Digitale signaalbewerking : theorie en toepassingen van digitale signaalbewerking : postacademische cursus, 15, 16, 22, 23, 29 en 30 november 1993 te Eindhoven

Citation for published version (APA):
Meer, van, A. C. P., & Verkroost, G. (1993). *Digitale signaalbewerking : theorie en toepassingen van digitale signaalbewerking : postacademische cursus, 15, 16, 22, 23, 29 en 30 november 1993 te Eindhoven*. Stichting voor Postacademisch Onderwijs in de Technische Wetenschappen.

Document status and date:
Gepubliceerd: 01/01/1993

Document Version:
Uitgevers PDF, ook bekend als Version of Record

POST-
ACADEMISCH
TECHNISCH
ONDERWIJS

Digitale Signaalbewerking

theorie en toepassingen van
digitale signaalbewerking

15, 16, 22, 23, 29 en 30 november 1993 in Eindhoven

cursusleiders

ing. A.C.P. van Meer
ir. G. Verkroost

Postacademische cursus

DIGITALE SIGNAALBEWERKING

Een 6-daagse cursus
op 15,16,22,23,29 en 30 november 1993
te Eindhoven

Cursusleiding
ing. A.C.P. van Meer
ir. G. Verkroost

PATO — Stichting voor postacademisch onderwijs in de technische wetenschappen

Sectie Elektrotechniek

Digitale Signaalbewerking

Inhoud van de cursus

Deel I Algemene behandeling

1. Signalen en systemen in continue tijd
2. Signalen en systemen in discrete tijd
3. z-Transformatie
4. Bemonstering van tijdcontinue signalen; interpolatie van tijddiscrete signalen
5. Realiseren van filters als samenstel van secties
6. Transformatie van tijdcontinue naar tijddiscrete systemen; het simuleren van tijdcontinue systemen
7. Bemonstering in het frekwentiedomein; discrete Fouriertransformatie
8. Tijdbegrenzing van signalen
9. Filters met een eindig lange impulsresponsie (FIR filters)
10. Verandering van bemonsteringsfrekwentie
11. Effecten van eindige woordlengte

Deel II Speciale onderwerpen

1. Programmeerbare Digitale Signaalprocessoren
   A.C.P. van Meer, TU Eindhoven
2. Speciale Filterstructuren
   J.H.F. Ritzerfeld, TU Eindhoven
3. Digitaal/Analoog- en Analoog/Digitaal omzetting
   M. van der Veen, Philips I & E Breda
4. Adaptieve Digitale Signaalbewerking
   P.C.W. Sommen, TU Eindhoven
5. Computer Aided Design (CAD) voor digitale signaalbewerking
   A.W.M. van den Enden, Philips Nat. Lab. Eindhoven
6. Digitale Beeldreconstructie in het Fourierdomein
   J. Biemond, TU Delft
7. Switched Capacitor Filters
   J.A. Hegt, TU Eindhoven
8. Spraaksynthese
   L.L.M. Vogten, IPO Eindhoven
9. Signaalbehandeling bij de CD
   R.W.C. Groen, Philips C.E. Eindhoven
10. Signaalbehandeling bij de DCC
    A.W.M. van den Enden, Philips Nat. Lab. Eindhoven

DIGITALE SIGNAALBEWERKING

Deel I

Algemene Behandeling

M.J. Bastiaans
G. Verkroost

Postacademische cursus
15,16,22,23,29,30 november 1993
Eindhoven

DIGITALE SIGNAALBEWERKING

deel I

Algemene Behandeling

De beschrijving vindt u in de syllabus van het
TUE-college "Digitale Signaalbewerking"

door

M.J. Bastiaans
G. Verkroost

Technische Universiteit Eindhoven

DIGITALE SIGNAALBEWERKING

Deel II

Speciale Onderwerpen

J. Biemond
A.W.M. van den Enden
R.W.C. Groen
J.A. Hegt
A.C.P. van Meer
J.H.F. Ritzerfeld
P.C.W. Sommen
M. van der Veen
L. Vogten

Postacademische cursus
15,16,22,23,29,30 november 1993
Eindhoven

DIGITALE SIGNAALBEWERKING

deel II.1

Programmeerbare Digitale Signaal Processoren

door

A.C.P. van Meer

Technische Universiteit Eindhoven

Programmeerbare Digitale Signaalprocessoren

A.C.P. van Meer

Technische Universiteit Eindhoven
Faculteit Elektrotechniek

Inhoud

6. Voorbeeld van een ontwerp.

6.1. Bepaling van de filtereigenschappen.
6.2. Bepaling van de filterstructuur.
6.3. Ontwerp van het systeem.
- Filter Design.
- Code-generator.
- Linker.
- Software Evaluation.
- Eindtest.

1. Inleiding.

1.1. Achtergronden van de DSP.

De golf van de digitale elektronika overspoelt een steeds groter deel van de elektronische bewerkingen. De signaalbewerking is een van de toepassingsgebieden die door de digitale golf zijn meegesleurd.

In de jaren '70 was een van de interessantste ontwikkelingen de komst van de microprocessorchip, een device dat vandaag de dag gemeengoed is geworden voor de elektronisch ontwerper. De miniaturisatie van de processor was een van de belangrijkste stappen die hebben geleid tot miniaturisatie van van de computer als geheel. Er is geen opzienbarende doorbraak geweest in de achterliggende principes; de ideeën die aan de processorchip ten grondslag liggen zijn in grote lijnen diegene, die in de enorme machines uit het verleden zijn toegepast.

Nieuw is echter het enorme gebied van toepassingen dat is komen open te liggen voor digitale bewerkingen. Toepassing van micro-elektronika in de digitale signaalbewerking is tamelijk traag op gang gekomen. De computers zijn natuurlijk ingezet in de processing van analoge signalen, maar de hoge kosten van snelle digitale rekeneenheden in het verleden beperkte de toepassingen tot bewerking van langzaam variërende signalen, off-line processing, of applicaties waar grote financiële uitgaven konden worden veroorloofd.

Dit plaatje is plotseling veranderd. Een nieuwe soort chip is op de markt gekomen. Dat chipfabrikanten bestaan bij de gratie van de grote aantallen is de reden dat zij zochten naar brede toepassingen voor bepaalde chips. Een van de toepassingen die een wijde wereld opende voor deze devices was de spraakbewerking. De spraakbewerking maakte het aantrekkelijk de microprocessor-architecturen van de jaren '70 te verlaten en iets nieuws te zoeken. Spraaksynthese was toen de eerste toepassing om een grote markt te bereiken. De chips die hier gebruikt werden zijn over het algemeen speciaal voor dit doel ontworpen en konden geen andere functies uitvoeren. In videospelletjes, homecomputers, auto's en vele andere apparaten worden deze chips in ruime mate ingebouwd.

De beperkte mogelijkheden (denk maar aan b.v. de eentonige uitspraak) spoorden de ontwerpers al snel aan om de prestaties te vergroten. Dit is zo ongeveer het punt waar de programmeerbare signaalprocessoren hun intrede doen. Het nivo waarop de programmatuur werd aangebracht was natuurlijk erg laag en was zeer gericht op de genoemde specifieke toepassing.

Spraaksynthese is ondertussen volwassen geworden, en spraakanalyse en -herkenning zijn hot-topics geworden. De belangrijkste componenten van spraakherkenning, bewerkingen als filtering, spectrale analyse en correlatie, zijn juist toepassingen waar de programmeerbare DSP zijn waarde kan bewijzen. Spraakbewerking is natuurlijk niet het enige gebied waar deze componenten kunnen worden toegepast. Digitale transmissie met o.a. detectie en correlatie van transmissie-errors is een toepassing waarin ook met redelijk beperkte snelheid digitale informatie wordt verwerkt. Sonar signaalprocessing is ook een toepassingsgebied, waarin echter bij toenemende bandbreedte de perfectie van de bewerking vermindert, maar waar de DSP een zeer waardevol hulpmiddel is.

Tenslotte is de universele programmeerbare DSP binnengedrongen in de ruime wereld van de signaalbewerking vanwege zijn toenemende rekenmogelijkheden en vooral vanwege zijn sterk toegenomen verwerkingssnelheid.
De elektronisch ontwerper van vandaag kan haast niet meer om de DSP heen.

1.2. De taken van de DSP.

De belangrijkste toepassingen van de digitale signaalbewerking liggen in de vakgebieden telecommunicatie, audiotechniek en spraaksynthese. De belangrijkste bewerkingen die hierbij worden uitgevoerd zijn lineaire bewerkingen van de voorkomende signalen, zoals convoluties en correlaties. De essentie van deze operaties is het vermenigvuldigen van twee operanden en het accumuleren van de produkten. Ook het vertragen (verschuiven) van signalen is een typische eigenschap van deze bewerkingen. Deze operaties zouden kunnen worden uitgevoerd met losse componenten, zoals multipliers, full adders en shift registers, maar de besturing van de betreffende schakelingen maakt het geheel echter zeer gecompliceerd. Het herhaald uitvoeren van genoemde taken, samen met het initialiseren en sturen van de processen maken dat dit een bij uitstek geschikte taak is voor een microcomputer of een microprocessor. Deze taken, gekoppeld met de mogelijkheid om op zeer hoge snelheid specifieke multiply/accumulate en shift instructies uit te voeren, hebben geleid tot de ontwikkeling van een speciale microprocessor, de digitale signaalprocessor.

1.3. Het ontstaan van de DSP.

Signaalprocessing is langer verbonden gebleven met zijn analoge verleden dan andere gebieden waar de computer snel zijn ingang heeft gevonden. De reden was het gebrek aan een voldoende snelle (en goedkope) vermenigvuldiger die het hart van de operatie zou moeten vormen. Het vervangen van een operationele versterker door een immense hoeveelheid digitale logica maakte dat zelfs de meest toegewijde ontwerper twee keer moest nadenken voor hij hieraan begon. De uitwerking van de digitale signaaltheorie (vanaf ca. 1960) is vooral op gang gekomen door het gebruik van general purpose computers in simulaties, waarbij men vrij was van real-time beperkingen.

Een belangrijke stap voorwaarts in de vervanging van analoge door digitale technieken was het beschikbaar komen van een digitale multiplier op een chip. Als eerste kwam TRW met een 16 x 16 bits vermenigvuldiger (MPY-16AJ). ondanks de 64 pootjes en een power-dissipatie van 8,0 Watt, waardoor grote koelvinnen nodig waren, was dit slechts een gering ongerief vergeleken bij de enorme borden met componenten voor dezelfde functie.

