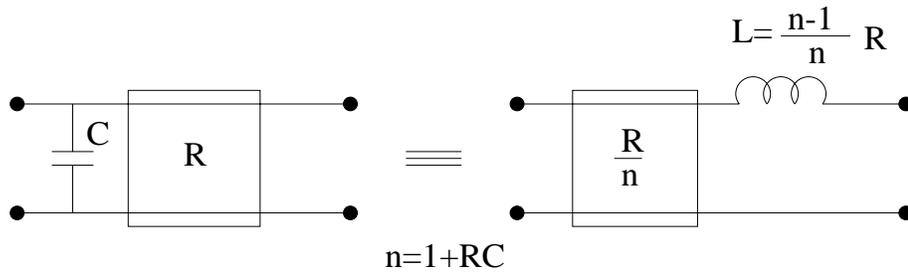


Figure 4.15: tweepoort-identiteit

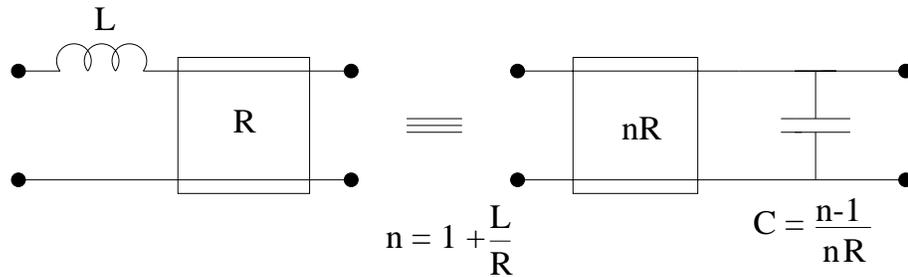


Figure 4.16: dezelfde identiteit als die uit figuur 4.15

4.5.3 Enkele w vlak (Kuroda) identiteiten

In de figuren 4.15 en 4.16 volgt een w -vlak tweepoort identiteit die verderop nuttig blijkt bij het vinden van equivalente netwerken.

of andersom:

Gebruikmakend van deze identiteiten en kunnen we de volgende mogelijke vertalingen van w -vlak resonantiekeringen afleiden:

De seriekring: zie figuur 4.17

De parallelkring: zie figuur 4.18

Uit de figuren 4.18 en 4.19 blijkt dat nu ook de situatie kan ontstaan dat we 2 poorten direkt moeten verbinden, m.a.w. we hebben de rekenstructuur voor een tweepoort adapter nodig. Deze kan gezien worden als vereenvoudiging naar twee poorten van de serie- of parallel adapter.

Dit resulteert in onderstaande matrixrelatie:

$$\hat{S} = \begin{bmatrix} -(1 - \alpha_1) & 2 - \alpha_1 \\ \alpha_1 & 1 - \alpha_1 \end{bmatrix} = \begin{bmatrix} -\alpha & 1 + \alpha \\ 1 - \alpha & \alpha \end{bmatrix} \quad (4.25)$$

waarin: $\alpha_1 = \frac{2g_1}{g_1 + g_2}$, en: $\alpha = 1 - \alpha_1 = \frac{R_1 - R_2}{R_1 + R_2}$

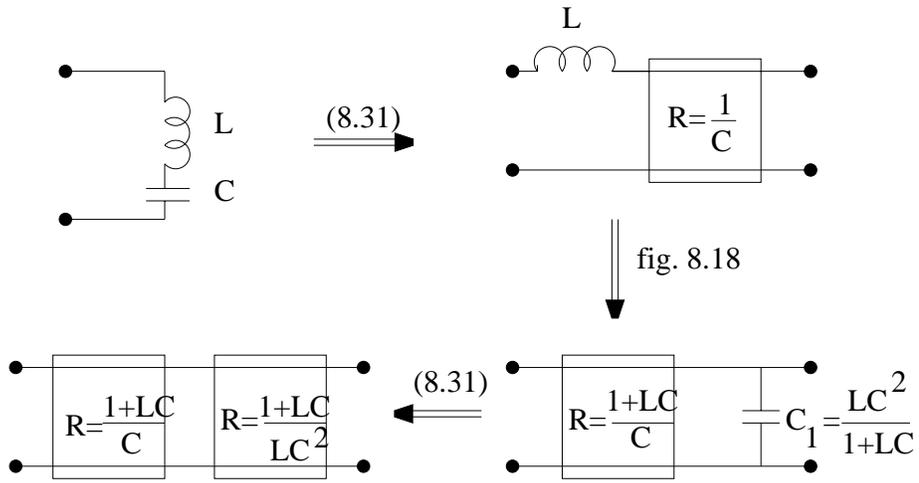


Figure 4.17: vertaling van de seriekring

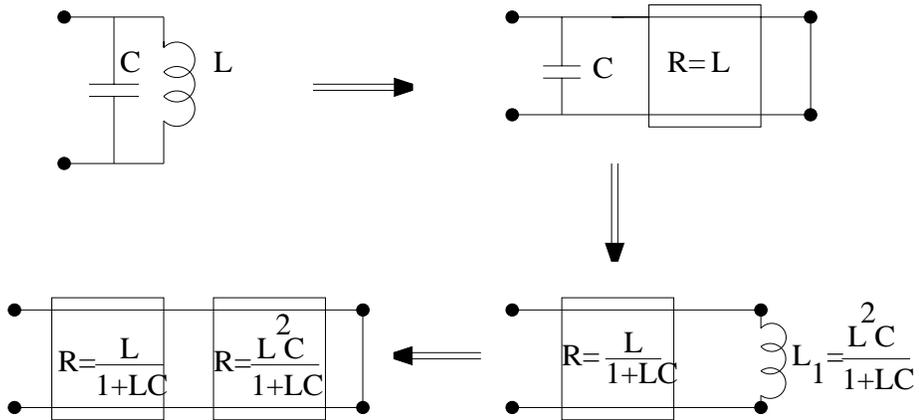


Figure 4.18: vertaling van de parallelkring

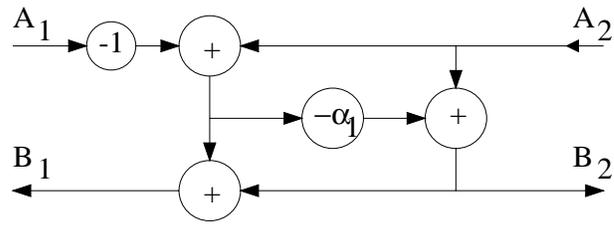


Figure 4.19: eerste rekenschema behorend bij de tweepoortadapter

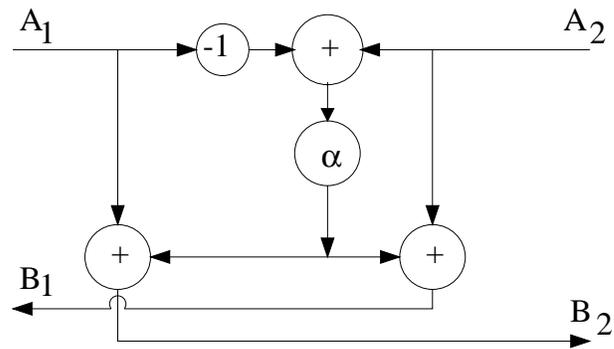


Figure 4.20: tweede rekenschema behorend bij de tweepoortadapter

De twee matrices geven aanleiding tot de figuren 4.19 en 4.20. Het adapter symbool is gegeven in figuur 4.21.

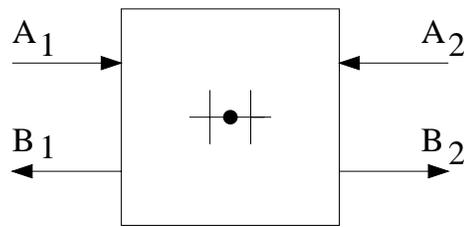


Figure 4.21: symbool voor de tweepoortadapter

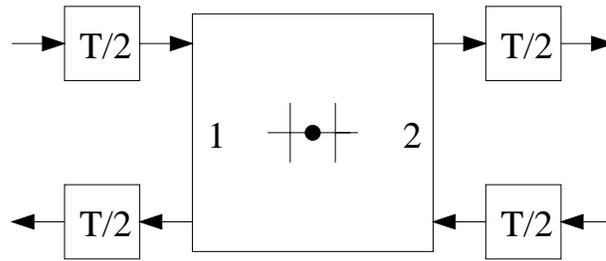


Figure 4.22: vertaling van een seriekring

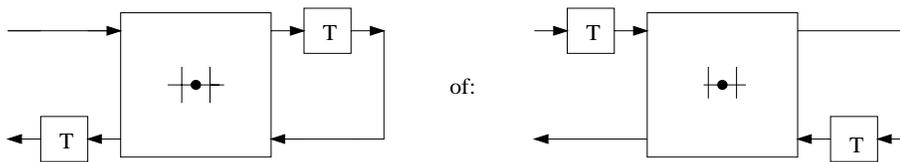


Figure 4.23: twee te gebruiken vertalingen van de seriekring

4.5.4 Enkele voorbeelden

Voorbeeld 1:

De vertaling van de seriekring uit figuur 4.17 m.b.v. de tweepoortadapter wordt figuur 4.22. (de vertaling van een open poort is hierin verwerkt).

Voor de adapter geldt:

$$g_1(= \frac{1}{R_1}) = \frac{C}{1 + LC}$$

$$g_2(= \frac{1}{R_2}) = \frac{LC^2}{1 + LC}$$

Door de adapterberekeningen $T/2$ eerder of later te doen, ontstaan de figuren van 4.23.

Voorbeeld 2:

De vertaling van de parallelkring uit figuur 4.18 wordt figuur 4.24 (hierin is de vertaling van een kortgesloten poort opgenomen).

Dit is op dezelfde manier als de vertaling van de seriekring te herleiden tot een rekenstructuur met slechts delay-elementen T .

Voor de adapter geldt:

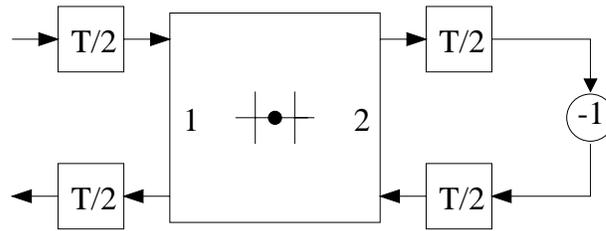


Figure 4.24: vertaling van de parallelkring

$$g_1(= \frac{1}{R_1}) = \frac{1 + LC}{L}$$

$$g_2(= \frac{1}{R_2}) = \frac{1 + LC}{L^2 C}$$

Voorbeeld 3:

De vertaling van een verliesvrije impedantie bestaande uit, afwisselend, L en C elementen, wordt dus op deze manier een kaskadering van eenheidselementen (met de laatste poort open- of kortgesloten).

Zelf doen: || Nu we de tweepoortadapter beschikbaar hebben en alternatieven voor de parallelkring, bestaan er meerdere mogelijke vertalingen van ladders.

4.6 Een andere realisatiestructuur

We hebben gezien dat verliesvrije laddernetwerken, afgesloten in weerstanden, belangrijk zijn als filterrealisatievorm en ook hebben we aan de hand van enkelen voorbeelden laten zien hoe we de netwerken vinden.

Behalve het laddernetwerk is er nog een andere structuur van belang, zoals we hebben gezien: het Jauman netwerk.

Dit is een kanonieke realisatievorm voor verliesvrije, reciproke, symmetrische tweepoort netwerken.

Het Jauman netwerk is opnieuw getekend in figuur 4.25. Hierin zijn $2Z_2$ en $\frac{1}{2}Z_1$ verliesvrije impedanties.

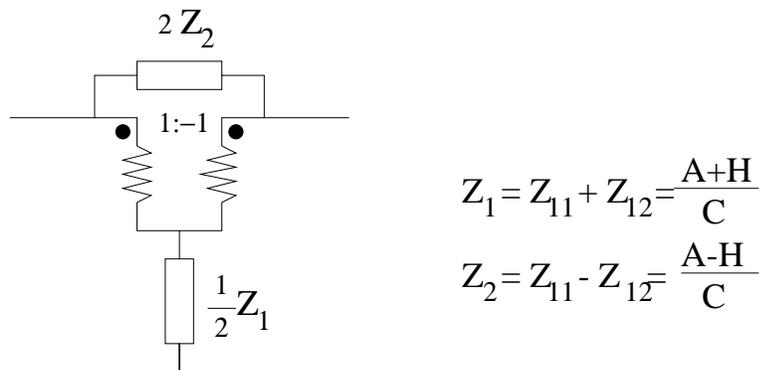


Figure 4.25: Jauman netwerk

De tijdsdiskrete versie van de Jauman structuur.

In figuur 4.25 hebben we 2 impedanties ($2Z_2$ en $\frac{1}{2}Z_1$) aangesloten op de 4-poort van onderstaande figuur 4.26.

Als we voor dit netwerk de spanningsverstrooiingsparameters definiëren op de gebruikelijke manier, dan kan worden afgeleid dat geldt:

$$\hat{S} = \begin{bmatrix} 0 & 0 & \frac{1}{2} & 1 \\ 0 & 0 & -\frac{1}{2} & 1 \\ 1 & -1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \end{bmatrix} \quad \text{met} \quad \begin{aligned} g_1 &= g_2 \\ 4g_3 &= g_4 \\ g_1 &= 2g_3 \end{aligned}$$

Als we de referentieweerstanden $R_1 = R_2 = 1\Omega$ stellen, dan wordt:

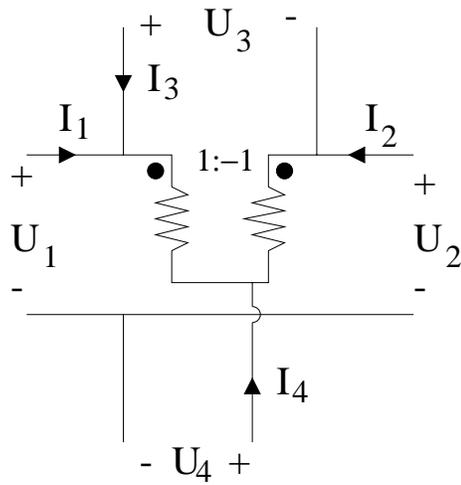


Figure 4.26: Jauman 4-poort

$$g_3 = \frac{1}{2} \quad \text{en} \quad g_4 = 2.$$

Het bijbehorende adapternetwerk wordt figuur 4.27.

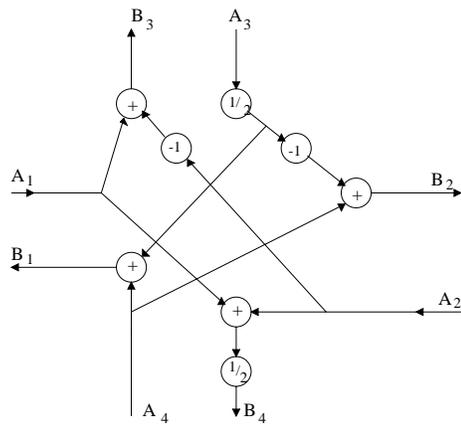


Figure 4.27:

Voor het praktische geval dat $A_2 = 0$ wordt dit figuur 4.28.
 Als we stellen: $A_3 = H_1(z) \cdot B_3$ en $A_4 = H_2(z) \cdot B_4$,

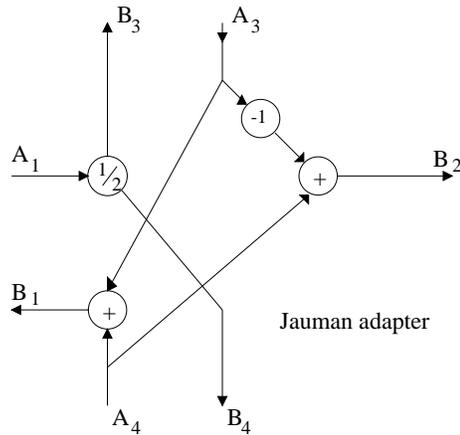


Figure 4.28: Jauman adapter

dan volgt:

$$\frac{B_2}{A_1} = \frac{1}{2}[H_2(z) - H_1(z)]$$

$$\frac{B_1}{A_1} = \frac{1}{2}[H_1(z) + H_2(z)]$$

Deze relaties zijn opgenomen in figuur 4.29.