Hoe heeft men vervolgens de multiplier ingeschakeld in het produktieproces van de signaalbewerking?
De introductie van de multiplier is samengegaan met de introductie van de eerste microprocessor chips. De general-purpose microprocessor was echter een trage machine en om deze nu te gebruiken om de snelle multiplier (100 nsec) te bedienen is als het stoken van een stoommachine met een theelepel. De kolenschop die hiervoor nodig was is gekomen in de vorm van de bit-slice signaalprocessor.

De bit-slice systemen zijn eigenlijk bedoeld voor de realisatie van een general-purpose minicomputer, maar in veel research-laboratoria zijn hiermee universele programmeerbare DSP's gebouwd. Een bit-slice unit neemt, zoals de naam al zegt, de bewerking van slechts een deel van het datawoord voor zijn rekening, terwijl een aantal andere de rest aanpakken. Een speciale program-sequencer stuurt vanuit een brede instructiebus (>40 bits) gelijktijdig een aantal units die alle een bepaalde functie uitvoeren, zoals een multiplier, een arithmetic/logic unit, een shiftunit en een I/O-interface. Vanwege de directe aansturing en de hoge mate van parallellisme is het hiermee mogelijk op hoge snelheid digitale signaalprocessing te plegen. Bekend is de bit-slice familie van AMD: de AM2900 (later de AM29000).

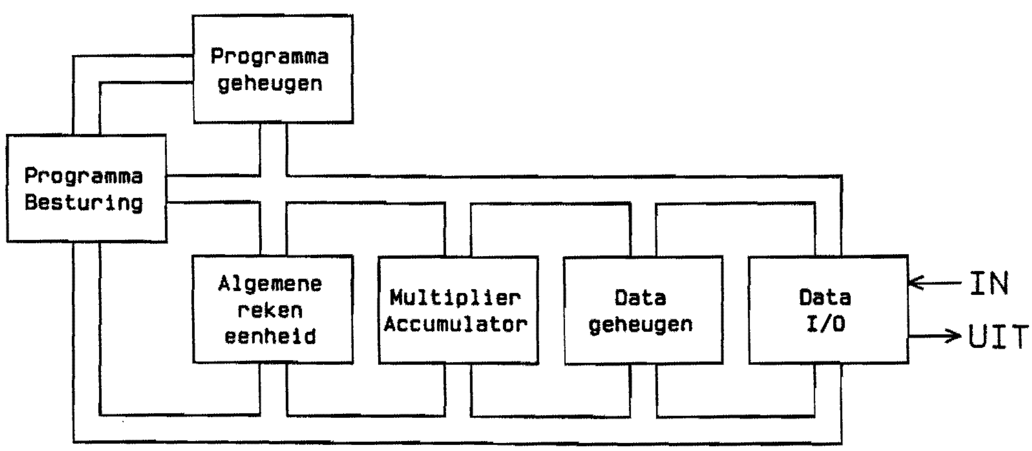
De bit-slice systemen zijn in principe rechttoe rechtaan, maar de programmering is gecompliceerd en tijdrovend. De hardware is massaal en de snelle componenten zorgen voor de nodige timingproblemen. De fabrikanten van chips hebben het merendeel van deze bezwaren weggenomen door de ontwikkeling van een single-chip processor met daarin het overgrote deel van de bit-slice logica. Dat was de geboorte van de DSP.

2. Functiebeschrijving.

2.1. Wat is een DSP ?

Een digitale signaalprocessor is een digitale chip met een microprocessor-achtige structuur, die speciaal is ontworpen voor het uitvoeren van digitale signaalbewerkingen. De nadruk ligt hierbij op het met hoge snelheid kunnen vermenigvuldigingen en accumuleren. De DSP heeft hiervoor in het algemeen een speciale structuur. Voor het op hoge snelheid kunnen uitvoeren van gecompliceerde bewerkingen is uitgegaan van een hoge mate van parallellisme. Op de chip is naast een of meerdere rekeneenheden een echte parallelle vermenigvuldiger aan boord, vaak zelfs een vermenigvuldiger/accumumulator. Instructies en datawoorden kunnen gelijktijdig via afzonderlijke wegen worden getransporteerd, waardoor de verschillende rekeneenheden optimaal kunnen worden gebruikt. Het programmageheugen wordt op verschillende manieren gerealiseerd. Naast een volledig extern geheugen zijn er uitvoeringen met ROM, met EPROM en eventueel combinaties daarvan. Data I/O kan eveneens op verschillende manieren worden gerealiseerd. Een volledig parallelle I/O poort wordt vaak gecombineerd met een of meerdere seriële poorten.

Men zou de digitale signaalprocessor in een sterk vereenvoudigd functioneel blok diagram kunnen weergeven als in figuur 1.
Door de hier genoemde en nog vele andere gebruiksmogelijkheden is de DSP een zeer complexe VLSI-chip, die alleen met behulp van de modernste technologieën kan worden geproduceerd.



Figuur 1. Functioneel blokdiagram DSP.

2.2. De functie van de DSP.

Een DSP kan vanwege zijn microprocessor-achtige structuur worden gebruikt in een systeem waar taken moeten worden uitgevoerd die ook door een microprocessor zouden kunnen worden gedaan. De kracht van de DSP is echter daarin gelegen dat hij naast deze "eenvoudige" taken ook in staat is om een aantal gecompliceerde taken op hoge snelheid uit te voeren. Het is de taak van de programmeur om de dataflow tussen het geheugen en de interne registers enerzijds en de rekenunits anderzijds zodanig in de juiste volgorde en op het juiste moment te laten verlopen dat de gewenste procedures foutloos worden uitgevoerd. Omdat het met de DSP-architectuur mogelijk is om verschillende operaties parallel uit te voeren zijn de conventionele sequentiële hogere programmeertalen zoals Fortran, Pascal en C erg inefficiënt, vooral als de verwerkingssnelheid kritisch is. De uitvoering van een specifiek algorithme hangt dan ook in belangrijke mate samen met de architectuur van de DSP en de mogelijkheden die hierdoor geschapen worden vooral voor wat betreft datatransport en geheugenadres-generatie. Een belangrijke faktor in dit geheel is ook de vereiste rekennauwkeurigheid. Men kan natuurlijk voor een bepaald systeem gebruik maken van processoren met een grotere woordlengte (bijv. 32 bits i.p.v. 16) of de algorithmes uitvoeren met dubbele precisie. Voor speciale toepassingen bestaan tegenwoordig ook DSP's die werken met floating-point waarden i.p.v. de standaard fixed-point uitvoeringen.

2.3. De architectuur van de DSP.

De opbouw van een DSP wordt natuurlijk bepaald door de eisen van hoge snelheid en grote mate van parallellisme. De meeste DSP's hebben daarom, in tegenstelling tot de bij veel microprocessoren toegepaste *Von Neumann* architectuur, een (al of niet gemodificeerde) *Harvard* architectuur. De Harvard architectuur wordt gekenmerkt door een gescheiden programmabus en databus. Hierdoor is het mogelijk om gelijktijdig een programma-instructie op te halen en een datatransport te plegen, hetgeen ook betekent dat het uitvoeren van de ene instructie gelijktijdig kan gebeuren met het ophalen van de volgende programma-instructie. Tevens kunnen vanwege de parallelprocessing een aantal verschillende units tijdens dezelfde instructiecyclus een operatie uitvoeren. Op een of andere manier beschikt de DSP altijd over een *Arithmetic Logic Unit* (ALU) voor optellen, aftrekken en logische operaties, een *Shifter* voor het schalen van van data voor of na een of andere operatie, een *Multiplier*, eventueel gekoppeld met een *Accumulator*, en tenminste één *Adresberekeneenheid.*

Vaak wordt ook *pipelining* toegepast, wat het mogelijk maakt om operaties of delen daarvan reeds te starten voordat de vorige operatie is afgerond. Bijvoorbeeld kan bij een Multiply/Accumulate instructie reeds een nieuwe multiply worden opgestart terwijl de vorige Accumulate nog bezig is. Deze specialiteit vereist van de programmeur de uiterste zorgvuldigheid bij het ontwerpen van de software.

Evenals in de conventionele microprocessoren werken de subsystemen samen onder de controle van een z.g. *Sequencer*, die voorzieningen heeft voor het testen van condities, subroutine- en interruptafhandeling, handshaking met andere devices en soms regeling van master en slave processing.

De genoemde *Adresberekeneenheid* is evenwel ook van essentieel belang. De noodzaak van herhaald vermenigvuldigen/accumuleren van data met coëfficiënten uit verschillende delen van het geheugen (RAM of ROM) maakt dat een snelle berekening van geheugenadressen onontbeerlijk is. Voor het minimaliseren van de procestijd van bijvoorbeeld een Fast Fourier Transformatie is het noodzakelijk om snel de sequentieële adressen in een circulair buffer voor input en output te bepalen en de bit-reverse adressen te genereren. In de meer geavanceerde DSP's wordt hierin voorzien door verschillende adresgeneratoren.

3. Beschrijving van een specifieke DSP.

3.1. De TMS320 familie.

Een van de belangrijkste, waarschijnlijk de belangrijkste, producent van DSP's is de firma Texas Instruments. De TMS320 familie van DSP's is in de loop van de tijd uitgegroeid tot drie generaties.

De eerste generatie bestaat heden uit een aanzienlijk aantal leden, die allen als uitgangspunt hebben de TMS32010, een microcomputer met een 32 bits interne Harvard structuur en een 16 bits externe interface die 5 MOPS aan kan.

De tweede generatie bestaat op dit moment uit de TMS32020 en de daarvan afgeleide TMS320C25. De TMS32020 heeft met een aangepaste architectuur en een uitgebreidere instructieset een throughput van 2 tot 3 maal die van de TMS32010.

De derde generatie is juist uitontwikkeld en begint met de TMS320C30 die, in tegenstelling tot de voorgaande generaties, vooral is ontworpen voor het uitvoeren van floating-point arithmetiek.

Als typisch voorbeeld van de TMS320 familie zal hier de TMS320C25 nader worden bekeken.

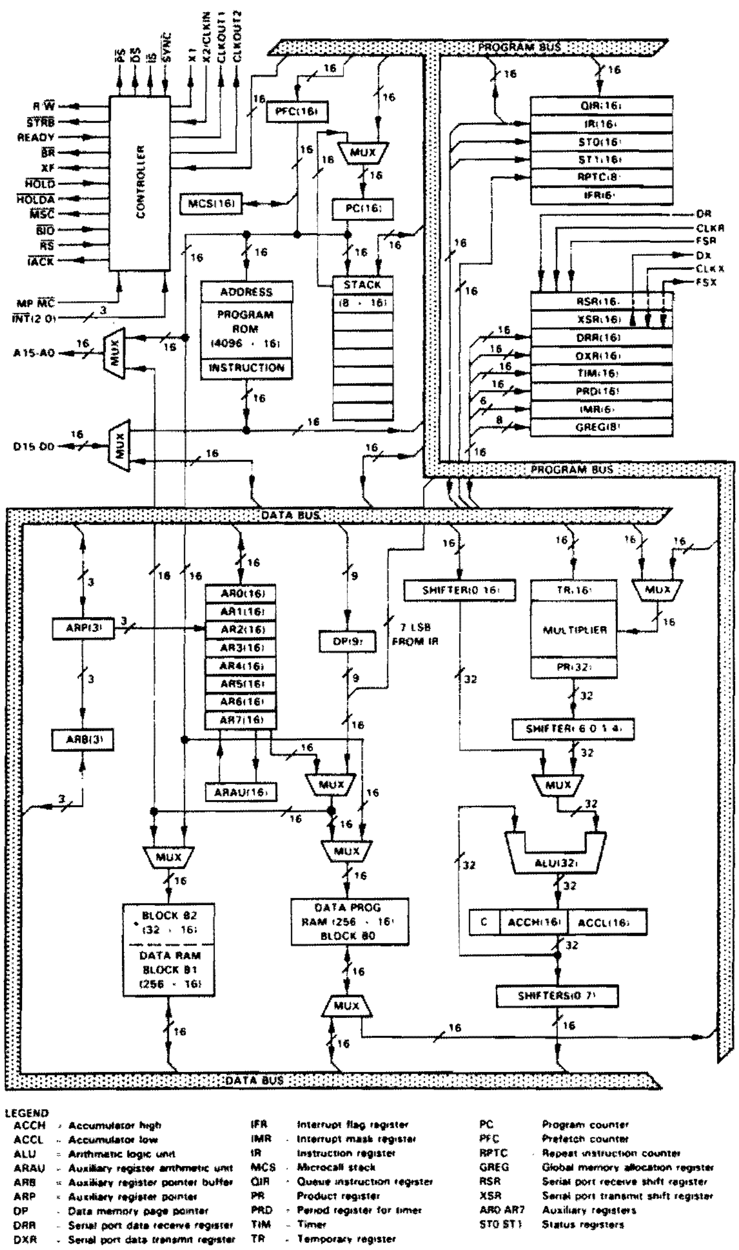
3.2. De TMS320C25.

De TMS320C25 is een in 1.8 um CMOS technologie uitgevoerde 68-pins PLCC chip met een enkele 5V voedingsspanning. De instructiecyclus bedraagt 100 nsec, waarbinnen een grote mate van parallel-operatie mogelijk is.

3.2.1. De architectuur.

De TMS320C25 is uitgerust met een *Harvard-architectuur.* Het kenmerk hiervan is de scheiding van programma- en datageheugen door een aparte programma- en databus. TI heeft echter een mogelijkheid aangebracht om ook data te kunnen uitwisselen tussen beide bussen, zodat men hier kan spreken van een gemodificeerde Harvard structuur. De scheiding van programma- en databus heeft als grote voordeel de verhoogde datatransportsnelheid; datatransport kan gelijktijdig gebeuren met een instructiefetch.

3.2.2. Het blokschema.

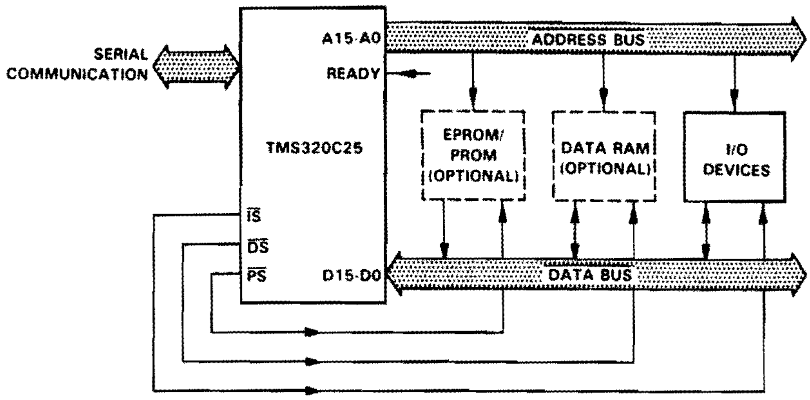


Figuur 2. Blokschema TMS320C25.

Het blokschema van de TMS320C25 is afgebeeld in figuur 2. De belangrijkste functies van de processor zijn hierin weergegeven, zoals de interne programmabus en databus en de hierop aangesloten functionele units, I/O-interfacing en externe signalen. Een beschrijving van deze units en signalen wordt gegeven in de volgende paragrafen.

3.2.3. De signalen.

De TMS320C25 heeft een 16 bits brede adresbus (A15-A0) en een 16 bits brede databus (D15-D0), die zowel voor extern programmageheugen als voor extern datageheugen als voor I/O kan worden gebruikt. Figuur 3 geeft een overzicht van een minimaal systeem.



Figuur 3. Een minimaal Processorsysteem.

De controller verzorgt een aantal signalen voor verschillende functies, zoals:

- keuzesignaal voor programma /data /I/O-geheugen.
- kloksignalen en synchronisatiesignalen.
- interrupt en handshaking.
- algemene controlesignalen.

De DSP levert verder een aantal signalen die kunnen worden gebruikt voor seriële I/O, zoals input, output, klok en sync.

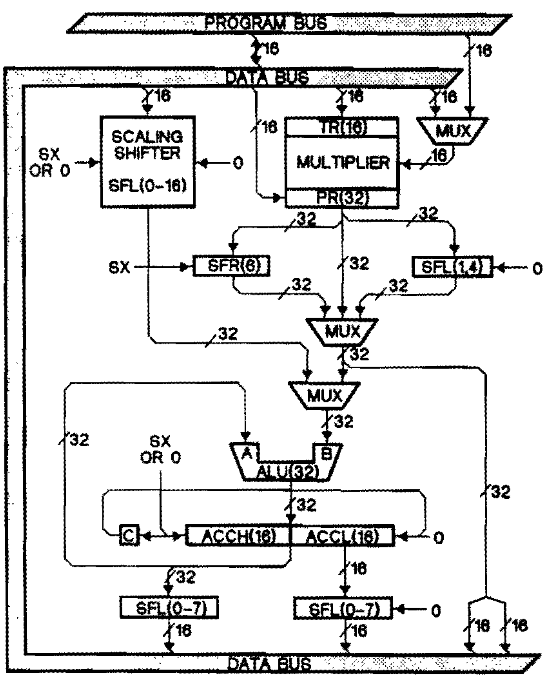
3.2.4. Het geheugen.

Het extern adresseerbare geheugen van de TMS320C25 beslaat in totaal 128k woorden, te verdelen in 64k programmawoorden, 64k datawoorden en 16 I/O poorten. Alle woorden, zowel instructies als data als I/O, bestaan uit 16 bits.

Intern heeft de processor de beschikking over 4k programma (ROM) en 544 datawoorden (RAM). Het on-chip ROM kan al of niet worden geactiveerd. Het interne RAM is verdeeld in 3 blokken, waarbij 1 blok van 256 woorden of als programmageheugen of als datageheugen kan worden geconfigureerd. Deze specialiteit biedt verschillende toepassingsmogelijkheden, zoals o.a. het downloaden van een programma uit traag (goedkoop) extern geheugen naar snel intern programmageheugen om vervolgens van hieruit tijdkritische procedures te kunnen uitvoeren.

De processor is in staat om met traag extern geheugen te communiceren via het inlassen van een of meer wait-states en heeft verder de mogelijkheid van een DMA mode voor bijv. multiprocessing.

3.2.5. De CALU.



Figuur 4. Central Arithmetic Logic Unit.

De TMS320C25 bevat een CALU, *Central Aritmetic and Logic Unit*, die bestaat uit een 16 bits shift unit, een 16 * 16 bits parallel multiplier, een 32 bits ALU, een 32 bits accumulator en scaling units na de multiplier en de accumulator (zie figuur 4).

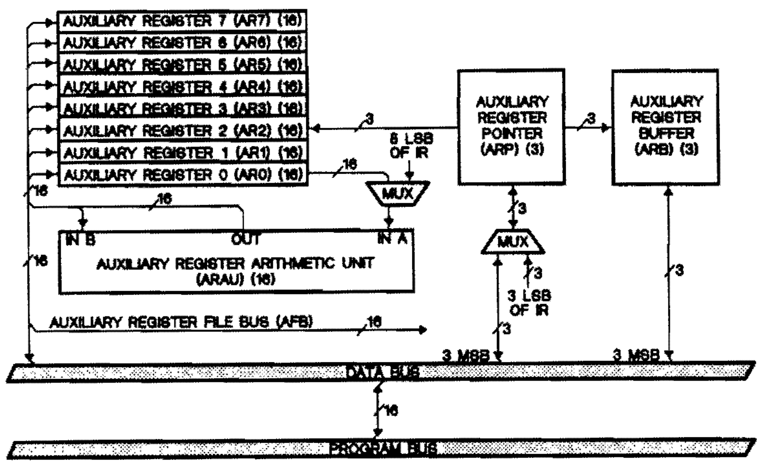
Een typische ALU instructie bestaat uit de volgende stappen:
1. De data wordt van de databus gehaald.
2. De data wordt via de shift unit naar de ALU getransporteerd.
3. In de ALU wordt de bewerking uitgevoerd.
4. Het resultaat wordt naar de accumulator doorgeleid.

Een van de ingangssignalen van de ALU is altijd de accumulator terwijl het tweede kan komen van de databus via de shift unit of van het produktregister van de multiplier. De multiplier krijgt op zijn beurt zijn operanden van de databus of van de programmabus.

Alle operaties worden in het algemeen uitgevoerd in 2-complement code, waarvoor een aantal extra voorzieningen zijn aangebracht voor manipulaties met het tekenbit en met de overflow.