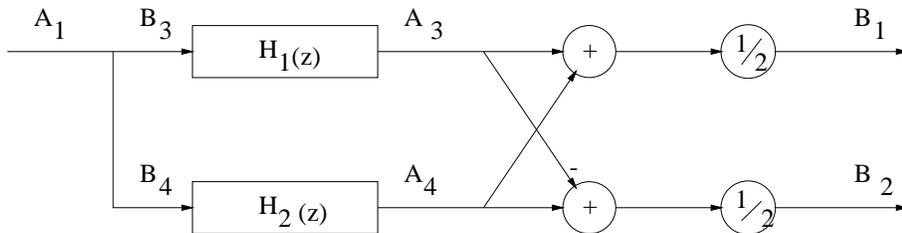


Figure 4.29:

Hierin zijn $H_1(z)$ en $H_2(z)$ reflektiefuncties van de verliesvrije impedanties $\frac{1}{2}Z_1$ en $2Z_2$ en dus all-pass functies.

Bovendien: $\frac{B_2}{A_1}$ en $\frac{B_1}{A_1}$ hebben hier

de eigenschap dat geldt:

$$\left| \frac{B_2}{A_1} \right|^2 + \left| \frac{B_1}{A_1} \right|^2 = 1 \quad (\text{verliesvrijheid, } S\text{-matrix eigenschap})$$

Dit blijft identiek gelden, ook bij kleine variaties in de vermenigvuldigingscoëfficiënten waarmee de all-pass functies worden gerealiseerd (de 2 referentie-

weerstanden van poort 1 en 2: beide 1Ω , veranderen niet. Dit in tegenstelling tot de situatie bij laddernetwerken).

De overdrachten $H_1(z)$ en $H_2(z)$ kunnen op verschillende manieren worden gerealiseerd:

1. Na berekening van $\frac{1}{2}Z_1$ en $2Z_2$ volgen hieruit rechtstreeks $H_1(z)$ en $H_2(z)$, deze kunnen vervolgens d.m.v. de standaard digitale structuren (direkte, parallele, cascade....) worden gerealiseerd.
2. De impedanties $\frac{1}{2}Z_1$ en $2Z_2$ kunnen worden gerealiseerd d.m.v. Foster- of Cauer netwerken. Deze netwerken kunnen dan d.m.v. de golfILTERVERTALING en 3-poort adapters worden gerealiseerd.
3. De impedanties $\frac{1}{2}Z_1$ en $2Z_2$ hebben een realisatie d.m.v. een cascade van eenheidselementen \rightarrow vertaling \rightarrow 2-poort adapters.

Berekening van $H_1(z)$ en $H_2(z)$:

$$H_1(s) = \rho(s) = \frac{2Z_2 - 2}{2Z_2 + 2} = \frac{Z_2 - 1}{Z_2 + 1} \left(= \frac{A - H - C}{A - H + C} \right)$$

$$H_2(s) = \rho(s) = \frac{\frac{1}{2}Z_1 - \frac{1}{2}}{\frac{1}{2}Z_1 + \frac{1}{2}} = \frac{Z_1 - 1}{Z_1 + 1} \left(= \frac{A + H - C}{A + H + C} \right)$$

Hierna volgen $H_1(z)$ en $H_2(z)$ weer via de bilineaire z-transformatie.

Tenslotte:

De Jauman structuur heeft als voordeel boven de ladderstructuur:

1. Symmetrisch,
2. Kanoniek in het aantal elementen.

Nadeel:

Alleen voor oneven orde filternetwerken.

Ook:

De tijdsdiskrete versie van de Jauman structuur levert de mogelijkheid van het realiseren van een laagdoorlaat én hoogdoorlaat in dezelfde frekwentieband waarvoor identiek geldt:

$$\left| \frac{B_1}{A_1} \right|^2 + \left| \frac{B_2}{A_1} \right|^2 = 1 \quad (\text{dus onafh. van coeff.variaties})$$

4.7 Een filtervoorbeeld

Gekozen is voor een derde orde Chebyshev laagdoorlaat.

Uit het boek: E.Christian, E.Eisenmann, "Filter design tables and graphs", John Wiley, 1966, pagina 43 (fac. bibl.:Gd 96) komt de volgende informatie:

$$Y(s) = \frac{1}{C_1(s + a_0)(s^2 + 2a_1s + a_1^2 + b_1^2)} \quad \text{met:}$$

C_1	=	2.3094
a_0	=	0.45322
a_1	=	0.22661
b_1	=	0.95082
Ω	=	0.866025

De karakteristiek wordt dan figuur 4.30:

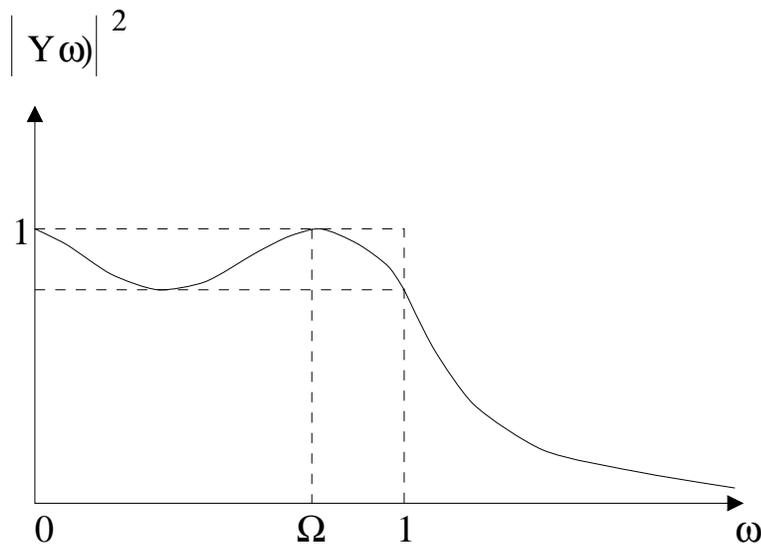


Figure 4.30:

De functie $Y(s)$ voldoet aan alle eisen om te dienen als $S_{21}(s)$ van een verliesvrij netwerk, m.a.w. de polynomen f en g zijn bekend. Ook het polynoom h is (op een teken na) bekend:

$$h = \pm C_1 s(s^2 + \Omega^2).$$

We kiezen voor het + teken.

Ook de Tellegen polynomen A , B , C , D , H en K kunnen nu berekend worden: ($A=B$, $K=0$).

$$2H = 1.$$

Uit het even deel van het polynoom G volgt:

$$2A = 2.0933s^2 + 1$$

De polynomen C en D halen we uit het oneven deel van g en het polynoom h:

$$\begin{aligned} C + D &= 2.3094s^3 + 2.681s & \Rightarrow & 2C = 0.9489s \\ -C + D &= 2.3094s^3 + 1.7321s & & 2D = 4.6188s^3 + 4.4131s \end{aligned}$$

Deingangsimpedantie Z van het verliesvrije netwerk afgesloten in 1Ω vinden we uit de reflectie:

$$Z = \frac{g + h}{g - h} \quad (\rho = \frac{h}{g})$$

$$\begin{aligned} Z &= \frac{2.3094s^3 + 2.0933s^2 + 2.681s + 1 + 2.3094s^3 + 1.732s}{2.3094s^3 + 2.0933s^2 + 2.681s + 1 - 2.3094s^3 - 1.732s} \\ Z &= \frac{4.6188s^3 + 2.0933s^2 + 4.413s + 1}{2.0933s^2 + 0.949s + 1} \\ Z &= 2.2065s + \frac{1}{0.9487s + \frac{1}{2.2065s + 1}} \end{aligned}$$

Hieruit volgt het filternetwerk van figuur 4.31:

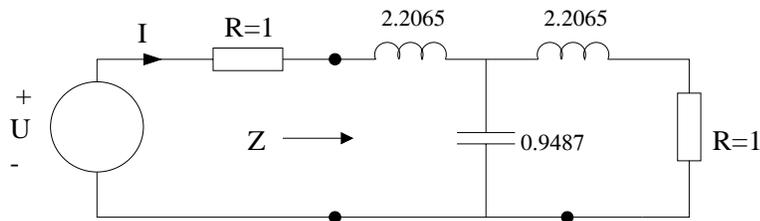


Figure 4.31: (Symmetrisch netwerk!)

Bereken zelf het netwerk dat ontstaat als we voor het polynoom h het - teken kiezen!

Concluderend tot zover:

1. We hebben een $Y(s)$ met een Chebyshev laagdoorlaat karakteristiek. M.b.v. de bilineaire z -transformatie kunnen we hieruit $Y(z)$ vinden en die realiseren.

2. We hebben een $Y_2(s)$ met een hoogdoorlaat karakteristiek

$$Y_2(s) = \frac{h}{g}$$

in dezelfde frekwentieband.

3. Er geldt:

$$|Y(s)|_{s=j\omega}^2 + |Y_2(s)|_{s=j\omega}^2 = 1. \quad (|Y(z)|_{|z|=1}^2 + |Y_2(z)|_{|z|=1}^2 = 1)$$

4. We hebben 2 laddernetwerken waarin $Y(s)$ en $Y_2(s)$ als overdracht en reflectie functie worden gerealiseerd.

Een digitale golffiltervertaling van deze netwerken levert alternatieve realisaties op van $Y(z)$ en $Y_2(z)$, met golffilter eigenschappen.

De figuren 4.32 en 4.33 geven twee adapterrealisaties hiervan.

Figuur 4.33 maakt slim gebruik van de symmetrie van figuur 4.31 om daarmee het aantal verschillende coefficienten te beperken.

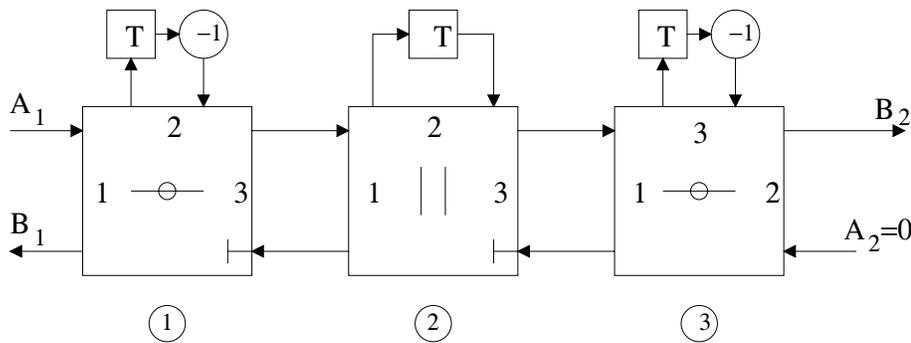


Figure 4.32:

Figuur 4.32 bevat in totaal 4 vermenigvuldigers met 4 verschillende coefficienten.

De adapters 1 en 3 van figuur 4.33 bevatten dezelfde schakeling (vanwege de symmetrie zo gekozen) en dezelfde coefficient.

Adapter 2 heeft 2 poorten met dezelfde poortgeleiding en dus is een schakeling mogelijk met één coefficient.

Figuur 4.33 bevat dus 3 vermenigvuldigers waarvan 2 coefficienten verschillend.

Figuur 4.32:

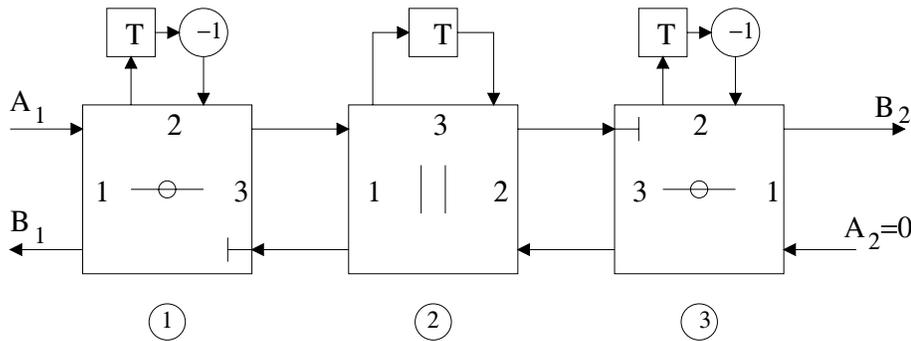


Figure 4.33:

adapter 1 : figuur 4.10 : $R_1=1, R_2=2.2065, R_3=3.2065, \alpha_1=0.3119$
 2 : 4.7 : $g_1=0.3119, g_2=1.0541, g_3=1.3660, \alpha_1 = 0.2283$
 3 : 4.9 : $R_1=0.7321, R_2=1, R_3=2.2065, \alpha_1 = 0.3717, \alpha_2=0.5078$

Figuur 4.33:

adapter 1 : figuur 4.10 : $R_1=1, R_2=2.2065, R_3=3.2065, \alpha_1=0.3119$
 2 : 4.6 : $g_1=0.3119, g_2=0.3119, g_3=1.0541, \alpha_1 = \alpha_2=0.3717$
 3 : 4.10 : $R_1=1, R_2=2.2065, R_3=3.2065, \alpha_1=0.3119$

Een volgende categorie mogelijke digitale realisaties ontstaat als we $\frac{B_2}{A_1}$ en $\frac{B_1}{A_1}$ schrijven als het verschil (of de som) van 2 all-pass functies. Voor de berekening van deze $H_1(z)$ en $H_2(z)$ hebben we de impedanties Z_1 en Z_2 uit de Jauman structuur nodig:

$$Z_1 = \frac{A+H}{C} = \frac{2.0933s^2 + 2}{0.9489s} = 2.206s + \frac{1}{0.4745s}$$

$$Z_2 = \frac{A-H}{C} = \frac{2.0933s^2}{0.9489s} = 2.206s$$

Hiermee hebben we de Jauman variant van figuur 4.31 te pakken, zie figuur 4.34.

Er volgt nog:

$$H_2(s) = \frac{Z_1 - 1}{Z_1 + 1} = \frac{2.0933s^2 - 0.9489s + 2}{2.0933s^2 + 0.9489s + 2} \quad \text{en:} \quad H_1(s) = \frac{2.206s - 1}{2.206s + 1}$$

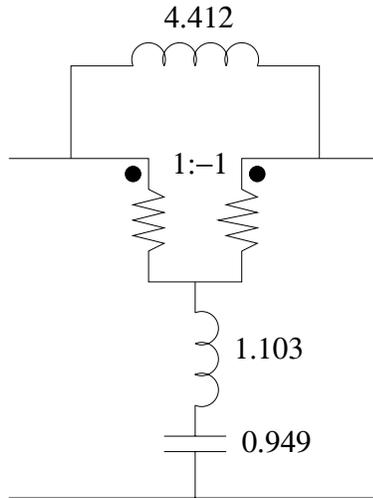


Figure 4.34:

Hieruit volgen via de bilineaire transformatie $H_1(z)$ en $H_2(z)$, etc.

Met behulp van klassieke realisaties van deze twee functies kunnen we ook figuur 4.29 vullen.

Nog een andere realisatie variant voor $\frac{B_2}{A_1}$ en $\frac{B_1}{A_1}$ ontstaat als we figuur 4.34 vertalen naar figuur 4.35.