Voor de overgang van de 32 bits ALU naar de 16 bits databus is het van groot belang de signalen goed te schalen en/of te kwantiseren.

3.2.6. De ARAU.

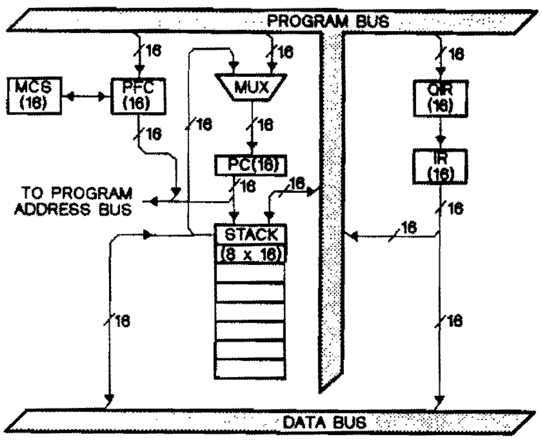


Figuur 5. Auxiliary Registers.

De TMS320C25 is voorzien van een *Auxiliary Register Arihmetic Unit* (zie figuur 5). Deze rekeneenheid bedient de 8 Auxiliary Registers die kunnen worden gebruikt voor indirecte adressering van het datageheugen. De ARAU is bedoeld voor de verwerking van positieve getallen, zoals incrementeren, optellen, aftrekken en bit-reverse operaties. Daar deze operaties op de Auxiliary Registers niet door de CALU hoeven worden uitgevoerd kan deze voor andere bewerkingen worden ingezet.

3.2.7. De Program Sequencer.

De *Program Sequencer* verzorgt het verloop van het programma en is in feite onzichtbaar voor de gebruiker. Het Program Counter Register bepaalt het adres van de uit te voeren instructie en wordt automatisch geïncrementeerd of aangepast door een al of niet conditionele branch instructie. Voor subroutines en interrupts is voorzien in een 8 woorden diepe stack, die ook voor dataopslag kan worden benut. Een Program Repeat Counter biedt de mogelijkheid om een instructie tot 256 keer te laten herhalen (zie figuur 6).



Figuur 6. Program Counter met omgeving.

3.2.8. De I/O interface.

De processor kan met andere processoren of met signaalomzetters (bijv. A/D en D/A) communiceren via een seriële of via een parallelle interface. De seriële interface werkt op interruptbasis als een zender of als ontvanger. Het is mogelijk om te werken met verschillende formaten en in verschillende modes (zoals al of niet continu, wel of niet synchroon) met een maximale overdrachtsnelheid van 5 Mbit/sec. Via de parallelle interface kan men 16 input- en 16 outputpoorten aanspreken. Elke poort wordt aangestuurd door de volledige 16 bits databus.

3.2.9. De instructieset.

De TMS320C25 beschikt over in totaal 133 instructies die het mogelijk maken om rekenintensieve signaalbewerkingsoperaties en "general purpose" operaties door elkaar heen te gebruiken. De instructies worden gegeven in mnemonics, maximaal 4 letterige afkortingen van de instructies die door een assembler worden omgezet in de instructiecode van de processor.

De instructies bestaan voor het grootste deel uit enkelwoord instructies die in één klokcyclus worden uitgevoerd. Instructies die een 16 bits waarde of adres in zich hebben zijn tweewoord instructies en beslaan meer dan een klokcyclus. Doordat de executietijd van elke instructie is gegeven kan hieruit ook de totale benodigde verwerkingstijd van een algorithme worden bepaald.

De instructies maken gebruik van 3 adresseermogelijkheden:
- Directe adressering: de instructie bevat tevens het data-adres.
- Indirecte adressering: het data-adres wordt gegeven door een register.
- Impliciete adressering: de instructie bevat zelf de data.

De instructies zijn in de databoeken uitgebreid beschreven. Enkele, zeer willekeurige, voorbeelden van instructies zijn:

ADD	<dma>	data van adres <dma> wordt opgeteld bij de accumulator
ABS		neem de absolute waarde van de accumulator
MPYK	<const>	vermenigvuldig het T-register met een constante
MAR	<dma>	modificeer aux. register met waarde uit <dma>
BGEZ	<pma>	branch naar <pma> indien accu >= 0
IN	<dma>,<pa>	input data van poort <pa> naar <dma>
RXF		reset externe flag bit.

4. Realisering van een DSP-systeem.

Voor het ontwerpen van een systeem voor tijddiscrete signaalbewerking is het uitgangspunt de theoretische beschrijving ervan. Als men ziet hoeveel problemen een theoreticus soms heeft met de praktische oplossing van een probleem en een practicus met de theorie zal het geen overbodige luxe zijn om hier een uiteenzetting te geven van van de toepassing van deze theorie in de praktijk. Er ligt natuurlijk een wereld van verschil tussen de beschrijving van een overdrachtsysteem en de toepassing hiervan in bijv. een computergestuurde draaibank of een copiëermachine. Ik zal hier proberen te laten zien hoe men uitgaande van de systeemfunctie het systeem kan realiseren met behulp van een Digitale Signaalprocessor.

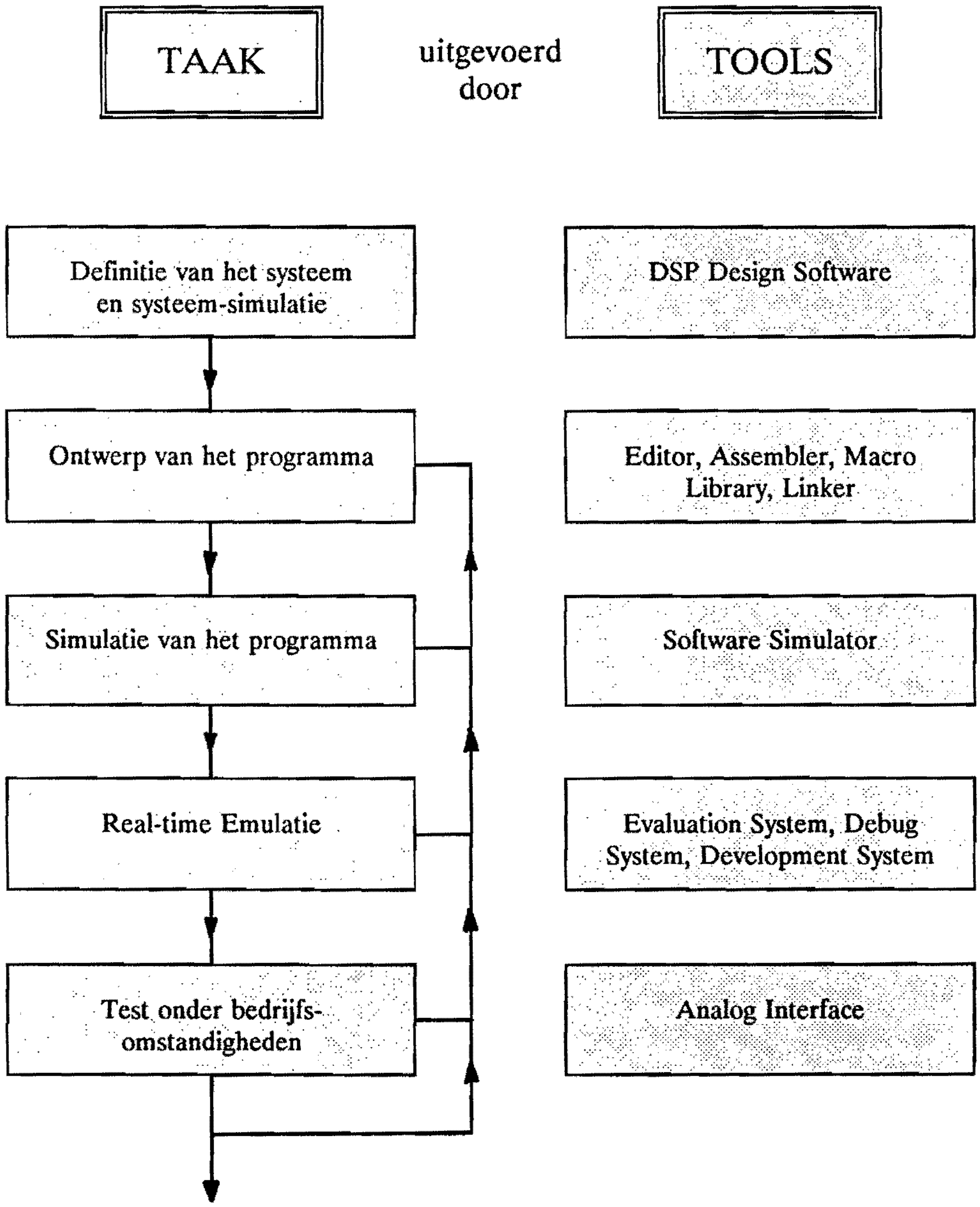
4.1. De probleemstelling en de aanpak van het probleem.

Voor het realiseren van een bepaald signaalbewerkingssysteem zal de ontwerper zich dienen af te vragen of dit mogelijk is door gebruik te maken van DSP's. Zo is de snelheid waarmee de operaties moeten worden uitgevoerd een van de belangrijkste overwegingen of een DSP in aanmerking kan komen. Tevens zal de hoeveelheid op te bergen data een onomkoombaar gegeven zijn. Als er een bepaalde DSP bestaat die de gestelde problemen kan oplossen en die ook om een aantal andere redenen (prijs, verkrijgbaarheid) aanvaardbaar is, zal de ontwerper de kloof moeten overbruggen tussen de probleemstelling en de praktische oplossing. Een aantal fabrikanten van DSP's heeft ten behoeve van deze ontwerpers de brug gebouwd om de betreffende kloof op een eenvoudige wijze te nemen. Het totaal van deze ontwerpgereedschappen biedt de gebruiker de gelegenheid om met betrekkelijk eenvoudige middelen zijn doel te bereiken.

4.2. Overzicht van taken en tools.

De krachtigste architectuur en de flexibelste instructieset hebben voor de gebruiker geen enkele zin als hem niet de middelen ten dienste staan om de processor op eenvoudige wijze te programmeren. Deze ontwikkelhulpmiddelen spelen daarom in het concept van een DSP-familie een belangrijke rol. Dit is de reden waarom er hier dan ook een plekje voor wordt ingeruimd.

Figuur 7 geeft een algemeen overzicht van de uit te voeren taken met de bijbehorende ontwikkeltools.



Figuur 7. Overzicht van taken en tools.

4.3. Het uitvoeren van de ontwerptaak.

De ontwerper zal achtereenvolgens de volgende taken moeten uitvoeren.

- **Bepaling van de systeemfunctie.**

  Als de ontwerper zijn systeem exact heeft gedefinieerd kan hij de systeemfunctie gaan berekenen. In veel gevallen zal hij zich daarbij kunnen bedienen van tabellenboeken. Tegenwoordig is hiervoor de nodige design software beschikbaar. Design packages zoals bijv. DFDP (Digital Filter Design Package) van ASPI of ILS (Interactive Laboratory System) van STI bieden de mogelijkheid om op een zeer handzame manier digitale filters te ontwerpen. De resulterende filters leveren een systeemfunctie op met de bijbehorende, al of niet gekwantiseerde, coëfficiënten. De designsoftware kan tevens van de ontworpen systemen de karakteristieken, zoals de frekwentie-karakteristiek, de fasekarakteristiek, de groepslooptijd, de impulsresponsie en het pool-nulpunten diagram in reproduceerbare vorm genereren. Als het systeem dan aan de gestelde eisen voldoet kan worden begonnen met het programmeren van de DSP.

- **Het ontwerp van een programma voor de DSP.**

  De ontwerper zal uit de databoeken van de DSP zich de instructieset eigen moeten maken. In de databoeken van de DSP vindt de ontwerper de DSP-instructieset in binaire code en in assembler-mnemonics. Het programma wordt natuurlijk geschreven in mnemonics die aan een assembler kunnen worden doorgespeeld. Het is met ongeveer elke tekstverwerker mogelijk om een tekstfile te produceren die als sourcefile dient voor een assembler. De assembler produceert, uitgaande van de sourcefile, de processor-codefile. Ook kan de assembler vaak een listfile leveren met een listing van de source en de bijbehorende codes en eventuele boodschappen en/of waarschuwingen. Voor samenwerking met een simulator kan een assembler soms ook hiervoor de files aanmaken. Ter vereenvoudiging van de programmeertaak van de ontwerper bestaan er reeds voorgeprogrammeerde deeltaken in de vorm van macro-instructies in een macro-library, die eenvoudig in het programma kunnen worden aangeroepen. Indien men met erg uitgebreide programmatuur te maken heeft kan het van voordeel zijn om een programma op te bouwen uit verschillende units die afzonderlijk kunnen worden geassembleerd en met een Linker programma kunnen worden samengevoegd tot een uitvoerbaar programma.

- **Simulatie van het programma.**

  In veel gevallen worden software en hardware afzonderlijk ontwikkeld, en worden pas samengevoegd als beide naar behoren functioneren. Dit geeft de ontwerper de zekerheid dat problemen die ontstaan na de koppeling van software en hardware niet hun oorzaak vinden in een foutief programma. Om te verifiëren of het programma de bedoelde operaties ook daadwerkelijk uitvoert kan men gebruik maken van een Software Simulator. Hiermee kan men op het scherm van zijn computer de registers van de processor en de inhoud van de geheugens tonen tijdens de uitvoering van het programma. Dit programma kan daartoe stap voor stap worden doorlopen, of tot vooraf ingestelde breakpoints. In het algemeen is simulatie van I/O ook mogelijk, door het aangeven van een datafile als ingangswaarden voor een inputpoort en het gebruik van een tweede datafile die de door een outputpoort uitgezonden waarden opslaat. Het doen en laten van het programma kan hiermee dus uitvoerig worden nagetrokken, hoewel dit natuurlijk niet real-time kan gebeuren.

- **Real-time emulatie.**

  Wil men de programma's real-time testen, d.w.z. met de kloksnelheid van de processor zelf, dan kan dat gebeuren met behulp van speciale ontwikkelsystemen. Deze worden vaak aangeduid met namen als Evaluation System, Development System of Debug System. Een dergelijk ontwikkelsysteem is een apparaat dat door een computer wordt gestuurd (of is ingebouwd) en dat in feite een DSP kan nabootsen. Intern bevat het apparaat natuurlijk een universele versie van de betreffende processor en extern reageert het exact als een DSP. Hiermee kan op verschillende manieren worden gewerkt.

  1. Met enkel het ontwikkelsysteem kan men de DSP programmeren en testen door stap voor stap of op volle snelheid tot aan een break-point door het programma te lopen zonder de I/O erin te betrekken, dus enigszins als een snelle versie van een simulator.

2. Het ontwikkelsysteem wordt gekoppeld met een analoge interface, waardoor de processor via een universele analoge I/O kan communiceren met de buitenwereld.

3. Het ontwikkelsysteem wordt gekoppeld met een prototype van het uiteindelijke systeem, dat als targetsysteem fungeert, waarin de geëmuleerde DSP de plaats inneemt van de oorspronkelijke processor.

In de praktijk worden de voorgaande mogelijkheden stap voor stap doorlopen. Als in stap 1 blijkt dat het programma goed functioneert kunnen in stap 2 de I/O-commando's voor de universele analoge interface worden toegevoegd en getest.

- **Testen onder bedrijfsomstandigheden.**

  De bedrijfsomstandigheden kunnen in de meeste gevallen worden nagebootst door stap 2. Veel fabrikanten kunnen de daarvoor benodigde analoge interface leveren. In stap 3 wordt het ontwikkelsysteem gekoppeld met een prototype van het uiteindelijke systeem, dat dan het target systeem is. Als hierbij de I/O in het programma wordt aangepast aan die van het targetboard zal dit in het algemeen de laatste stap zijn die leidt tot een goed functionerend digitaal signaalbewerkingssysteem.

5. Het Filterprogramma.

Bij de functieomschrijving van de DSP is aangegeven hoezeer de functie is toegesneden op het uitvoeren van digitale signaalbewerkingen. Zowel de afzonderlijke units als de architectuur van de DSP stellen ons in staat om op efficiënte wijze digitale signalen te bewerken. De beschrijving van de TMS320C25 geeft aan dat deze processor de vereiste eigenschappen in ruime mate bezit. Deze processor is dus uitermate geschikt om digitale signaalbewerkingen uit te voeren.
Een groot gebied van toepassingen is gebaseerd op de relatie tussen het ingangssignaal $x[n]$ en het uitgangssignaal $y[n]$ volgens:

$$y[n]=\sum_{k=1}^{N} a_k y[n-k]+\sum_{k=0}^{M} b_k x[n-k] \tag{1}$$

Dit is een lineaire differentievergelijking met constante coëfficiënten, waardoor twee klassen van filters kunnen worden gerepresenteerd:

1. Finite Impulse Response (FIR) filters en
2. Infinite Impulse Response (IIR) filters.

De volgende secties zullen achtereenvolgens de implementatie van beide filters op de TMS320C25 beschrijven.

5.1. FIR-filters.

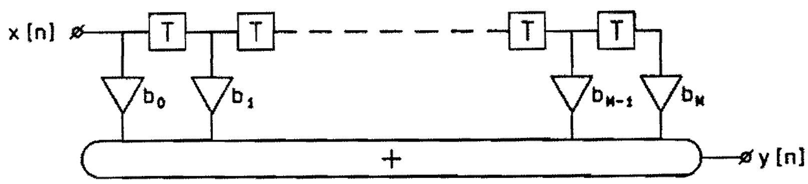
Bij FIR-filters zijn alle coëfficiënten a_k van vergelijking (1) nul. Daardoor wordt vergelijking (1) gereduceerd tot:

$$y[n]=\sum_{k=0}^{M} b_k x[n-k] \tag{2}$$

waarin $M+1$ de lengte van het filter voorstelt.
Het uitgangssignaal van een FIR-filter is dus simpelweg een eindige gewogen som van de huidige en de vorige ingangswaarden van het filter. In formule (2) is direct te zien dat de waarden b_k de waarden van de impulsresponsie voorstellen. Bij de implementatie zal verder worden uitgegaan van de differentievergelijking in plaats van de systeemfunctie. De differentievergelijking kan namelijk rechttoe rechtaan worden vertaald naar een efficiënte implementatie op een DSP.

De volgende 3 uitgangspunten vormen de basis voor het programma:

1. De relatie tussen de systeembeschrijving met met zijn impulsresponsie en de structuur van het filter.
2. De mogelijkheden van de benodigde instructies.
3. De mannier waarop de coëfficiënten en vooral de signaalwaarden worden opgeslagen in het geheugen van de processor.



Figuur 8. FIR-filter structuur.

Figuur 8 geeft de structuur van het FIR-filter, waarin de coëfficiënten $b_0 ... b_M$ de waarden van de impulsresponsie voorstellen.

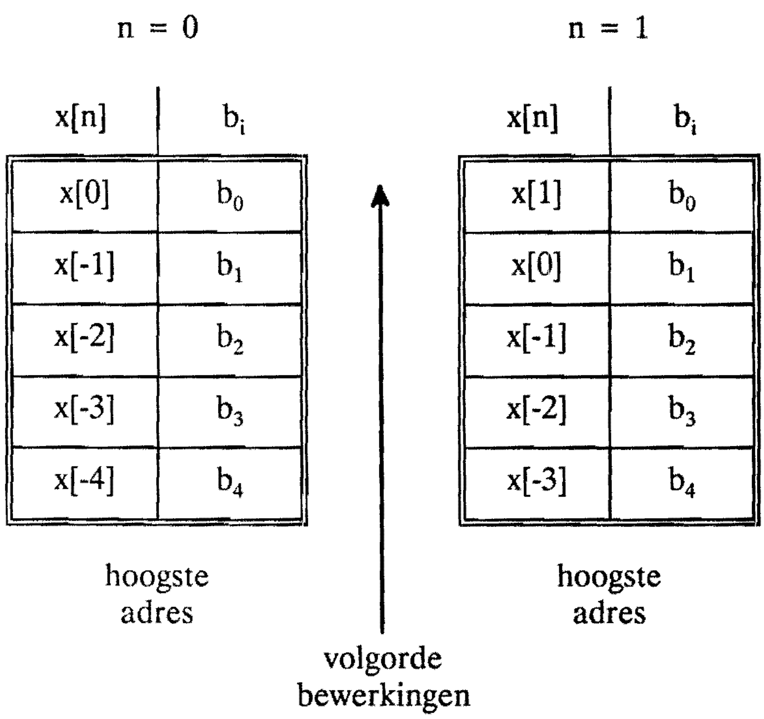
Drie operaties vormen hierin de basis van de bewerking:

- Vermenigvuldigen van het ingangssignaal met de coëfficiënten.
- Optellen van alle afzonderlijke produkten.
- Het doorschuiven van de ingangswaarden.