Figuur 4.35:

Adapter 1: figuur 4.19 of 4.20.

$g_1=0.5$, $g_2=0.2267$, $\alpha_1=1.3761$

Jauman adapter: $g_3=0.5$, $g_4=2$.

Adapter 2: figuur 4.9.

$R_1=1.0537$, $R_2=1.103$, $R_3=0.5$, $\alpha_1=0.7932$, $\alpha_2=0.8304$.

Tenslotte kunnen we de seriekring uit figuur 4.34 ook nog vertalen d.m.v. figuur 4.17. Uiteindelijk ontstaat dan figuur 4.36.

Figuur 4.36:

Adapter 1: figuur 4.19 of 4.20.

$g_1=0.5$, $g_2=0.2267$, $\alpha_1=1.3761$

Jauman adapter: $g_3=0.5$, $g_4=2$.

Adapter 2: figuur 4.19 of 4.20.

$g_1=2$, $g_2=0.4636$, $\alpha_1=1.6236$.

Adapter 3: figuur 4.19 of 4.20.

$g_1=0.4637$, $g_2=0.4853$, $\alpha_1=0.9771$.

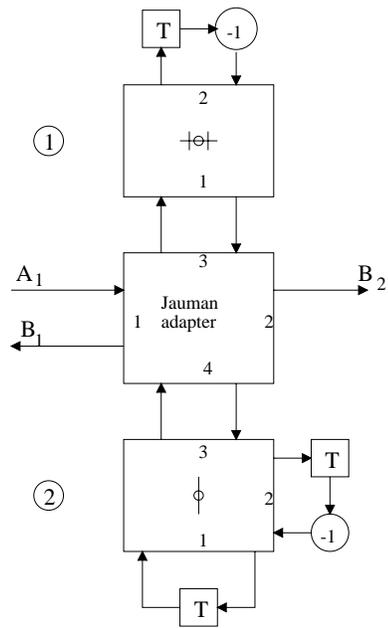


Figure 4.35:

Figuur 4.35 en 4.36 hebben beide 3 coefficienten.

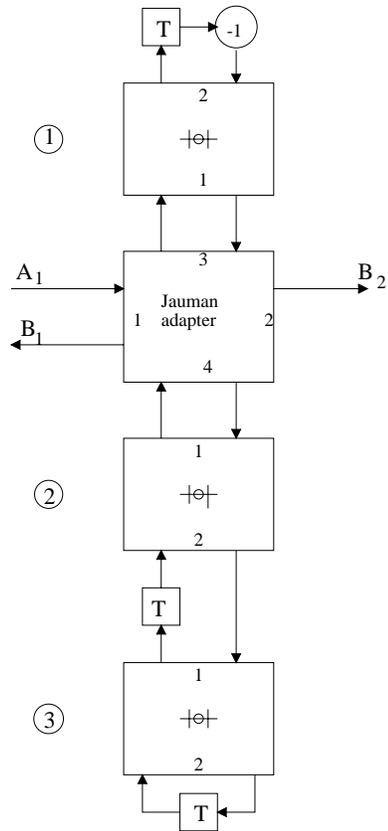


Figure 4.36:

Chapter 5

Het implementeren van tijds-diskrete structuren in (programmeerbare) digitale hardware.

5.1 Introductie.

Tijds-diskrete structuren zoals die in figuur 5.1 behoren bij een overdrachtsfunctie $H(z)$ en werken op het vaste ritme T . Het schema veronderstelt dat het rekenwerk tijdloos snel gebeurt en dat de variabelen in de vertragingselementen T steeds een tijd T vertraagd worden.

In de praktijk van digitale realisatie van deze structuren zal rekenwerk echter niet tijdloos zijn maar een bepaalde tijd t kosten. Dit betekent in feite dat de uitgangsvariabele van het filter een tijd t te laat bekend is en dat de vertragingen niet T maar kleiner moeten zijn. De maximale tijd die voor het rekenwerk beschikbaar is, is T . Kunnen we het rekenwerk in een kortere tijd realiseren dan kan in feite T kleiner gekozen worden en daarmee wordt de maximale bemonsteringsfrequentie van het filter hoger. Als we de tijd die nodig is voor het rekenwerk weten te verkorten dan wordt daarmee het filter geschikt voor gebruik bij hogere frequenties.

Tijds-diskrete structuren bevatten vaak parallelisme. Hiermee bedoelen we de mogelijkheid delen van een structuur onafhankelijk van elkaar en gelijktijdig met verschillende stukken hardware uit te voeren. Het is dan mogelijk daarmee de rekentijd voor de structuur te verkorten.

Met "pipelining" is de rekentijd verder te verkorten. Onder pipelining verstaan we hier de mogelijkheid dat een deel van de hardware die gebruikt wordt om de structuur mee te realiseren, nog bezig is met cyclus n , terwijl een ander deel reeds het rekenwerk voor cyclus $n+1$ kan beginnen. De bewerkingen

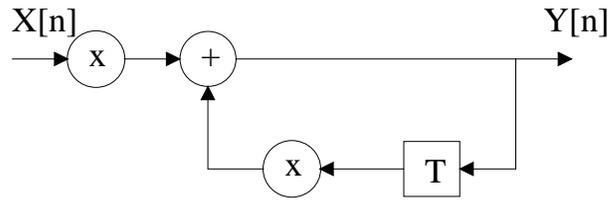


Figure 5.1: eerste orde rekursieve sectie

voor de cycli overlappen elkaar dus in de hardware, waardoor tijd bespaard kan worden. De gewonnen tijd kan gebruikt worden om de bemonsteringsfrequentie op te voeren, de reken nauwkeurigheid te verhogen of het filter uit te breiden.

Het verdelen van bewerkingen over een aantal identieke stukken hardware of rekenorganen, zodanig dat de rekestijd minimaal is, is een "scheduling" probleem.

Inzicht in parallelisme van een structuur kan worden verkregen met het tekenen van het bijbehorende "precedence (of: volgorde) diagram". Dit wordt hierna besproken. Ook zullen we in het volgende enkele voorbeelden bespreken van programmeerbare hardware (DSP's, of: digitale signaalprocessors). Een signaalprocessor is veel geschikter voor filtertoepassingen dan een "general-purpose processor". Voor deze hardware zijn verschillende principiële architecturen mogelijk waarvan slechts de elementaire mogelijkheden worden besproken zoals de instructie sets die voldoende zijn voor het programmeren van tijds-diskrete structuren.

5.2 Parallelisme in tijds-diskrete structuren.

Het schema van figuur 5.1 is op een andere manier opnieuw getekend in figuur 5.2.

Bovenaan in figuur 5.2 zijn de ingang en het delay element T (voor zover het als ingang van het rekenwerk voor cyclus n fungeert) getekend. Onderaan zijn de uitgang en het delay element T (voor zover het de ingang betreft, behorende bij de volgende cyclus $n+1$) getekend.

De bedoeling is dat het rekenwerk wordt doorlopen van boven naar beneden, het wordt zo beter mogelijk de volgorde van berekeningen te volgen en een indruk te krijgen van eventueel parallelisme aanwezig in het schema. Het schema, op deze manier getekend, heet het "precedence" (volgorde) diagram. Naast elke bewerking (en ook de uitgangen) is een getal aangegeven: het z.g. level.

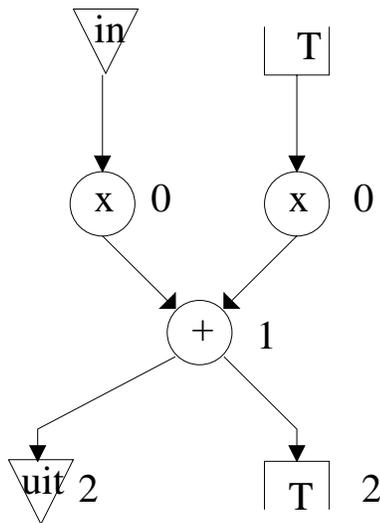


Figure 5.2: volgorde diagram behorend bij figuur 5.1

Een bewerking krijgt een level "0" als er geen bewerking vooraf gaat. Een level wordt verder als volgt bepaald:

$$\text{Level bewerking} = \max(\text{level voorafgaande bewerkingen}) + 1.$$

Het hoogste level in figuur 5.2 is dus 2.

Bewerkingen met hetzelfde level zijn in principe tegelijkertijd (parallel) uit te voeren.

Figuur 5.3 is een schema behorend bij een tweede orde rekursieve sekte. In figuur 5.4 is het bijbehorende precedence diagram getekend met daarin ook de levels. Het maximale level (in figuur 5.4: 4) geeft ook het aantal bewerkingen aan dat aanwezig is in het "kritieke pad" door het rekenschema.

Het totaal aantal bewerkingen in figuur 5.4 is 8, de lengte van het kritieke pad (maximale level) is 4. Het quotient van deze twee getallen ($8/4$) heet de Concurrency Factor (C_F) en geeft een indruk van het parallelisme dat aanwezig is in het rekenschema. Een grotere waarde voor C_F betekent meer parallelisme.

In figuur 5.4 is dus: $C_F = 2$.

In figuur 5.2: $C_F = 1.5$.

Figuur 5.5 geeft nog een andere rekursieve structuur met het bijbehorende precedence diagram in figuur 5.6 ($C_F = 8/3$).

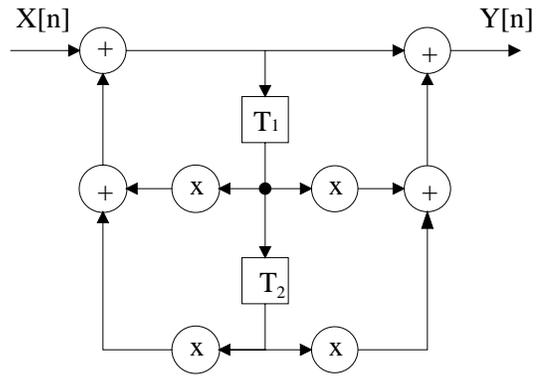


Figure 5.3: Tweede orde rekursieve sekte

Hetzelfde geldt voor de figuren 5.7 en 5.8. ($C_F = 8/3$)

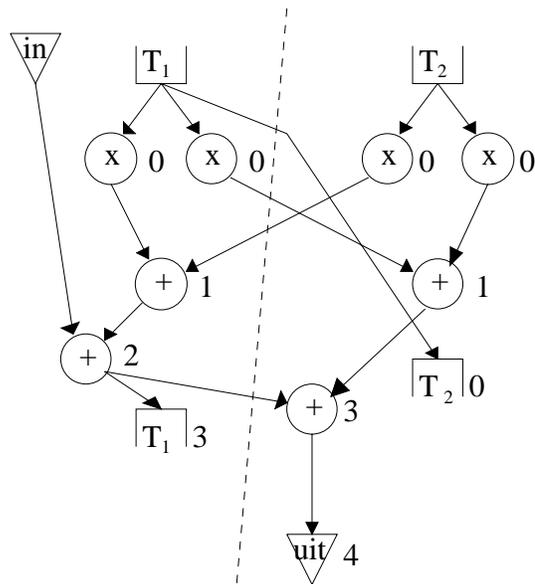


Figure 5.4: Precedence diagram behorend bij figuur 5.3

Enkele conclusies, opmerkingen en voorbeelden.

1. Onder de aannames:

- Alleen de bewerkingen + en x kosten tijd. (eventueel andere bewerkingen worden verondersteld hierin begrepen te zijn of worden verwaarloosd)
- Er is voldoende hardware aanwezig om elke bewerking met hetzelfde level, parallel te doen.
- De bewerkingstijd van + en x is gelijk en bedraagt t ns.

dan is de maximale bemonsteringsfrequentie:

$$f_{max} = \frac{1}{\text{aantal levels} \cdot t}$$

We zullen dit toepassen op de figuren 5.2, 5.4, 5.6 en 5.8 met $t=50\text{ns}$.

figuur 5.2:

Het kritieke pad kost 2 bewerkingen: $T_{min} = 100\text{ns}$. zodat: $f_{max} = 10\text{Mhz}$.

level 0: 2 vermenigvuldigen parallel

level 1: 1 optelling.

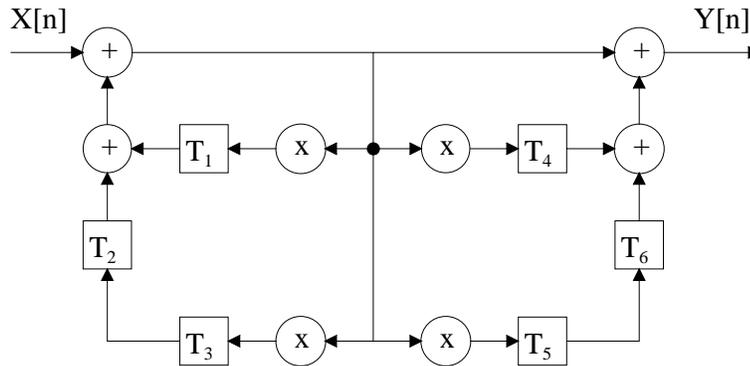


Figure 5.5: Rekursieve structuur

figuur 5.4:

Het kritieke pad kost 4 bewerkingen: $T_{min} = 200ns$. zodat: $f_{max} = 5Mhz$.

level 0: 4 vermenigvuldigingen parallel

level 1: 2 optellingen parallel

level 2: 1 optelling

level 3: 1 optelling.

figuur 5.6: Het kritieke pad kost 3 bewerkingen: $T_{min} = 150ns$. zodat: $f_{max} = 6.67Mhz$.

level 0: 2 optellingen parallel

level 1: 1 optelling

level 2: 1 optelling en 4 vermenigvuldigingen parallel.

figuur 5.8: Het kritieke pad kost 3 bewerkingen: $T_{min} = 150ns$. zodat: $f_{max} = 6.67Mhz$.

level 0: 1 optelling en 2 vermenigvuldigingen parallel

level 1: 1 optelling en 2 vermenigvuldigingen parallel

level 2: 2 optellingen parallel.

2. Er is veel discussie mogelijk bij de hiervoor genoemde aannames:

Behalve optelling en vermenigvuldiging zijn er verschillende andere tijd vergende operaties die nodig zijn als een digitale signaalprocessor een tijdsdiskreet netwerk moet realiseren dat b.v. moet werken in een analoge omgeving. Het is dan nodig dat er A/D conversie en D/A conversie gebeurt. Deze conversies kosten tijd, een tijd die meestal afhankelijk zal zijn van de nauwkeurigheid in bits die bereikt moet worden. De tijd die

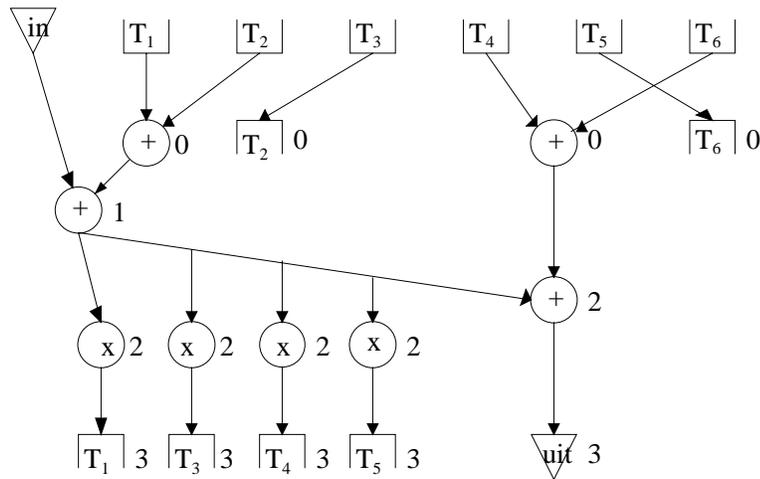


Figure 5.6: Precedence diagram behorend bij figuur 5.5

b.v. een A/D conversie kost, moet worden opgeteld bij de tijd die het tijdsdiskrete netwerk kost aan rekestijd en het gevolg is een lagere bemonsteringsfrequentie voor het gerealiseerde filter. Veelal zorgt een signaalprocessor zelf voor de vast in te stellen bemonsteringsfrequentie. Elke keer dat een nieuw monster beschikbaar is wordt het tijdsdiskrete rekenwerk in de vorm van een interrupt subroutine afgehandeld,, waarna de processor wacht op het volgende monster aan de ingang van het netwerk. Als de bemonsteringsfrequentie omhoog moet, is het zaak uit te zoeken of tijdens de tijd die de A/D omzetting kost ook rekenwerk gedaan kan worden, een vorm van parallelisme dus.