Hierbij worden de volgende instructies gebruikt:

LT	<dma>	Laad een operand (reg. T) van de vermenigvuldiger met een waarde uit het datamemory (met adres <dma>)
MPY	<dma>	Vermenigvuldigvuldig de waarde uit reg. T met een waarde uit het datamemory en plaats het produkt in register P
APAC		Tel de inhoud van reg. P op bij de Accumulator
DMOV	<dma>	De inhoud van datamemory <dma> wordt gecopiëerd naar <dma+1>

Door de manier waarop data kan worden doorgeschoven van het ene dataregister naar het andere is het van het grootste belang dat de ingangswaarden op de juiste manier in het geheugen worden opgeborgen en dat de bewerkingen in een bepaalde volgorde worden afgewerkt. Figuur 9 geeft deze ordening aan voor 2 opeenvolgende tijdstippen en toont de bewerkingsvolgorde.



Figuur 9. Inhoud van de dataregisters.

De wijze van programmeren kan worden onderscheiden in 3 niveaus, oplopend in complexiteit en verwerkingssnelheid. Het programma komt er dan, voor bijvoorbeeld een 5^e orde filter, in elementaire vorm uit te zien als in programma 1.

```
VLGX    IN      XN,PA2      Haal x[n] op via poortadres 2
        ZAC                 Maak de Accu nul
        LT      XNM4        Laad T met x[n-4]
        MPY     B4          Vermenigvuldig T met b4
        APAC                Tel P op bij Accu
        LT      XNM3
        MPY     B3
        APAC
        DMOV    XNM3        Verplaats x[n-3] naar x[n-4]
        LT      XNM2
        MPY     B2
        APAC
        DMOV    XNM2        Verplaats x[n-2] naar x[n-3]
        LT      XNM1
        MPY     B1
        APAC
        DMOV    XNM1        Verplaats x[n-1] naar x[n-2]
        LT      XN
        MPY     B0
        APAC
        DMOV    XN          Verplaats x[n] naar x[n-1]
        SACH    YN          Bewaar y[n]
        OUT     YN,PA3      Stuur y[n] uit via poortadres 3
        B       VLGX        Branch naar volgende sample
```

Programma 1. 5^e orde FIR-filter, niveau 1.

Afhankelijk van de systeemeisen kan de programmeur kiezen voor een reductie in de omvang van het programma door gebruik te maken van de indirecte adresseermode. Door gebruikmaking van een van de adresregisters met zijn autoincrement of autodecrement mode kan het FIR-filterprogramma worden herschreven in lusvorm. De programmalengte wordt hierdoor kleiner, doch de verwerkingstijd neemt toe vanwege de extra branch instructie. Dit moet door de ontwerper goed worden beseft en hij zal daarom een goede afweging moeten maken. Programma 2 geeft de lusvorm van programma 1.

Het is mogelijk om op een enigszins hoger niveau te programmeren omdat men vanwege de speciale architectuur van de TMS320 familie een aantal operaties heeft kunnen combineren tot één instructie. De instructies *APAC*, *LT* en *DMOV* heeft men gecombineerd tot de instructie *LTD* !

```
        LARP    AR0         Laad adresregister pointer met 0
VLGX    IN      XN,PA2      Haal x[n] op via poortadres 2
        LARK    AR0,XNM4    Adresregister 0 wijst naar x[n-4]
        LARK    AR1,B4      Adresregister 1 wijst naar b_4
        ZAC                 Maak de Accu nul
        LT      *-,AR1      Laad T met x[n-4], decrement AR0
                            en zet pointer op 1
        MPY     *-,AR0      Vermenigvuldig T met b_4, decrement
                            AR1 en zet pointer op 0
        APAC                Tel P op bij Accu
LUS     LT      *,AR1       Laad T met x[n-i] en zet pointer
                            op 1
        MPY     *-,AR0      Vermenigvuldig T met b_i, decrement
                            AR1 en zet pointer op 0
        APAC                Tel P op bij Accu
        DMOV    *-          Verplaats x[n-i] naar x[n-i-1] en
                            decrement AR0
        BANZ    LUS         Branch naar LUS zolang AR0 ≠ 0
        SACH    YN          Bewaar y[n]
        OUT     YN,PA3      Stuur y[n] uit via poortadres 3
        B       VLGX        Branch naar volgende sample
```

Programma 2. 5^e orde FIR-filter, in lusvorm.

De basisoperatie per FIR-filtertap kan dan worden teruggebracht tot 2 instructies: *LTD* en *MPY* ; dit leidt samen met de lusvorm tot zeer compacte programmas.
De elementaire operatie per filtertap wordt dan gelijk aan de vorm zoals die is weergegeven in programma 3.

```
        |       |
        |       |
        LTD     *-,AR1      Tel P op bij de Accu, laad T met
                            x[n-i], verplaats x[n-i] naar x[n-i-1],
                            decrement adresregister 0 en zet
                            adrespointer op 1
        MPY     *-,AR0      Vermenigvuldig T met b_i, decrement
                            adresregister 1 en zet adrespointer
                            op 0
        |       |
        |       |
```

Programma 3. FIR-filtertap, niveau 2.

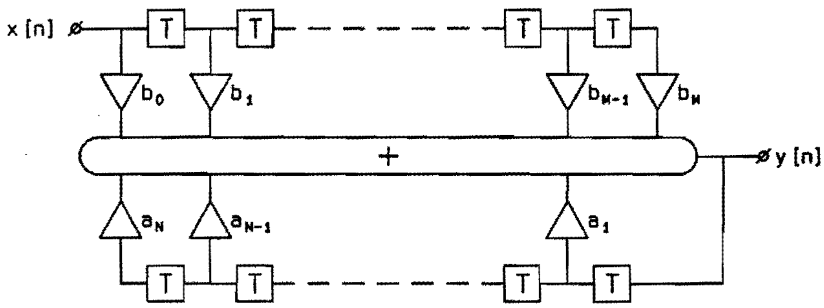
Afhankelijk van de toepassing kan dit instructiepaar of in lijn of in lusvorm worden gebruikt, terwijl hierbij een vorm van indirecte adressering van voordeel kan zijn.
Het instructiepaar LTD/MPYK kan worden gebruikt als de coëfficiënten impliciet in de MPYK instructie worden aangeleverd. De maximale coëfficiënt-woordlengte die hierbij kan worden gebruikt, slechts 13 bits, kan voor diverse toepassingen voldoende zijn. Twee voordelen van de MPYK instructie zijn de snellere verwerking en de beperking van het datageheugen van 2 plaatsen per tap naar 1.
In de tweede generatie DSP's van Texas Instruments, waartoe ook de TMS320C25 behoort, is de mogelijkheid geopend om een single-cycle multiply/accumulate instructie toe te voegen. MPYA is de multiply/accumulate en MPYS is de multiply/subtract instructie. Zelfs de combinatie van multiply/accumulate/datamove is mogelijk geworden in de vorm van de MACD instructie. De MACD instructie gebruikt als operanden een waarde uit het datageheugen en een uit het programmageheugen. Door een deel van het datageheugen te configureren als programmageheugen kan dus een van de operanden via de databus en de andere gelijktijdig via de programmabus worden aangevoerd. MACD bevat dus de complete operatie van één filtertap en biedt dus de mogelijkheid om zeer snel een tap te verwerken.
Combineren van de MACD instructie met de repeat mogelijkheid van de DSP, RPTK n, levert, vanwege het feit dat de pipeline gevuld blijft en de instructie-decodering slechts eenmalig is, het meest compacte en tevens het snelst mogelijke filterprogramma. Programma 4 laat zien hoe deze MACD instructie wordt toegepast als derde niveau in de filterprogrammas.

```
         CNFP               Configureer een blok datageheugen als
                            programmageheugen.
VLGX     IN      XN,PA2     Haal x[n] op via poortadres 2
         LRLK    AR1,DB     Adresregister 1 wijst naar datablok
         LARP    AR1        Laad adresregisterpointer met 1
         MPYK    0          Maak register P nul
         ZAC                Maak de Accu nul
         RPTK    NM1        Herhaal n-1 keer
         MACD    PB,*-      Multiply/accumulate programmablok en
                            datablok
         APAC               Extra optelling van P bij Accu
         SACH    YN,1       Bewaar y[n]
         OUT     YN,PA3     Stuur y[n] uit via poortadres 3
         B       VLGX       Branch naar volgende sample
```

Programma 4. FIR-filter, niveau 3.

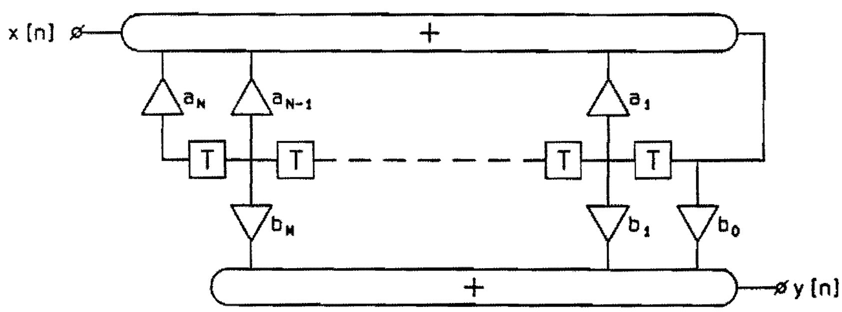
5.2. IIR-filters.

Het concept dat bij de FIR-filters geïntroduceerd is, kan worden doorgetrokken naar de IIR-filters. Bij de IIR-filters is echter minstens één van de a_k waarden in (1) ongelijk aan nul, waardoor de differentievergelijking van het FIR-filter gelijk is aan vergelijking (1).



Figuur 10. Directe-vorm.

Deze kan gerealiseerd worden met de netwerkstructuur zoals die is aangegeven in figuur 10. Deze structuur wordt de directe-vorm genoemd omdat hij rechtstreeks is af te leiden uit de differentievergelijking; de coëfficiënten in het netwerk zijn gelijk aan die in de differentievergelijking.

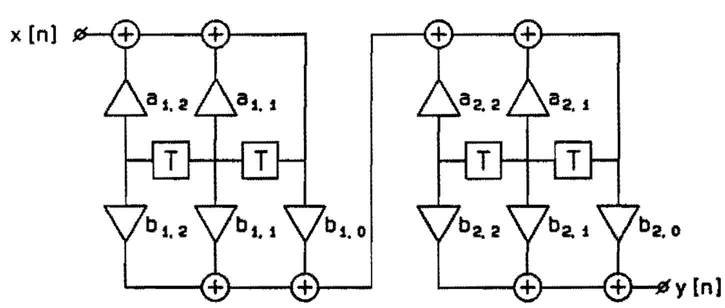


Figuur 11. Directe-vorm 2.

Figuur 11 geeft een netwerkstructuur die equivalent is aan die uit figuur 10. Deze structuur wordt de directe-vorm 2 genoemd en is ontstaan uit de directe-vorm door verwisseling van het recursieve en het niet-recursieve deel en na combineren van de vertragers. Aangezien de directe-vorm 2 een minimaal aantal vertragers heeft, vraagt deze structuur een minimum aan processor-registers. Het voordeel van deze structuur is dus een minimale hoeveelheid aan benodigd datamemory voor de implementatie van IIR-filters. De filtercoëfficiënten zijn hierbij net als bij de directe-vorm gelijk aan die uit de differentievergelijking.
De directe-vorm 2 is dus efficiënter wat betreft het aantal vertragers, maar de differentievergelijking wordt hiermee niet meer direct herkenbaar geïmplementeerd. Dit is echter weer wel het geval met de getransponeerde structuur van de directe-vorm 2, die de directe-vorm 1 wordt genoemd. De directe-vorm 1 komt in het volgende hoofdstuk nog aan de orde (Figuur 13).
Een derde structuur voor IIR-filters is de cascade-structuur. Hierbij wordt het IIR-filter geïmplementeerd als een cascade-schakeling van secties van tweede orde (biquads). Door het schrijven van de systeemfunctie in de vorm :

$$H(z)=\prod_{i=1}^{N}\frac{b_{i,0}+b_{i,1}z^{-1}+b_{i,2}z^{-2}}{1-a_{i,1}z^{-1}-a_{i,2}z^{-2}} \tag{3}$$

ziet men dat dit te realiseren is door een cascade te vormen van N tweede-orde secties, die elk voor zich is opgebouwd uit een directe-vorm 2 sectie. Figuur 12 geeft de structuur van een vierde-orde IIR-filter, opgebouwd uit een cascade van twee tweede-orde secties. Bedenk echter wel dat de coëfficiënten van de cascade-structuur *niet* gelijk zijn aan die van de totale differentievergelijking.



Figuur 12. Cascade-schakeling van een vierde orde IIR-filter.

De uitgangspunten voor het programma zijn dezelfde als die bij de IIR-filters, evenals de gebruikte instructies. Dit leidt tot de implementatie van een directe-vorm 2 IIR-filter, zoals die als voorbeeld in programma 5 is beschreven voor een eenvoudige tweede orde sectie.

```
VLGX    IN      XN,PA2      Haal x[n] op via poortadres 2
        LAC     XN,15       Laad de Accu met x[n]
        LT      DNM1        Laad T met d[n-1]
        MPY     A1          Vermenigvuldig T met a1
        LTA     DNM2        Tel P op bij de Accu en laad T met
                            d[n-2]
        MPY     A2          Vermenigvuldig T met a2
        APAC                Tel P op bij de Accu
        SACH    DN,1        Bewaar d[n]
        ZAC                 Maak de Accu nul
        MPY     B2          Vermenigvuldig T met b2
        LTD     DNM1        Tel P op bij de Accu, laad T met
                            d[n-1] en verplaats d[n-1] naar d[n-2]
        MPY     B1          Vermenigvuldig T met b1
        LTD     DN          Tel P op bij de Accu, laad T met
                            d[n] en verplaats d[n] naar d[n-1]
        MPY     B0          Vermenigvuldig T met b0
        APAC                Tel P op bij de Accu
        SACH    YN,1        Bewaar y[n]
        OUT     YN,PA3      Stuur y[n] uit via poortadres 3
        B       VLGX        Branch naar volgende sample
```

Programma 5. 2^e orde IIR-filter, directe-vorm 2.

5.3. FIR-IIR-vergelijking.

Op de vraag of men een FIR-filter of een IIR-filter moet implementeren is geen algemeen antwoord te geven. Verschillende overwegingen spelen hierbij een rol, zoals bijv. fase-eigenschappen of stabiliteitseisen.
Factoren die de programmeur in zijn beschouwing moet meenemen zijn bijvoorbeeld de woordlengte van de signalen en de coëfficiënten of de effecten van kwantisatie op bepaalde structuren. Deze overigens zeer belangrijke overwegingen vallen echter buiten het bestek van dit onderwerp en komen hier dus niet aan de orde.

Vervolgens een vergelijking van enkele karakteristieken van de implementatie van verschillende structuren op de TMS320C25.
De directe-vorm 2 en de cascadestructuur zijn gegeven als een 6^e orde structuur omdat die als voorbeeld in het volgende hoofdstuk voorkomen.
Bij het FIR-filter is uitgegaan van een lengte van 100 omdat deze lengte nodig zou zijn als men het betreffende voorbeeld met behulp van FIR-filters (volgens Parks-McClellan) zou realiseren.

In de volgende tabel is te zien dat de realisering van digitale filters met behulp van DSP's een oplossing is die reeds voor tamelijk gecompliceerde filters bij hogere sample-frekwenties zeer goede mogelijkheden biedt.

Structuur	Cycles	Executie-tijd (μsec)	Programma geheugen (woorden)	Data geheugen (woorden)
FIR-filter lengte 100	110	11,0	11	201
Directe-vorm 2 6^e orde	34	3,4	34	21
Cascade 3 * 2^e orde	40	4,0	40	26

6. Voorbeeld van een ontwerp.

De meest instructieve wijze waarop duidelijk kan worden gemaakt hoe men moet omgaan met de ontwikkeltools is de beschrijving van het ontwikkelen van een eenvoudig systeem en een demonstratie daarvan.

Als te ontwerpen systeem is gekozen voor een digitaal filterontwerp in de audioband, omdat hierin duidelijk een aantal specifieke hardware- en softwaretechnische aspecten aan de orde komt. Het ontwerp gaat uit van de TMS320C25 en wordt uitgevoerd met een Digital Filter Design software Package (DFDP) en een Software Development System (SWDS).

6.1. Bepaling van de filtereigenschappen.

Als voorbeeldfilter is hier gekozen voor een banddoorlaatfilter met de volgende eigenschappen :

bemonsterfrekwentie is 20 kHz
doorlaatband van 1,0 - 1,5 kHz
rimpel in de doorlaatband is 0,4 dB
demping in de sperband is 40 dB

Met behulp van het software-pakket "Digital Filter Design Package" van ASPI is een digitaal filter ontworpen dat voldoet aan de voorgaande eisen. Er is hierbij uitgegaan van realisatie met een IIR-filter. Een elliptisch filter bleek het meest efficient; hiermee kon aan de specificaties worden voldaan door een 6^e orde filter.

De systeemfunctie krijgt dan de vorm :

$$H(z)=\prod_{i=1}^{3}\frac{b_{i,0}+b_{i,1}z^{-1}+b_{i,2}z^{-2}}{1-a_{i,1}z^{-1}-a_{i,2}z^{-2}}$$

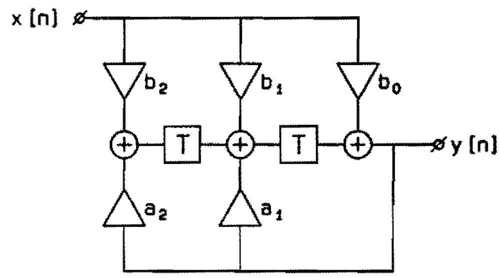
met als bijbehorende coefficienten :

i	$a_{i,1}$	$a_{i,2}$	$b_{i,0}$	$b_{i,1}$	$b_{i,2}$
1	-1,776184	0,916656	0,041883	0,000000	-0,041883
2	-1,743652	0,955933	0,200073	-0,314072	0,200073
3	-1,872742	0,969666	0,642090	-1,253723	0,642090

6.2. Bepaling van de filterstructuur.

Het filter zal worden gerealiseerd als een cascadeschakeling van 3 recursieve tweede orde secties. De structuur van het IIR-filter zal hier worden bepaald door het gebruikte ontwerpprogramma. Dit ontwerpprogramma gaat hierbij uit van een structuur die per tweede orde sectie een directe-vorm 1 is, die, vanwege specifieke eigenschappen van de toegepaste digitale signaalprocessor, zeer efficiënt is te programmeren.

Een afzonderlijke sectie van deze vorm is getekend in figuur 13.



Figuur 13. Tweede orde sectie van de directe-vorm 1.

6.3. Ontwerp van het systeem.

Uitgaande van de filtereigenschappen, zoals die in 6.1 zijn gegeven door de frekwentiekarakteristiek, is de daarbij behorende systeemfunctie bepaald en zijn de coëfficiënten uitgerekend.
Als uitgangspunt voor de realisatie wordt vervolgens een stuctuur gekozen, zoals is voorgesteld in 6.2.
De volgende stap is om met behulp van het concept uit 5.2 een filterprogramma te construeren voor de DSP, in dit geval de TMS320C25.
In dit hoofdstuk maken we het ons wat gemakkelijker door het ontwerpen van het programma uit te voeren met een speciaal stukje gereedschap, een Digital Filter Design Tool.
Het ASPI *Digital Filter Design Package* (DFDP) is een krachtig stuk ontwerpgereedschap dat de gebruiker in staat stelt om in enkele minuten tijd een digitaal filter te ontwerpen en te realiseren. Het pakket is menu-gestuurd en relatief eenvoudig te gebruiken als men ervan uit gaat dat de gebruiker bekend is met filterspecificaties. Het ontwerpprogramma loopt op een IBM (of compatibel) PC of AT waarbij een mathematische coprocessor zeer is aan te bevelen. Bij de beschrijving zal intensief gebruik worden gemaakt van de schermoutput van het programma. Deze output is gegeven in grijze kaders en zal de leidraad vormen voor de beschrijving van de ontwerpprocedure.