In termen van benodigde hardware of chipoppervlak is het ook niet vanzelfsprekend aan te nemen dat optelling en vermenigvuldiging evenveel tijd kosten. Een vermenigvuldiging is een veel ingewikkelder rekenoperatie als een optelling en kost dus of meer tijd of meer hardware of chipoppervlak. Als we zouden aannemen dat een vermenigvuldiging veel meer tijd kost als een optelling dat loont het wellicht de moeite de hardware zo in te richten dat tijdens een vermenigvuldiging nog iets anders gedaan kan worden. Bovendien kan de tijd die een vermenigvuldiging kost afhankelijk gemaakt worden van de gewenste precisie van een vermenigvuldigingscoefficient. Het is altijd nuttig de mogelijkheden van een signaalprocessor op dit gebied te onderzoeken.

Als meerdere processoren in parallel werken aan hetzelfde rekenwerk voor

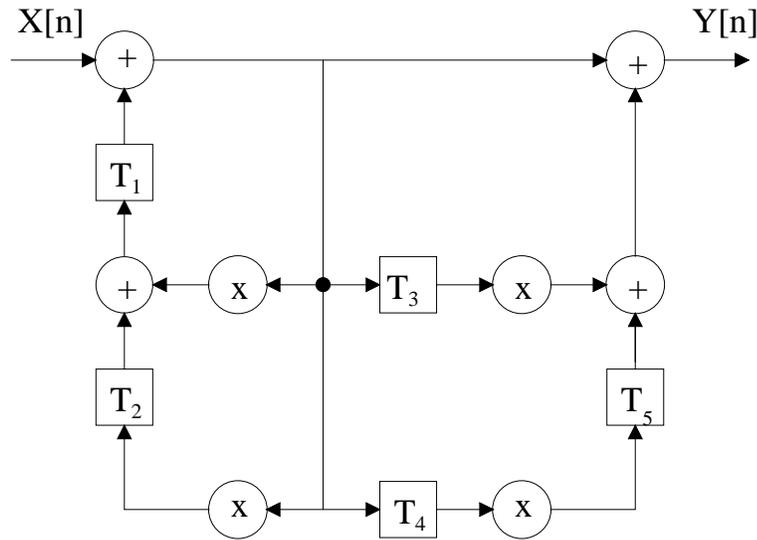


Figure 5.7: Rekursieve structuur

een tijdsdiskreet netwerk dan is het van belang dat eventuele uitwisseling van data tussen die processoren efficiënt geregeld wordt en dus niet teveel tijd in beslag neemt. Het kan anders zo uitpakken dat het gebruiken van meerdere processoren weinig tijdwinst oplevert. Dit spreekt des te sterker omdat uitwisseling van gegevens tussen eenvoudige signaalprocessoren meestal bitserieel gebeurt.

Uit een precedence diagram volgt meestal dat een aantal bewerkingen tegelijkertijd mogelijk is. Bijvoorbeeld twee vermenigvuldigingen. Hiervan kan dan alleen worden geprofiteerd als de signaalprocessor dit ook mogelijk maakt. Veelal is echter in een signaalprocessor slechts één bewerking tegelijk mogelijk. Als een signaalprocessor wel enkele bewerkingen tegelijk kan uitvoeren dan is het zaak bij een gegeven netwerk een precedence diagram (eventueel via transformaties) te vinden waarin hiervan zo goed mogelijk gebruik gemaakt wordt.

3. Gebruik maken van parallelisme betekent dat één of meer vertikaal lopende lijnen getrokken worden in een precedence diagram. Hiermee wordt het diagram verdeeld in stukken. Dit is bijvoorbeeld gedaan in figuur 5.4. De afgezonderde delen (in figuur 5.4: 2) worden dan verwerkt door parallel werkende rekenhardware componenten die ook variabelen moeten onderling kunnen uitwisselen. De waarde van de concurrency factor is (afgerond) een minimale maat voor het aantal zinvolle stukken waarin we

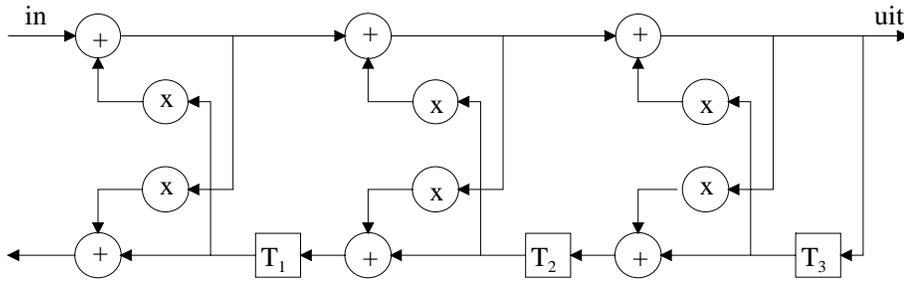


Figure 5.9: modelfilter

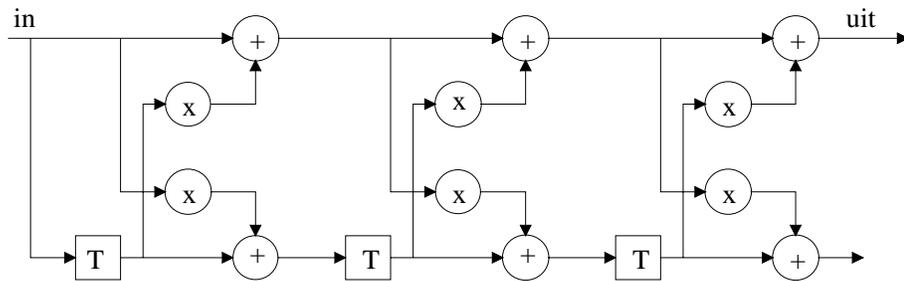


Figure 5.10: predictiefilter

In figuur 5.12 zijn een aantal transformaties aangegeven die we mogen toepassen op de filterstructuren die we bestuderen. De nummers 1, 2 en 3 hebben invloed op de plaatsing (en nummer 1 en 3 op het aantal) van de geheugenelementen T in de structuur. Bij de nummers 2 en 3 uit figuur 5.12 vindt een verschuiving plaats van een bewerking t.o.v. de geheugenelementen T . Deze transformaties hebben alle directe invloed op het precedence diagram van de structuur.

Een voorbeeld is gegeven in figuur 5.13 (figuur 5.13A is dezelfde als figuur 5.3). Achtereenvolgens zijn de transformaties 5.12-1 en 5.12-2 uitgevoerd op T_1 (resultaat figuur 5.13B). Daarna is in figuur 5.13B op de geheugenelementen T_1 en T_2 (in serie) de transformaties 5.12-1 en 5.12-2 toegepast met als resultaat figuur 5.13C.

Figuur 5.12B bevat 4 geheugenelementen, figuur 5.12C bevat 6.

Controleer zelf dat de overdrachtsfunctie van figuur 5.13C en 5.13B echter dezelfde is als 5.13A!

Figuur 5.13C komt overeen met het voorbeeld van figuur 5.5. Het gevolg is dat de precedence diagrammen figuur 5.4 en 5.6 beide behoren bij dezelfde tweede orde rekursieve sectie van figuur 5.3.

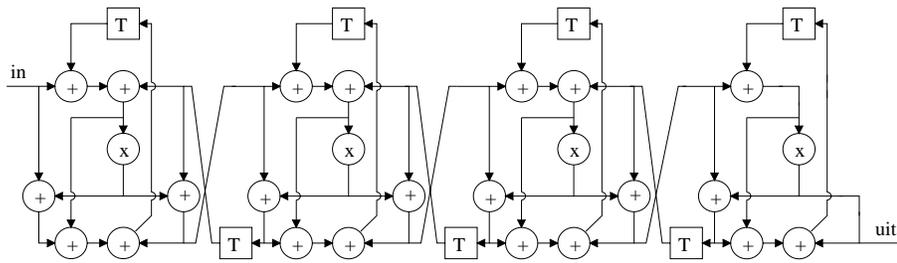


Figure 5.11: golffilter

We hebben geconcludeerd dat het maximale level van figuur 5.6: 3 is en van figuur 5.4: 4. Met andere woorden: de transformaties in figuur 5.13 heeft ons een lager maximaal level opgeleverd!

Bepaal zelf het precedence diagram en maximaal level van figuur 5.13B.

Een tweede voorbeeld van het gebruik van de transformaties is gegeven in figuur 5.14.

Ga zelf na, welke transformaties worden gedaan om van figuur 5.14A (zelfde als figuur 5.3), via 14B en 14C te komen bij figuur 5.14D (dit is ook figuur 5.7). Het precedence diagram van 5.14D is dus ook gegeven in figuur 5.8

Bepaal zelf het precedence diagram en maximaal level van de figuren 5.14B en 5.14C.

We kunnen dus de eigenschappen van de figuren 5.4, 5.6 en 5.8 vergelijken want ze komen alledrie overeen met een realisatie van figuur 5.2.

Figuur 5.4: maximale level is 4.

Figuur 5.6: maximale level is 3.

Figuur 5.8: maximale level is 3.

Conclusie: de figuren 5.6 en 5.8 maken een snellere realisatie mogelijk t.o.v. figuur 5.4.

Figuur 5.4: level 0: 4 vermenigvuldigingen parallel.

Figuur 5.6: level 2: 1 optelling + 4 vermenigvuldigingen parallel.

Figuur 5.8: level 0+1: 1 optelling + 2 vermenigvuldigingen parallel.

Conclusie: figuur 5.8 heeft de beste verdeling van bewerkingen over de levels want met maximaal 1 opteller en 2 vermenigvuldigers parallel werkend wordt het schema in 3 levels berekend.

Het schuiven met geheugenelementen wat we hebben gedaan d.m.v. de transformaties heet

retiming.

Dat figuur 5.8 met de kleinste hoeveelheid parallel werkende hardware het rekenwerk klaart, wil zeggen dat figuur 5.8 het gunstigst werkt voor wat be-

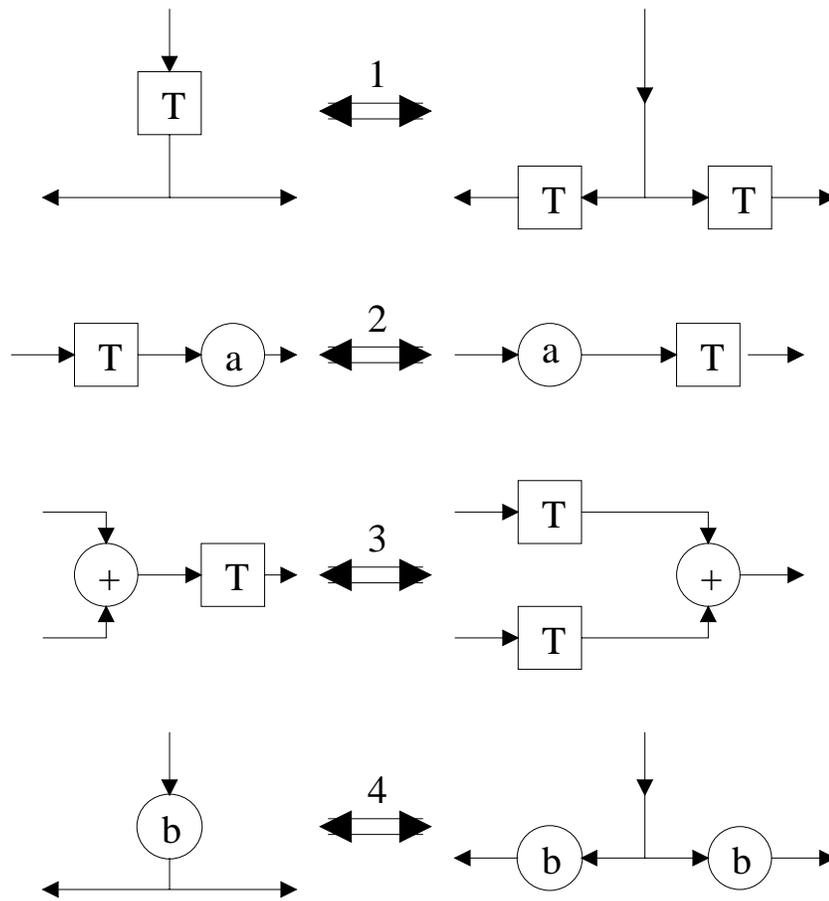


Figure 5.12: transformaties

treft:

resource sharing.

5.4 Realisatie in programmeerbare hardware

We kijken nog eens naar figuur 5.4 (opnieuw getekend in figuur 5.15).

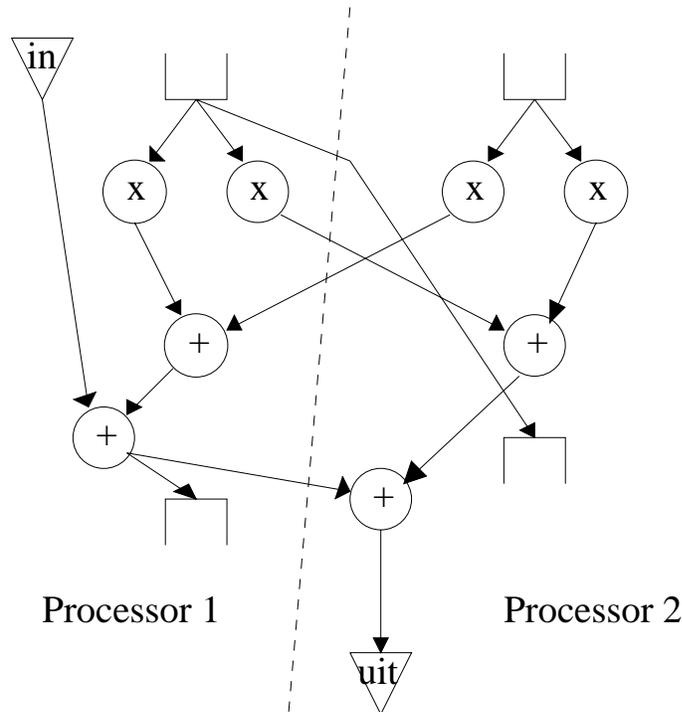


Figure 5.15: figuur 5.4, opgedeeld in 2 processoren

Het valt op dat we kunnen onderscheiden:

- Geheugen nodig voor de opslag van de waarden T. In een programmeerbare omgeving zal dit waarschijnlijk leiden tot een RAM.
- Rekenbewerkingen zoals optelling en vermenigvuldiging (met een constante). Dit leidt tot wat in het algemeen heet een rekenorgaan of ook veel genoemd: een ALU (Arithmetic Logic Unit).
- Een ingang en een uitgang. De ingang bevat mogelijk getallen afkomstig van een A/D (Analoog-Digitaal) omzetter. Op de uitgang is mogelijk een D/A (Digitaal/Analoog) omzetter aangesloten.
- In figuur 5.15 is duidelijk dat er tussen de aangegeven 2 processoren getallen moeten worden uitgewisseld. Dit kan met een of meer zogenaamde I/O poorten.

De beschrijving van een mogelijke filterprocessor die we in het vervolg van deze paragraaf zullen gebruiken, volgt nu. (figuur 5.16).

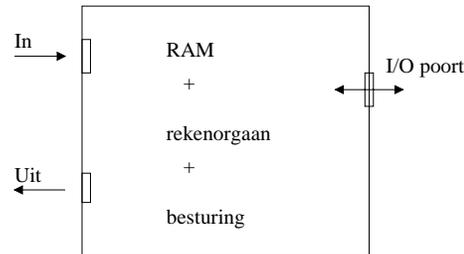


Figure 5.16: processor

Definitie:

- De ingangen en uitgangen zijn parallel en van een voldoende woordbreedte, evenals de processor zelf. Of het een fixed-point of floating-point processor is laten we in het midden. Wat we afspreken is dat elke bewerking één klokcyclus duurt. Hiermee hebben we al zeer belangrijke afspraken gemaakt over de eigenschappen van de processor.
- De ingang is een register waar alleen door de processor uit gelezen kan worden.
- De uitgang is een register waar alleen door de processor in geschreven kan worden.
- De I/O poort is een register waar door beide processoren die hiermee verbonden zijn zowel in geschreven als uit gelezen kan worden. Het wordt aan de programmeur van de processoren overgelaten om er voor te zorgen dat dit in goede banen wordt geleid. Bijvoorbeeld geeft het geen problemen als beide processoren het register tegelijk uitlezen, terwijl tegelijk in het register schrijven verboden is vanwege de onvoorspelbare resultaten die dit kan opleveren.
- Het RAM geheugen bevat voldoende ruimte om alle variabelen en coëfficiënten op te slaan. We nemen aan dat de benodigde vermenigvuldigingscoëfficiënten reeds in de RAM ingelezen zijn.

- Het rekenorgaan is de schakeling die mogelijk maakt (samen met de RAM) dat elke bewerking slechts één klokslag duurt.
- De besturing verzorgt alle benodigde stuursignalen, die worden afgeleid van de in een besturingsgeheugen opgeslagen serie instructies voor de volgorde van bewerkingen die nodig zijn.
- We zullen steeds aannemen dat als begintoestand de geheugenplaatsen waar de T elementen in staan, de waarde nul bevatten.

We zijn nu in staat als voorbeeld een serie instructies te geven die de bewerkingen die in figuur 5.15 nodig zijn, uitvoert. Hierbij geldt de afspraak dat de 4 coëfficiënten opgeslagen worden verondersteld in de RAM geheugen plaatsen 0 tot en met 3. (we gebruiken dus slechts één processor). Het geheel is gegeven in figuur 5.17.

We zien dat in figuur 5.17 9 cycli nodig zijn voor alle bewerkingen.

In figuur 5.18 is hetzelfde rekenwerk verdeeld over 2 processoren met elk één poort. We zien dat dit 8 cycli kost!. In figuur 5.19 is hetzelfde gedaan maar nu heeft elk van de twee processoren twee I/O poorten en die zijn paarsgewijs met elkaar verbonden.

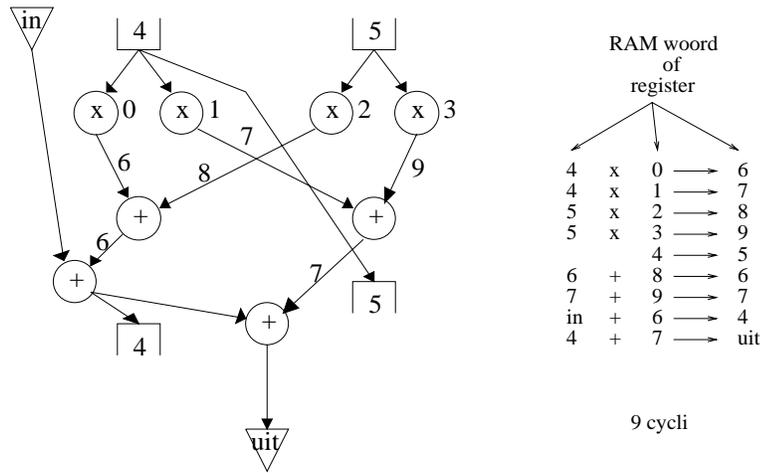


Figure 5.17: instructies voor één processor

In figuur 5.20 is één van de alternatieven voor figuur 5.4: figuur 5.6, genomen en het rekenwerk verdeeld over 2 processoren met elk 1 poort. We zien dat hier slechts 5 cycli per processor nodig zijn! Wel zijn dan de processoren een cyclus ten opzichte van elkaar verschoven. Processor 1 kan dus al met de volgende filterperiode beginnen als processor 2 nog bezig is met de huidige!

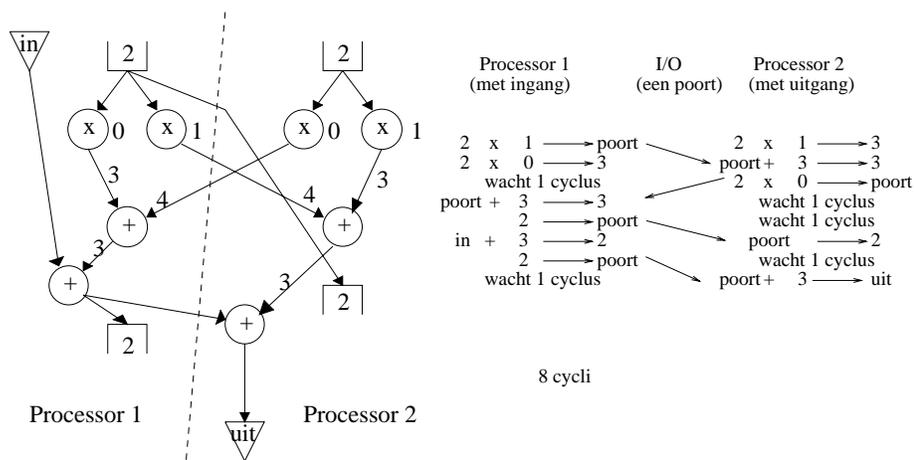


Figure 5.18: figuur 5.4, 2 processoren met één poort

We noemen dit:

pipelining op filtercyclus niveau

In figuur 5.21 is de andere variant van figuur 5.4: figuur 5.8, genomen en weer verdeeld over 2 processoren. Hieruit blijkt dat elke processor slechts één poort nodig heeft, die vervolgens met elkaar verbonden, al een berekening in 6 cycli mogelijk maakt.

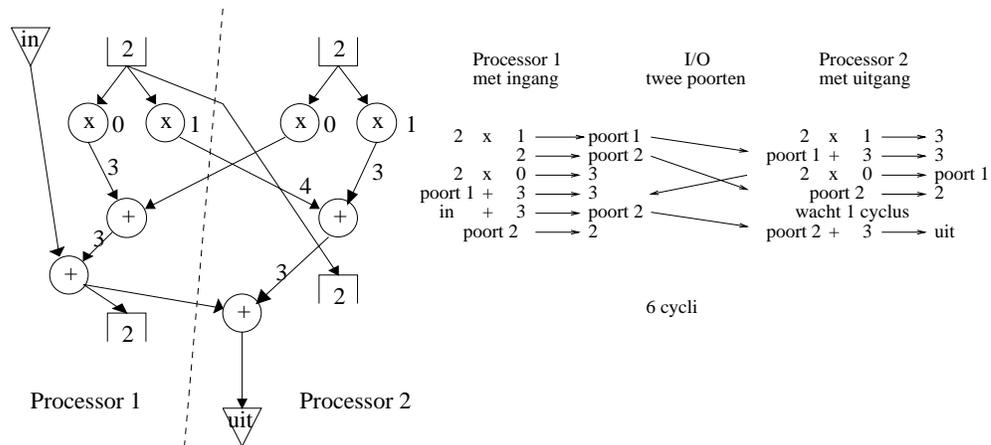
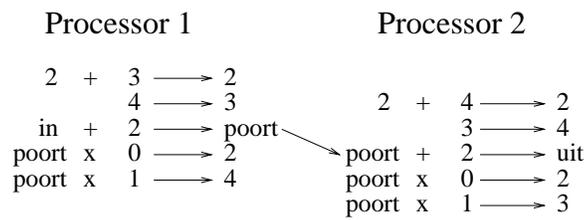
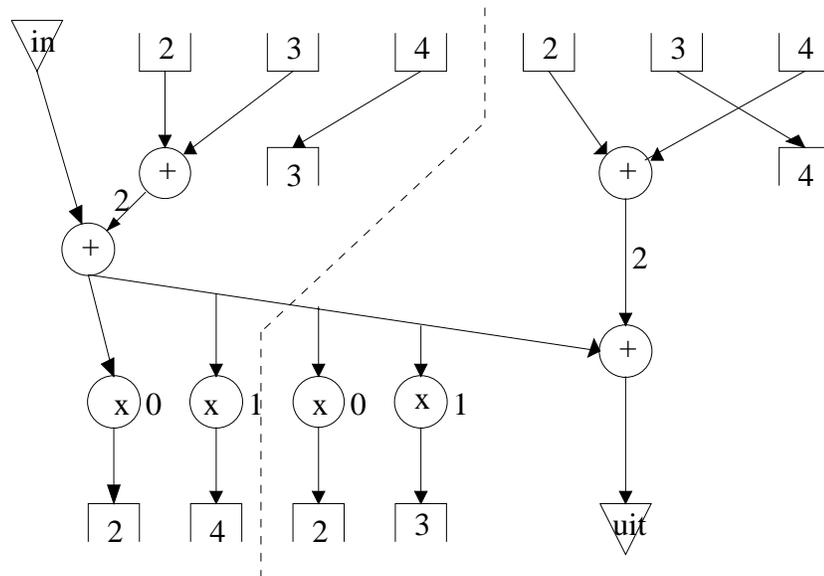


Figure 5.19: figuur 5.4, 2 processoren 2 poorten



5 cycli met pipelining, anders 6

Figure 5.20: figuur 5.6, 2 processoren 1 poort

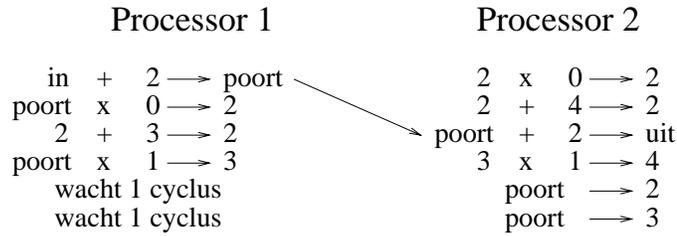
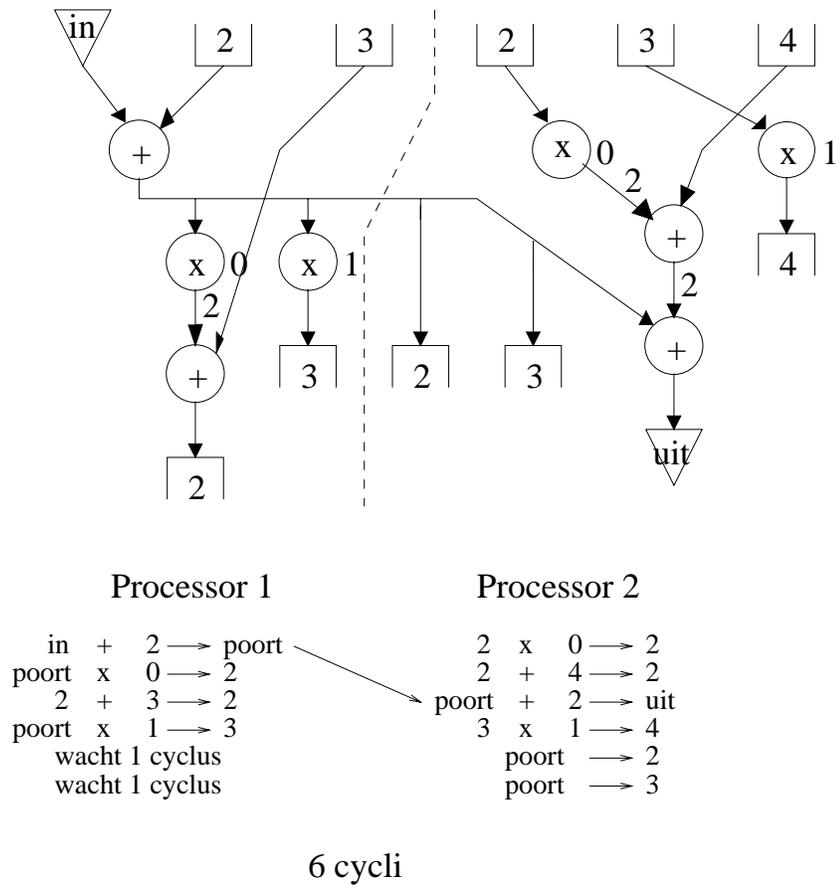


Figure 5.21: figuur 5.8, 2 processoren 1 poort

Opgaven:

1. Bepaal zelf het aantal cycli wat nodig is voor de verdeling van het rekenwerk zoals aangegeven is in figuur 5.22, met:
 - slechts één poort per processor,
 - 2 poorten per processor.
2. Bepaal enkele parallele realisaties voor de figuren 5.9, 5.10 en 5.11, m.b.v. exemplaren van de gedefinieerde processor.

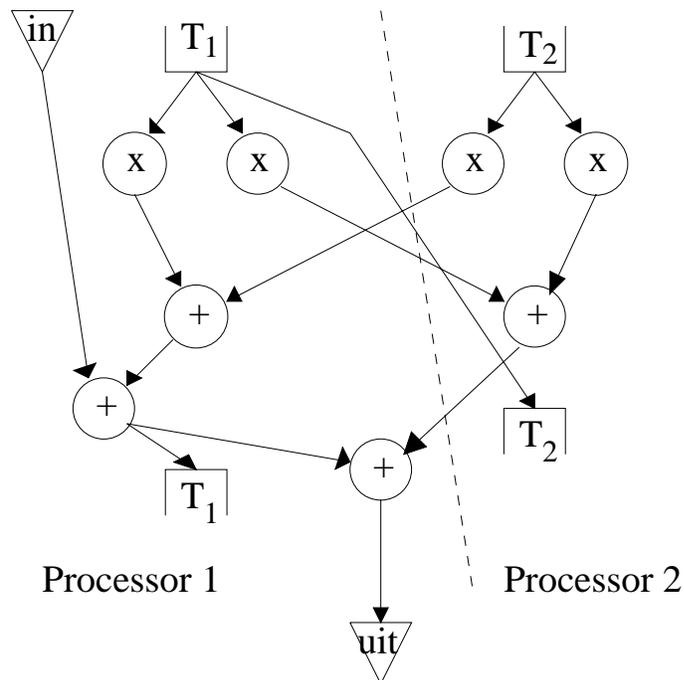


Figure 5.22: figuur 5.4 met andere verdeling over 2 processoren

5.5 Conclusies en opmerkingen.

1. Om enig inzicht te krijgen in de geschiktheid van filterstructuren voor implementatie d.m.v. snelle hardware hebben we de begrippen:

- level
- concurrency factor
- parallelisme en pipelining
- transformaties

ingevoerd voor gebruik op algorithmisch niveau.

Veel andere vrijheidsgraden die we zouden kunnen benutten om op zoek te gaan naar snelle en efficiënte implementaties blijven hier onbesproken. We volstaan met enkele te noemen:

- Er bestaan mogelijk alternatieve structuren, voor dezelfde overdrachtsfunctie, die geschikter zijn voor realisatie in snelle hardware.
- Een iets andere filterkarakteristiek, die ook de specificaties dekt, is mogelijk realiseerbaar met een geschiktere filterstructuur.

2. We hebben, zeer schematisch, een elementaire processor gedefinieerd.

Als we de eigenschappen hiervan wijzigen kan mogelijk het nut voor het doen van parallele rekenoperaties ingrijpend worden veranderd.

- Stel b.v. de I/O is niet parallel maar bitserieel. Een I/O operatie kost dan wellicht veel meer klokcycli, waardoor het opsplitsen in parallel werkende processoren veel minder nuttig wordt.
- Stel de vermenigvuldigingsoperatie kost b.v. 10 klokcycli (de hardware hiervan is hierdoor waarschijnlijk eenvoudiger en dus kleiner van chipoppervlak) en de opteloperatie 1 klokcyclus. In dit geval kan het nuttig zijn meerdere vermenigvuldigingsschakelingen in één processor op te nemen. We hebben dan de mogelijkheid van parallelisme binnen het rekenorgaan van één processor.
- Bestaande digitale signaalprocessoren zijn veel ingewikkelder van interne structuur dan onze definitie aangeeft. De reden is dat die signaalprocessoren ook geschikt gemaakt zijn voor andere signaalbewerkingen zoals b.v. Diskrete Fourier Transformatie.

De I/O tussen deze signaalprocessoren is vaak geregeld d.m.v. bit-seriele poorten met een eigen klok en "handshake" mechanisme.

3. Een echte VLSI implementatie van een digitaal netwerk stelt weer andere voorwaarden:

- Het benodigd chipoppervlak is direkt afhankelijk van de gebruikte woordlengte, minimalisatie hiervan is dus zinvol.
- Vermenigvuldigers en nauwkeurige coefficienten kosten veel ruimte en tijd op VLSI niveau, het loont dan de moeite netwerken te zoeken waarvan de vermenigvuldigingscoefficienten bestaan uit binaire representaties met slechts één of enkele énen. Hierdoor kan een vermenigvuldiging vervangen worden door één of enkele optellingen, wat ruimte bespaart.
- Tegenwoordig is "low-power" belangrijk voor signaalbewerking in draagbare apparatuur. Dit vergt in de praktijk extreme eisen op het gebied van parallelisme en pipelining op bitniveau.
- In het bijzonder voor F.I.R. netwerken is vaak de "multiplier-accumulator" structuur van belang. Deze realiseert de rekenoperatie:

$$Y_n = \sum_i \alpha_i x_i$$

Hierin zijn α_i coefficienten en x_i is de inhoud van delay element i . Y_n is de filteruitgang voor filtercyclus n . Hardware voor deze structuur levert de mogelijkheid van pipelining op bitniveau.

Chapter 6

Systemen met verschillende bemonsteringsfrequenties.

6.1 Inleiding

In signaalverwerkende systemen kan het nuttig zijn verschillende bemonsteringsfrequenties te gebruiken.

Het is ook vaak zo dat een bemonsteringsfrequentie niet een vrij te kiezen grootte is en dus wellicht een keer gewijzigd moet worden als systemen met verschillende frequenties gekoppeld worden.

In figuur 6.1 is nog weer eens de invloed van de bemonsteringsfrequentie op een signaal spectrum getekend.

Figuur 6.1a geeft een continue signaal $x(t)$ met een bijbehorend spectrum en daarin de hoogste frequentie f_b .

In figuur 6.1b wordt $x(t)$ bemonsterd met tijdstippen T_1 .

$$(T_1 = \frac{1}{f_1}, \quad f > 2f_b)$$

In figuur 6.1c wordt $x(t)$ bemonsterd met tijdstippen T_l .

$$(T_l = \frac{1}{f_l}, \quad f_l = 2f_b)$$

Verlaging van de bemonsteringsfrequentie zal aliasing opleveren omdat de spectra gaan overlappen. Verhoging van de bemonsteringsfrequentie zal betekenen dat de spectra verder uit elkaar schuiven.

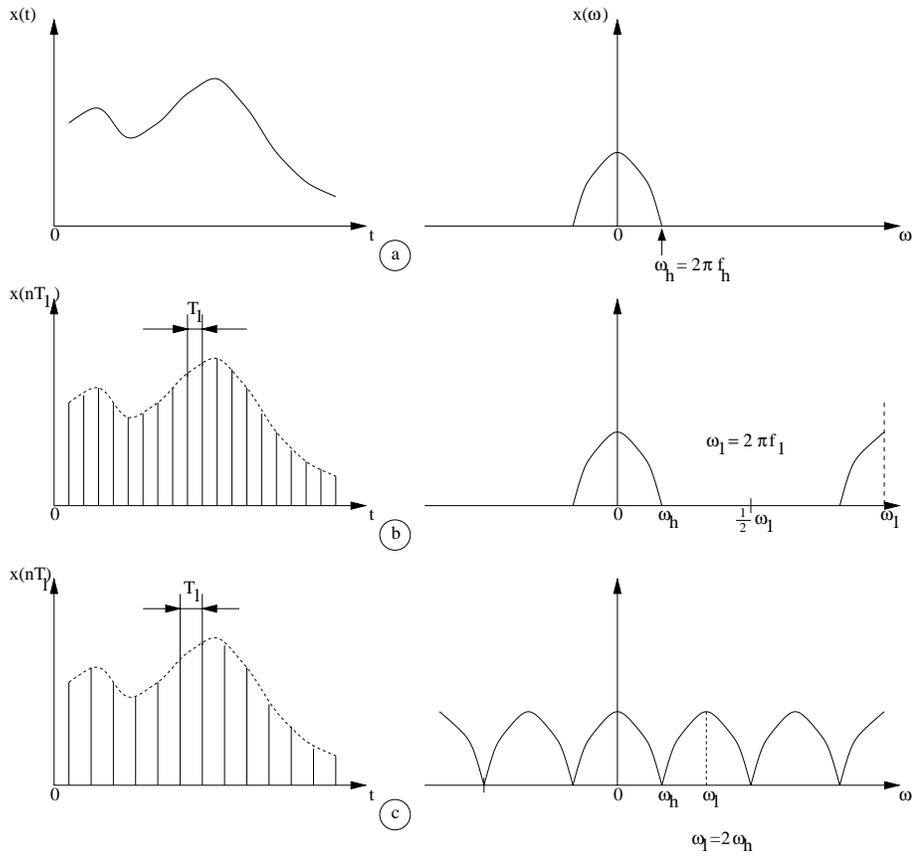


Figure 6.1:

6.2 Verlaging van de bemonsteringsfrequentie.

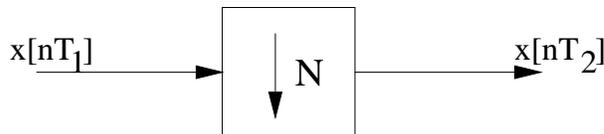


Figure 6.2:

We zullen ons beperken tot verlaging van die frequentie met factoren die een positief, geheel getal N zijn.

Het symbool is gegeven in figuur 6.2.

Hieruit volgt:

$$x[nT_2] = x[n.NT_1]$$

De faktor N heet de decimatie faktor.

Het effect is dat van elke N bemonsterde waarden in $x[nT_1]$ er slechts één waarde wordt overgenomen door $x[nT_2]$.

Het aantal bemonsteringen vermindert met de faktor N en daardoor wordt de bemonsteringstijd: $T_2 = NT_1$.

6.2.1 Het gevaar van "aliasing" als een decimatie-filter gebruikt wordt.

Stel een signaal $x[nT_1]$ ($T_1 = 1/f_1$) heeft een bemonsteringsfrequentie $2f_1$.

De bemonsteringsfrequentie moet verlaagd worden. Hiertoe wordt het signaal $x[nT_1]$ eerst gefilterd met een tijdsdiskreet filter zodanig dat de hoogst aanwezige frequentie in het fundamenteel interval weer kleiner wordt dan de helft van de nieuwe (lagere) bemonsteringsfrequentie.

Als dit tijdsdiskreet filter niet-ideaal is (in de praktijk is dit altijd het geval) worden niet alle ongewenste spektrum componenten geheel weggefilterd, slechts verzwakt. In verzwakte vorm komen deze spektrum componenten dus nog voor en kunnen aanleiding geven tot hinderlijke aliasing.

6.2.2 De realisatie van niet-rekursieve decimatie filters.

Een voorbeeld is gegeven in figuur 6.3.

In figuur 6.3 worden in het ritme T_1 vier vermenigvuldigingen gedaan en de resultaten opgeteld voor het bepalen van $y[nT_1]$.

Eén van de twee keer wordt dit resultaat echter niet gebruikt voor de uitgang $y[nT_2]$ omdat er een decimatie faktor 2 wordt gehanteerd.

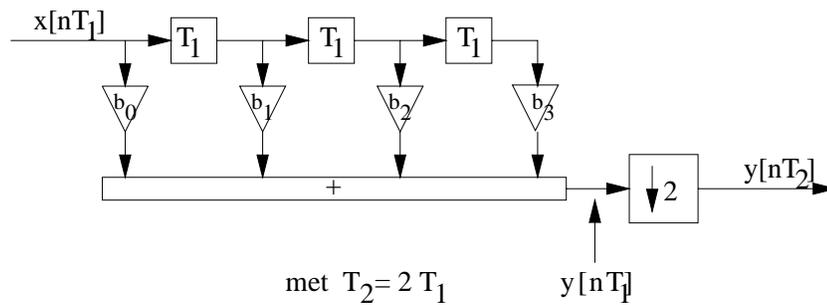


Figure 6.3:

Conclusie:

De berekeningen die nodig zijn voor de bepaling van $y[nT_1]$, kunnen één keer in de twee keren worden achterwege gelaten, het resultaat wordt toch niet gebruikt.

Dit betekent de situatie van figuur 6.4 waarin de decimatie factor in het reken-netwerk naar binnen is geschoven door de optelling en de vermenigvuldigingen heen.

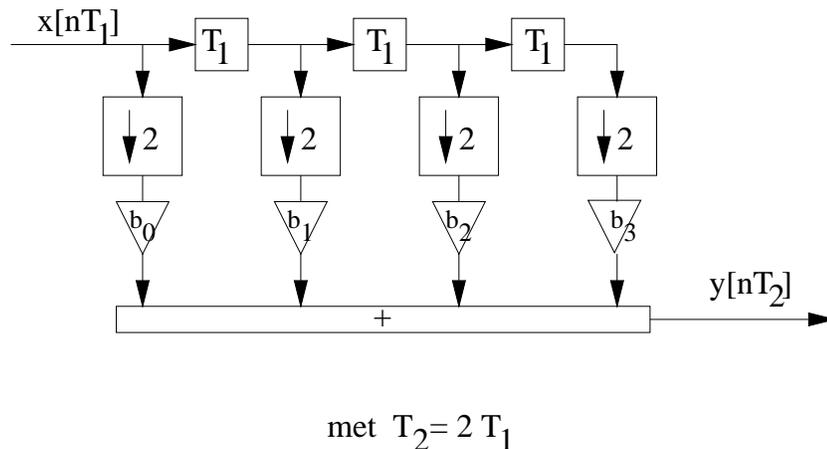


Figure 6.4:

Voor de 4 vermenigvuldigingen + optelling in figuur 6.4 is nu twee keer zoveel tijd beschikbaar omdat slechts de helft van de berekeningen in figuur 6.3 echt nodig blijkt.

Hierdoor kan het rekenwerk veréenvoudigd worden. Mogelijkheden om ditzelfde te doen voor rekursieve structuren worden hier niet besproken.

6.2.3 Voorbeeld van het gebruik van een decimator.

Veronderstel een audio-sigitaal $x(t)$ (figuur 6.5).

Hiervan wordt slechts het frekwentiedeel tussen 0 en 3.5 kHz van belang gevonden. Men wil dit signaal bemonsteren met een $f_c = 8$ kHz.

Om geen aliasing te krijgen moet het audio signaal dus zodanig gefilterd worden dat tussen 0 en 3.5 kHz wordt doorgelaten. Het filter mag een overganggebied van doorlaat- naar spergebied hebben van 3.5 kHz naar 4 kHz, daarboven moet alles gesperd worden. Dit signaal kan dan bemonsterd worden (A/D conversie).

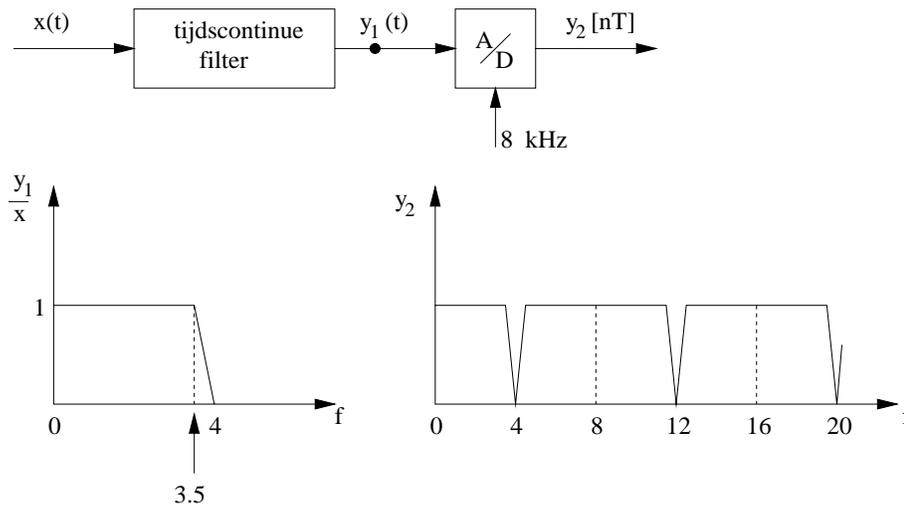


Figure 6.5:

Bij deze methode volgens figuur 6.5 is dus een selectief tijdscontinue filter nodig dat waarschijnlijk ingewikkeld wordt en dus lastig te maken in het algemeen.

We kunnen hetzelfde audioprobleem ook oplossen via figuur 6.6 en figuur 6.7.

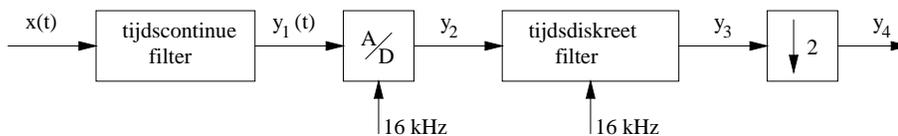


Figure 6.6:

$x(t)$:

zelfde audio signaal, alleen de band tussen 0 en 3.5 kHz wordt van belang gevonden.

$y_1(t)$:

moet 0 zijn vanaf 8 kHz, anders is er aliasing. Overigens van het deel van het frekwentie spektrum dat niet van belang gevonden wordt.

Het tijdsdiskrete deel levert het spektrum y_3 uit het spektrum y_2 .

Het tijdscontinue filter mag nu een veel groter overgangsgebied hebben en kan van een veel lagere orde zijn dan bij de eerste oplossing.

WE HEBBEN DUS TIJDSCONTINUE FILTERING INGEWISSELD VOOR TIJDSDISKRETE FILTERING!

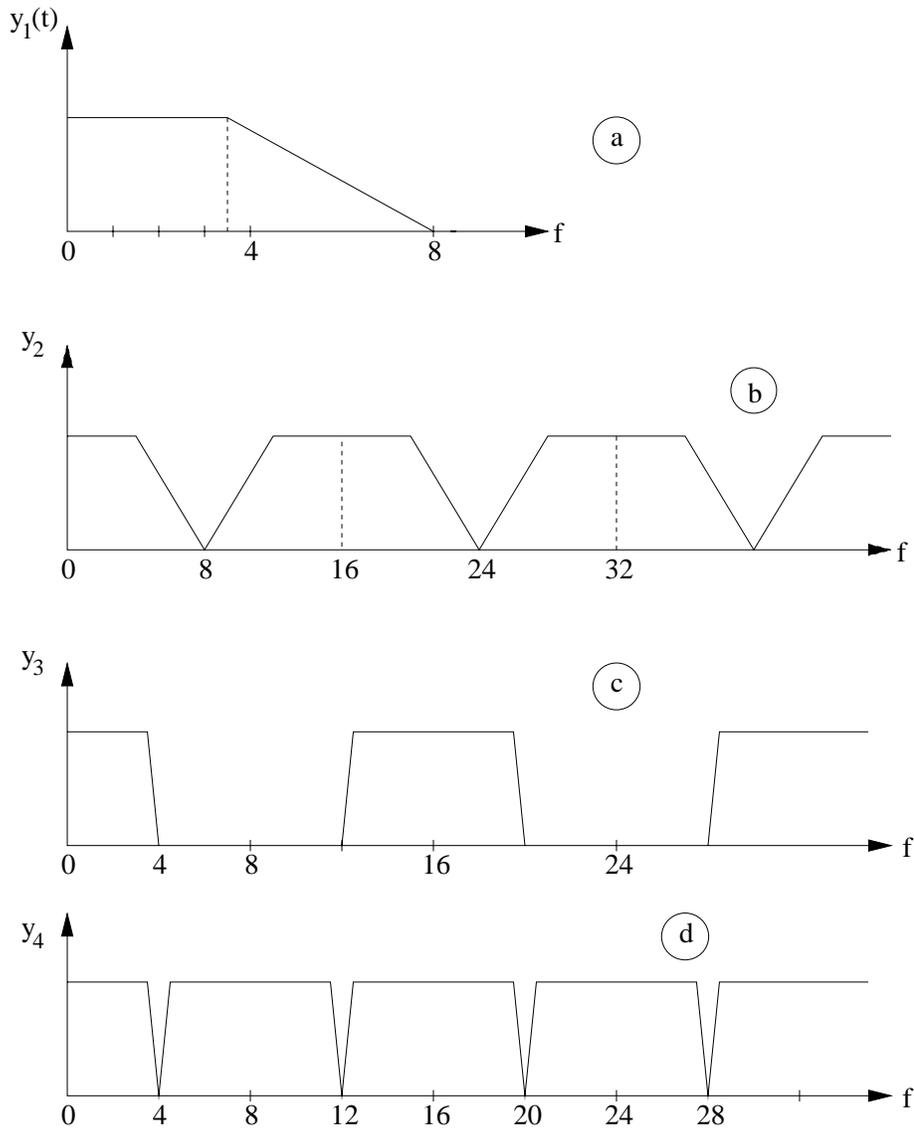


Figure 6.7:

6.3 Verhoging van de bemonsteringsfrequentie.

Deze situatie is geschetst in figuur 6.8 waarbij een verhoging met een positief geheel getal (hier 3) is gebruikt.

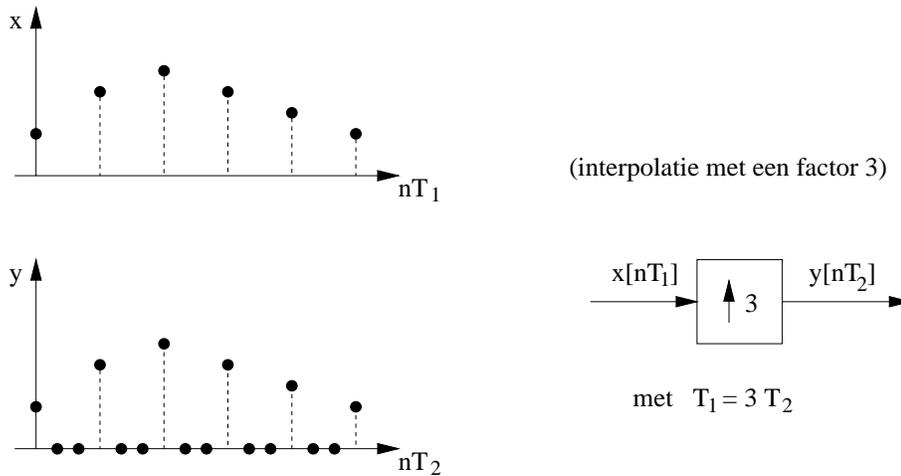


Figure 6.8:

De spektra die horen bij $x[nT_1]$ en $y[nT_2]$, met hun respectievelijke fundamenteel interval, zijn getekend in figuur 6.9.

$$f_1 = \frac{1}{T_1} = 4kHz, \quad f_2 = \frac{1}{T_2} = 12kHz.$$

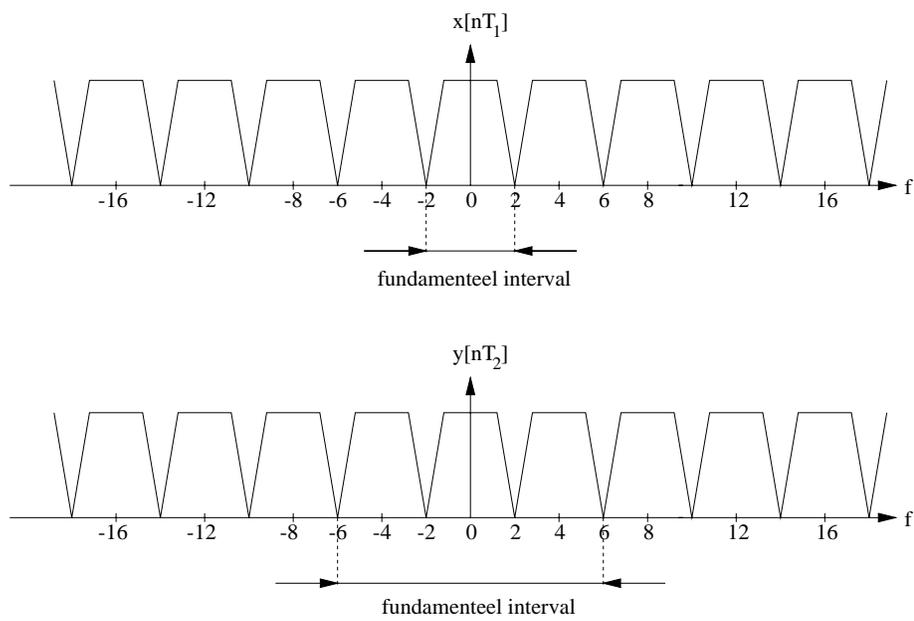


Figure 6.9:

6.3.1 Voorbeeld van het gebruik van een interpolator.

Veronderstel het resultaat van een audio-sigitaalverwerkingssysteem (tijdsdiskreet) is het signaal $x[nT]$ met een bemonsteringsfrequentie van 8 kHz. De hoogste signaalfrequentie in het fundamentele interval is 3 kHz (figuur 6.10).

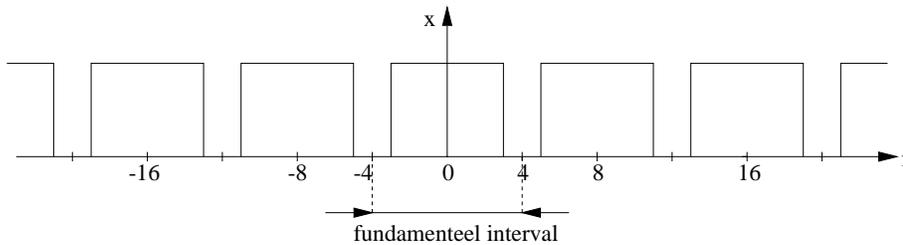


Figure 6.10:

Dit signaal moet d.m.v. D/A conversie weer teruggewonnen in tijdscontinue vorm, zie figuur 6.11.

Figuur 6.11 toont D/A conversie en tijdscontinue filtering om alleen het spectrum uit het fundamentele interval (figuur 6.11b) over te houden.

Hiervoor is een overdrachtsfunctie $H(j\omega)$ nodig met de frequentie karakteristiek die het verloop van figuur 6.11c minimaal (een groter overgangsgebied is niet mogelijk) dient te hebben.

We kunnen dit probleem ook oplossen volgens figuur 6.12.

Het tijdsdiskrete signaal van figuur 6.10 wordt eerst ge-interpoleerd met een factor 2. Met een tijdsdiskreet filter wordt daarna gefilterd zodanig dat daarna na D/A conversie, een veel eenvoudiger tijdscontinue filtering nodig blijft.

In figuur 6.13 zijn de signalen d.m.v. hun spektra getekend.

Het benodigde frequentie spectrum van het tijdsdiskreet filter is getekend in figuur 6.14.

Het benodigde tijdscontinue filter moet vanuit figuur 6.13c, figuur 6.13d opleveren. Figuur 6.15 toont het spectrum van dit filter met zo laag mogelijke orde. Dit is duidelijk een veel eenvoudiger tijdscontinue filter, vergeleken met het benodigde filter in de eerste oplossing.

Ook hier is dus tijdsdiskreete filtering gebruikt om de tijdscontinue filtering te vereenvoudigen.

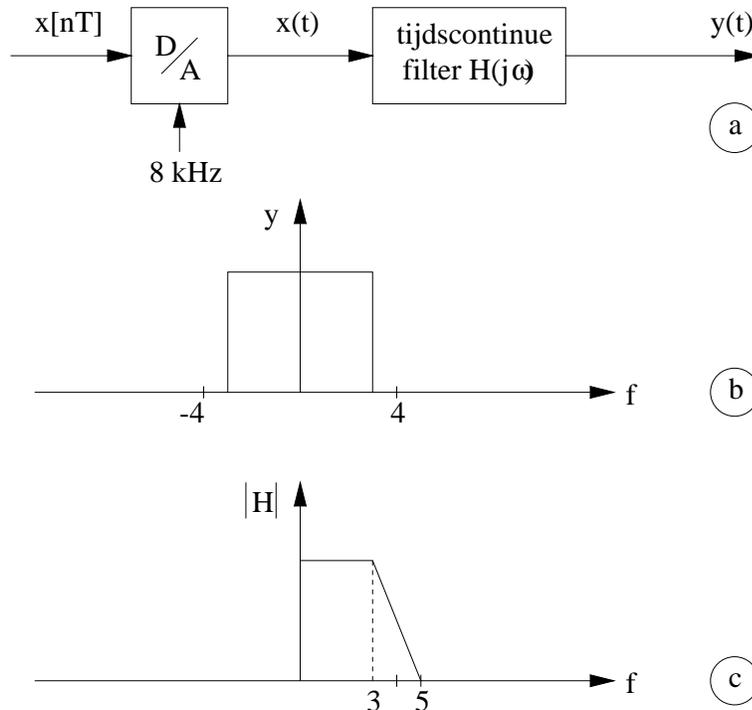


Figure 6.11:

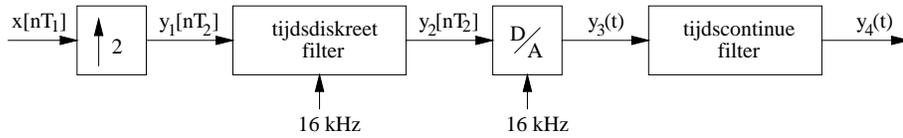


Figure 6.12:

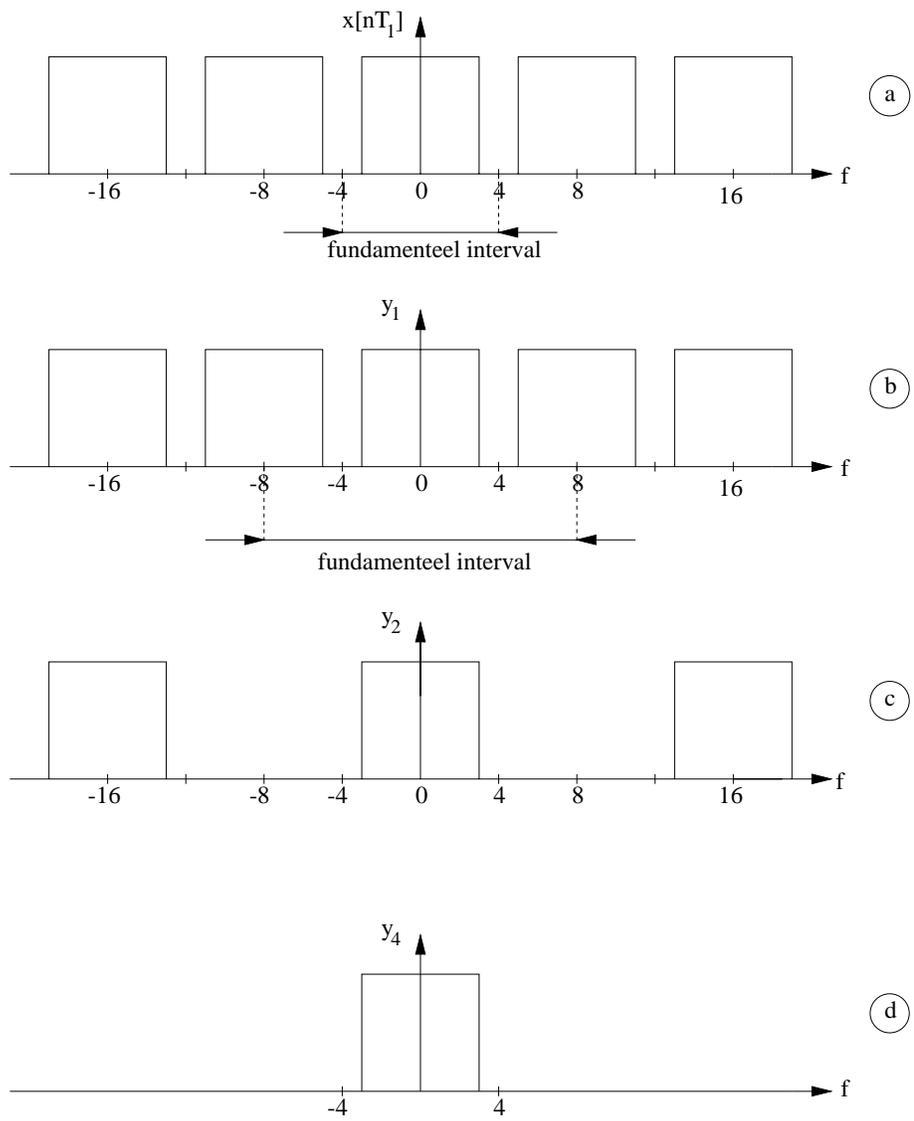


Figure 6.13:

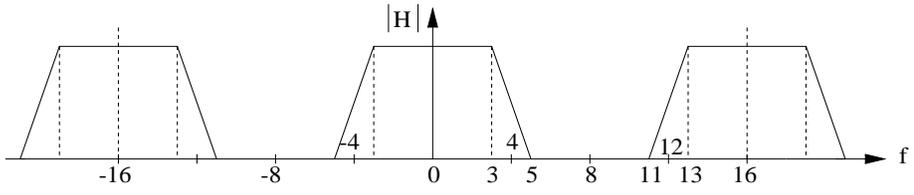


Figure 6.14:

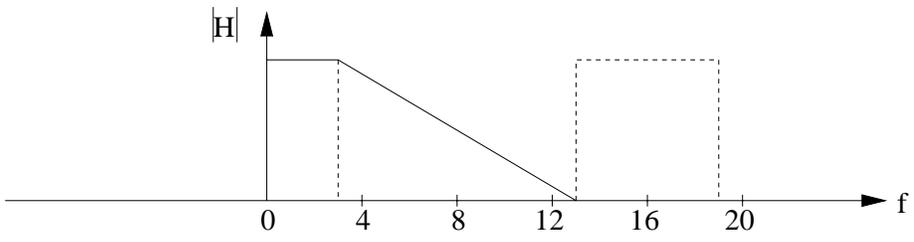


Figure 6.15:

6.4 Opgaven.

Opgave 1:

Gegeven is het tijdscontinue signaal $X_1(j\omega)$ met als hoogste frekwentie: $f_1 = 5kHz$. Dit signaal wordt bemonsterd met een frekwentie $f_2 = 12kHz$, het resultaat is signaal $X_2(j\omega)$.

Op dit signaal $X_2(j\omega)$ wordt interpolatie toegepast (interpolatiefactor n , n = geheel en positief). Het resultaat is $X_3(j\omega)$.

Om in het fundamentele frekwentie-interval van $X_3(j\omega)$ alleen het spektrum $X_1(j\omega)$ over te houden wordt een interpolatie filter gebruikt ($H(j\omega)$). Dit filter heeft een fundamenteel interval volgens figuur 6.16. ω_s is de sample-radiaal frekwentie van het filter.

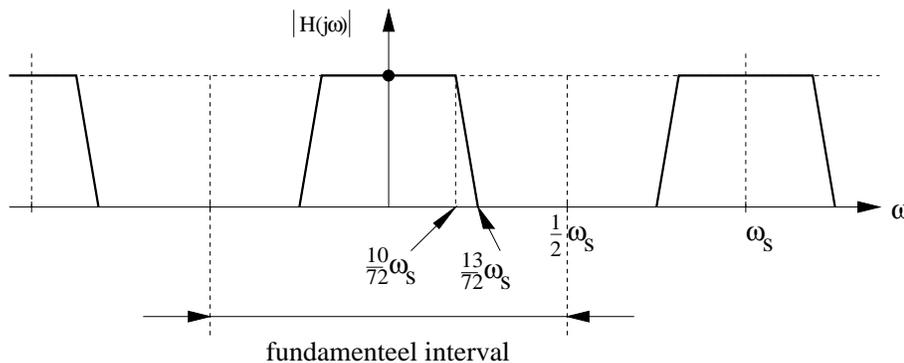


Figure 6.16:

Gevraagd:

- Teken het spektrum van $X_2(j\omega)$.
- Voor welke interpolatiefactor n is het interpolatie filter geschikt? Motiveren!
- Teken het spektrum $X(j\omega)$ dat uit het interpolatie filter komt.

Opgave 2:

Een analogoog signaal $x(t)$ heeft een frekwentie inhoud van 0 t.e.m. 15 kHz. Voor de verdere verwerking is slechts de frekwentie inhoud tussen 0 en 4 kHz van belang. Het is dan ook mogelijk van $x(t)$ een tijdsdiskreet signaal te maken met een bemonsteringsfrekwentie van 10 kHz. We zullen dit doen op 2 manieren:

Methode a:

Door middel van analoge filtering en bemonstering op 10 kHz via het schema van figuur 6.17.

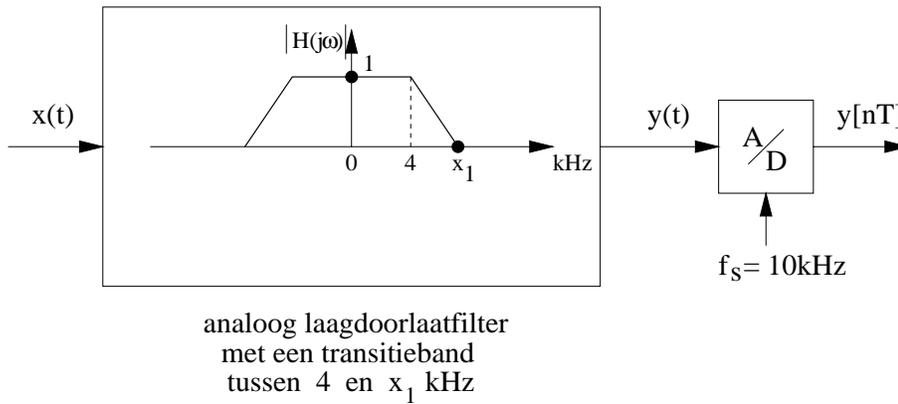


Figure 6.17:

Gevraagd:

Teken het spektrum van $y[nT]$ met x_1 zodanig hooggekozen dat geen aliasing valt in de band tussen 0 en 4 kHz. Wat is de grootste waarde van x_1 ?

Methode b:

Door middel van analoge filtering, bemonstering op 20 kHz en daarna d.m.v. een decimator naar 10 kHz volgens figuur 6.18.

Gevraagd:

Teken het spektrum van $y_1[nT_1]$, $y_2[nT_1]$, $y_3[nT_2]$ en motiveer de keus van x_2 en x_3 .

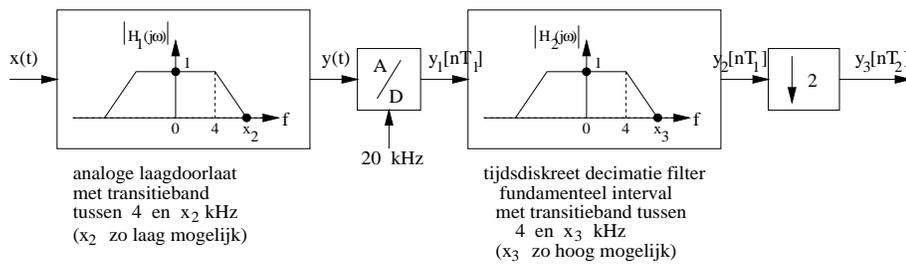


Figure 6.18:

Chapter 7

Eindige woordlengte effecten in tijdsdiskrete systemen.

7.1 Inleiding

De belangrijkste effecten die we bespreken zijn:

1. Kwantisatie.

Dit gebeurt b.v. bij A/D omzetting als het bemonsterde signaal moet worden omgezet in een digitale waarde die slechts bestaat uit een beperkt aantal bits.

Kwantisatie treedt ook op als vermenigvuldigingscoëfficiënten slechts een eindige nauwkeurigheid hebben omdat slechts een beperkt aantal digits beschikbaar is.

2. Overflow.

Overflow treedt op tijdens een rekenproces als de uitkomst van een berekening niet meer gerepresenteerd kan worden in het aantal digits dat gebruikt wordt.

Hierdoor kan een zeer storend nietlineair effect van "overflow-oscillatie" ontstaan. Deze oscillaties kunnen blijven bestaan als het signaal zelf niet meer aanwezig is.

3. Underflow.

Het gedrag bij een rekenproces als zeer kleine getallen worden verwerkt, kan flink afwijken van het lineaire, gewenste, gedrag.

De benodigde beperking in de digit representatie na een vermenigvuldiging is de oorzaak.

7.2 Kwantisatie

We onderscheiden drie soorten kwantisatie karakteristieken:

1. rounding
2. magnitude truncation
3. value truncation

Bij rounding van een gebroken getal naar het dichtstbijzijnde gehele getal wordt het gebroken gedeelte tussen 0 en 0.5 afgerond naar 0.

d.w.z:

$$\begin{aligned}x &= 0 & \text{voor } 0 \leq x < 0.5 \\x &= 1 & \text{voor } 0.5 \leq x < 1\end{aligned}$$

Dus voor "rounding":

$$\begin{array}{ll}10.4 \Rightarrow 10 & -10.4 \Rightarrow -10 \\10.5 \Rightarrow 11 & -10.5 \Rightarrow -11 \\10.6 \Rightarrow 11 & -10.6 \Rightarrow -11\end{array}$$

Het teken is hierbij dus niet van belang.

Magnitude truncation:
(afbreken van de absolute waarde)

$$\begin{array}{ll}10.4 \Rightarrow 10 & -10.4 \Rightarrow -10 \\10.5 \Rightarrow 10 & -10.5 \Rightarrow -10 \\10.6 \Rightarrow 10 & -10.6 \Rightarrow -10\end{array}$$

Ook hierbij is het teken niet van belang. De gebroken rest wordt gewoon weggelaten.

Value truncation:

Voor positieve getallen werkt dit gelijk aan magnitude truncation, voor negatieve getallen niet.

$$\begin{array}{ll}10.4 \Rightarrow 10 & -10.4 \Rightarrow -11 \\10.5 \Rightarrow 10 & -10.5 \Rightarrow -11 \\10.6 \Rightarrow 10 & -10.6 \Rightarrow -11\end{array}$$

Er wordt steeds het gehele getal genomen dat er onder ligt.

Voor kwantisatie van vermenigvuldigingscoëfficiënten heeft deze eindige precisie tot gevolg dat polen en nulpunten van een overdrachtsfunctie in z niet meer op elke plaats in het z -vlak kunnen liggen maar slechts op een diskreet, eindig aantal posities. Dit zal per realisatie-structuur verschillen.

Kwantisatie bij A/D omzetting heeft tot gevolg dat het tijdsdiskrete spectrum niet slechts de periodieke versie is van het spectrum van het continue signaal. Er is kwantisatieruis aan toegevoegd.

7.3 Overflow:

We onderscheiden 3 overflow karakteristieken:

1. "Verzadiging"
2. "Nul-maken"
3. "Zaagtand"

"Verzadiging":

Veronderstel dat in een rekensysteem een getal slechts kan liggen tussen -1000 en +1000. De optelling b.v. van de twee getallen 510 en 620 levert een te groot getal op dat buiten het bereik valt, aan de bovenkant.

Bij gebruik van de verzadigingskarakteristiek wordt dan het grootste getal (aan de bovenkant hier dus) genomen dat nog gerepresenteerd kan worden.

Voorbeelden:

$$\begin{array}{rclcl} 510 & + & 620 & \Rightarrow & 1000 \\ 510 & - & 620 & \Rightarrow & -110 \\ -510 & - & 620 & \Rightarrow & -1000 \\ 510 & \times & 2 & \Rightarrow & 1000 \\ -510 & \times & 2 & \Rightarrow & -1000 \end{array}$$

"Nul-maken":

Bij hetzelfde bereik als daarnet, wordt nu een resultaat dat buiten het bereik valt, vervangen door het getal nul.

Voorbeelden:

$$\begin{array}{rcl}
510 & + & 620 \Rightarrow 0 \\
510 & - & 620 \Rightarrow -110 \\
-510 & + & 620 \Rightarrow 0 \\
510 & x & 2 \Rightarrow 0 \\
-510 & x & 2 \Rightarrow 0
\end{array}$$

”Zaagtand”:

We veronderstellen weer hetzelfde bereik.

Wordt een getal nu te groot dan wordt het totale bereik er afgetrokken. Een te groot positief getal wordt dus negatief, en omgekeerd.

Voorbeelden:

$$\begin{array}{rcl}
510 & + & 620 \Rightarrow 1130 - 2000 = -870 \\
510 & - & 620 \Rightarrow -110 \\
-510 & - & 620 \Rightarrow -1130 + 2000 = +870 \\
510 & x & 2 \Rightarrow 1020 - 2000 = -980 \\
-510 & x & 2 \Rightarrow -1020 + 2000 = +980
\end{array}$$

Een grensgeval ontstaat als het getal ± 1000 ontstaat. Of hier nu 2000 wordt afgetrokken (of opgeteld)

Deze zaagtandkarakteristiek is echter afkomstig uit de 2’s complement reken-techniek bij een eindig aantal bits. Ontstaat hier een te groot (positief of negatief) getal dan ontstaat automatisch een karakteristiek met een zaagtand-vorm als het foute 2’s complement getal als resultaat wordt geaccepteerd. Er wordt dan verder geen correctie uitgevoerd.

7.4 Underflow.

Dit demonstreren we aan de hand van het voorbeeld van figuur 7.1.

Gevraagd wordt $y[nT]$ te berekenen voor $n=0, 1, 2, 3, 4, 5$. En dit voor de volgende 4 gevallen:

1. Q_1 heeft geen invloed, de berekening is lineair, nauwkeurig.
2. Q_1 is rounding.
3. Q_1 is magnitude truncation.
4. Q_1 is value truncation.

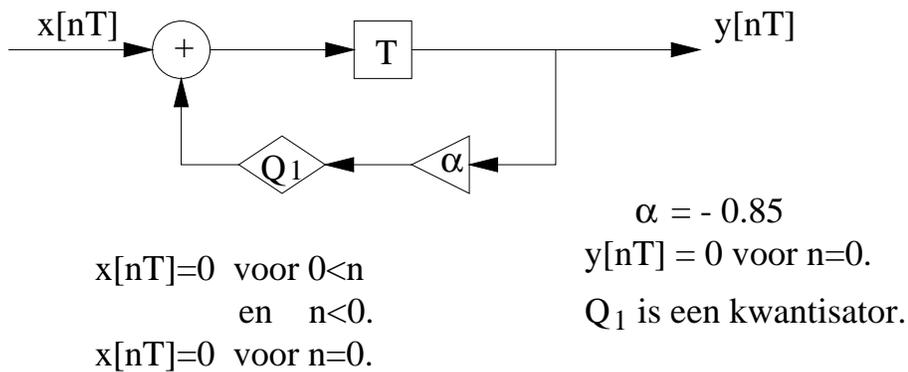


Figure 7.1:

Het resultaat hiervan is gegeven in de tabel van figuur 7.2.

Conclusie:

Bij de verwerking van kleine getallen wijkt de werking af van de lineaire situatie.

Er kunnen zelfs oscillaties optreden die blijven bestaan als er geen signaal meer is! Deze oscillaties heten "limit-cycles" en zijn een uiting van volledig nietlineair gedrag.

7.5 Opgaven:

Opgave 1:

Gegeven is de schakeling van figuur 7.3.

Vraag:

Welke waarden mag b aannemen zodanig dat $x_1[nT] = x_2[nT] = 1$ voor alle n vanaf $n=0$.

Opgave 2:

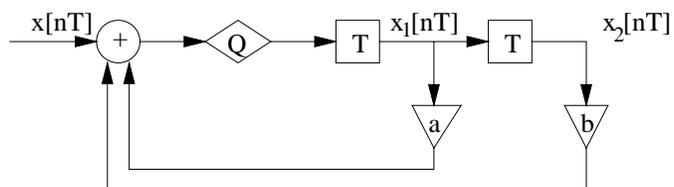
Gegeven is de schakeling van figuur 7.4.

Vraag:

Bereken de grafiek $y[nT]$ als functie van n vanaf $n=0$.

n	x[nT]	y[nT]			
		1	2	3	4
0	100	0	0	0	0
1	0	100	100	100	100
2	0	-85	-85	-85	-85
3	0	72.25	72	72	72
4	0	-61.4125	-61	-61	-62
5	0	52.2006	52	51	52

Figure 7.2:



Q is magnitude truncation,

$$a = 1.8$$

$x[nT] = 0$ voor alle n

$$x_1[0] = x_2[0] = 1$$

Figure 7.3:

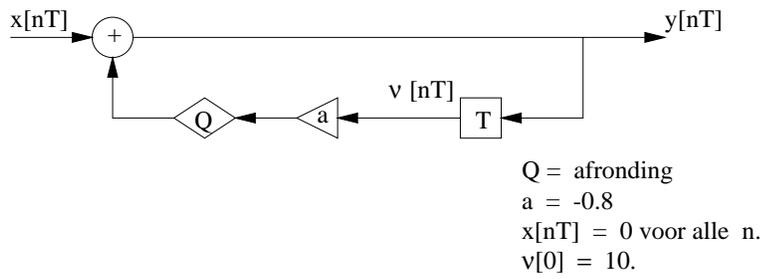


Figure 7.4:

7.6 A/D conversie kwantisatie.

.

.

.