Filter Design.

```
*** Digital Filter Design Package ***

(C) COPYRIGHT,1984,1985,1986: ATLANTA SIGNAL PROCESSORS INC ,VERSION 2.12
SN: IBM21274

PROGRAM SELECTION MENU

ENTER THE NUMBER CORRESPONDING TO THE DESIGN TECHNIQUE DESIRED

1. RECURSIVE (IIR) FILTER DESIGN
2. KAISER WINDOW NONRECURSIVE (FIR) FILTER DESIGN
3. PARKS-McCLELLAN EQUIRIPPLE (FIR) FILTER DESIGN
4. TMS320 CODE GENERATOR
5. QUIT

OPTION DESIRED = 1
```

In de aankondiging van DFDP zijn de mogelijkheden van het programma opgesomd.

Van deze mogelijkheden kiezen we de eerste, RECURSIVE (IIR) FILTER DESIGN, voor het ontwerpen van het filter:

```
** Digital Filter Design Package **

IIR BILINEAR TRANSFORM DESIGN PROGRAM

(C) COPYRIGHT,1984,1985,1986: ATLANTA SIGNAL PROCESSORS INC.,VERSION 2.12
SN: IBM21274

THIS FILTER DESIGN PROGRAM DESIGNS RECURSIVE DIGITAL
FILTERS FROM BUTTERWORTH, CHEBYSHEV, AND ELLIPTIC
ANALOG PROTOTYPES.
```

Het IIR-ontwerpprogramma kondigt zich aan en het ontwerp van het BANDPASS filter wordt vervolgens gestart door optie 3.

```
*** IIR BILINEAR TRANSFORM MAIN MENU ***

ENTER THE NUMBER CORRESPONDING TO THE FILTER TYPE DESIRED

    1. LOWPASS
    2. HIGHPASS
    3. BANDPASS
    4. BANDSTOP

OR TAKE THE FOLLOWING ACTION

    5. READ SAVED FILE
    6. RETURN TO PROGRAM SELECTION MENU
    7. QUIT (RETURN TO DOS)

OPTION DESIRED: 3
```

De filtereigenschappen worden in het volgende menu ingevoerd. Let er op dat alle voorkomende frekwenties lager zijn dan de halve samplefrekwentie.
De rimpel in de doorlaatband en in de sperband worden als volgt gedefinieerd.
In de doorlaatband ligt de modulus van de overdrachtfunctie tussen $1+\delta_1$ en $1-\delta_1$ en in de sperband tussen nul en δ_2.
Om verwarring te voorkomen moeten de rimpelwaarden worden gegeven op een lineaire schaal. De bijbehorende demping in decibels wordt dan in de doorlaatband gegeven door $20 \log_{10} (1+\delta 1)$ en in de sperband door $20 \log_{10} (\delta_2)$.

```
ALL FREQUENCIES MUST BE ENTERED IN KILOHERTZ

ENTER SAMPLING FREQUENCY (KHZ) = 20

ENTER BANDPASS FILTER CUTOFF FREQUENCIES

LOWER STOPBAND CUTOFF FREQUENCY (KHZ) = 0.7
LOWER PASSBAND CUTOFF FREQUENCY (KHZ) = 1.0
UPPER PASSBAND CUTOFF FREQUENCY (KHZ) = 1.5
UPPER STOPBAND CUTOFF FREQUENCY (KHZ) = 2.0
PASSBAND RIPPLE =  0.05
STOPBAND RIPPLE =  0.01
```

```
TO MEET YOUR SPECIFICATIONS WOULD REQUIRE FILTERS
OF THE FOLLOWING ORDERS:

    BUTTERWORTH: 12
    CHEBYSHEV I: 8
    CHEBYSHEV II: 8
    ELLIPTIC:  6

IF ONE OF THESE IS SATISFACTORY, ENTER THE
CORRESPONDING NUMBER:

        1. BUTTERWORTH
        2. CHEBYSHEV I
        3. CHEBYSHEV II
        4. ELLIPTIC
OTHERWISE
        5. RETURN TO IIR BILINEAR TRANSFORM MAIN MENU
        6. RETURN TO PROGRAM SELECTION MENU
        7. QUIT (RETURN TO DOS)

OPTION DESIRED: 4
```

We zien dat het betreffende filter kan worden gerealiseerd met behulp van verschillende approximatiemethoden, waarbij de orde van het filter afhankelijk is van de methode. DFDP heeft de mogelijke filterorden reeds bepaald en biedt de ontwerper de keuze. We kiezen hier voor het filter met de laagste orde, ELLIPTIC. Het filter kan dan worden gerealiseerd als een 6^{e} orde filter.
Na enig rekenwerk doet DFDP een voorstel.

*** CHARACTERISTICS OF DESIGNED FILTER ***

ELLIPTIC BANDPASS FILTER

FILTER ORDER = 6
Sampling frequency = 20.000 KiloHertz

	BAND 1	BAND 2	BAND 3
LOWER BAND EDGE	.00000	1.00000	2.00000
UPPER BAND EDGE	.70000	1.50000	10.00000
NOMINAL GAIN	.00000	1.00000	.00000
NOMINAL RIPPLE	.01000	.05000	.01000
MAXIMUM RIPPLE	.00966	.04905	.00967
RIPPLE IN dB	-40.29863	.41593	-40.29389

We zien hier dat het mogelijk is om binnen de gegeven eisen het filter te realiseren. De maximale rimpel in doorlaat en sperband zijn lager dan de opgegeven waarden. Deze waarden zijn nu ook in dB gegeven.

Het volgende menu biedt onder andere de mogelijkheid om de filterorde te verhogen indien de eisen wat aangescherpt moeten worden.

ENTER CORRESPONDING NUMBER FOR
INSTRUCTION DESIRED

1. AUTOMATICALLY INCREMENT FILTER ORDER
2. PLOT RESPONSES
3. DISPLAY FILTER COEFFICIENTS
4. OUTPUT FILTER COEFFICIENTS
5. QUANTIZE COEFFICIENTS

6. RETURN TO IIR BILINEAR TRANSFORM MAIN MENU
7. RETURN TO PROGRAM SELECTION MENU
8. QUIT (RETURN TO DOS)

OPTION DESIRED = 3

Optie 3 toont ons wat DFDP heeft uitgerekend voor de waarden van de filtercoëfficiënten.

INFINITE IMPULSE RESPONSE (IIR)
ELLIPTIC BANDPASS FILTER
UNQUANTIZED COEFFICIENTS

FILTER ORDER = 6
SAMPLING FREQUENCY = 20.000 KILOHERTZ

I	A(I,1)	A(I,2)	B(I,0)	B(I,1)	B(I,2)
1	-1.776235	.916656	.041884	.000000	-.041884
2	-1.743691	.955922	.200070	-.314068	.200070
3	-1.872807	.969655	.642075	-1.253701	.642075

DFDP heeft de bilineaire transformatie gebruikt om het filter te transformeren naar het z-domein. De overdrachtfunctie is opgesplitst in een cascade van tweede orde secties. Hoewel de volgorde van de individuele secties mathematisch niet relevant is, wordt hier voor de reductie van ruis en overflow toch een volgorde bepaald. Hierbij gelden de volgende overwegingen.

- De tweede orde secties krijgen een Q-volgorde zodat het polenpaar met de hoogste Q en het dichtstbijzijnde nulpuntenpaar het verst achteraan wordt geplaatst.

- Ieder volgend polenpaar vormt samen met het dichtstbijzijnde nulpuntenpaar een tweede orde sectie.

- Iedere tweede orde sectie wordt zo geschaald dat zijn maximale versterking kleiner is dan 1.

Voor de implementatie op de digitale signaalprocessor TMS320C25 is het verder nog nodig om de coëfficiënten te beperken in aantal bits. Deze 16-bits processor gebruikt in zijn programmatuur coëfficiënten van 16 bits lengte. De optie QUANTIZE COEFFICIENTS kwantiseert de coëfficiënten tot deze lengte. Dit kan echter betekenen dat de waarden van de gekwantiseerde coëfficiënten zoveel afwijken van de berekende waarden dat er niet meer aan de gestelde eisen wordt voldaan. In ons voorbeeld is dit nu precies aan de orde. We kunnen dit zien in de karakteristieken van de gekwantiseerde versie. De rimpel in de doorlaatband is boven 0,05 uitgestegen. DFDP geeft dan ook de melding "THIS FILTER DOES NOT MEET YOUR SPECIFICATIONS". Het is nu aan de ontwerper om te bepalen of hij een hogere orde neemt of dat hij tevreden is met een geringe overschrijding.
Wij zullen hier de geringe overschrijding aanvaarden. Dit levert ons dan een nieuw stelsel coëfficiënten op.

*** CHARACTERISTICS OF DESIGNED FILTER ***

ELLIPTIC BANDPASS FILTER

FILTER ORDER = 6
Sampling frequency = 20.000 KiloHertz

	BAND 1	BAND 2	BAND 3
LOWER BAND EDGE	.00000	1.00000	2.00000
UPPER BAND EDGE	.70000	1.50000	10.00000
NOMINAL GAIN	.00000	1.00000	.00000
NOMINAL RIPPLE	.01000	.05000	.01000
MAXIMUM RIPPLE	.00965	.05043	.00968
RIPPLE IN dB	-40.30880	.42734	-40.28484

THIS FILTER DOES NOT MEET YOUR SPECIFICATIONS!

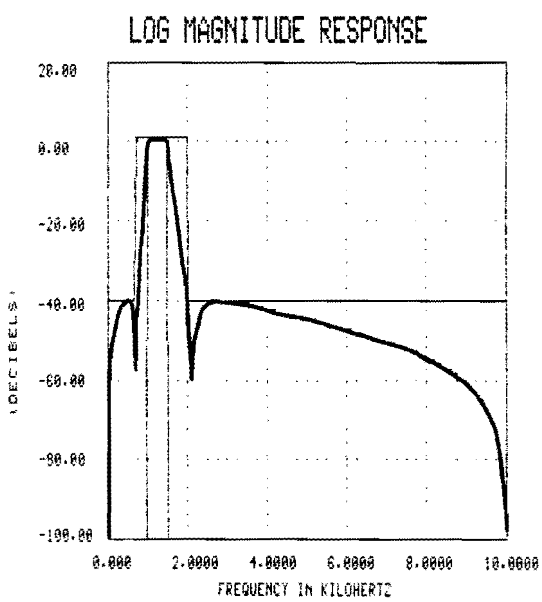
INFINITE IMPULSE RESPONSE (IIR)
ELLIPTIC BANDPASS FILTER
16-BIT QUANTIZED COEFFICIENTS

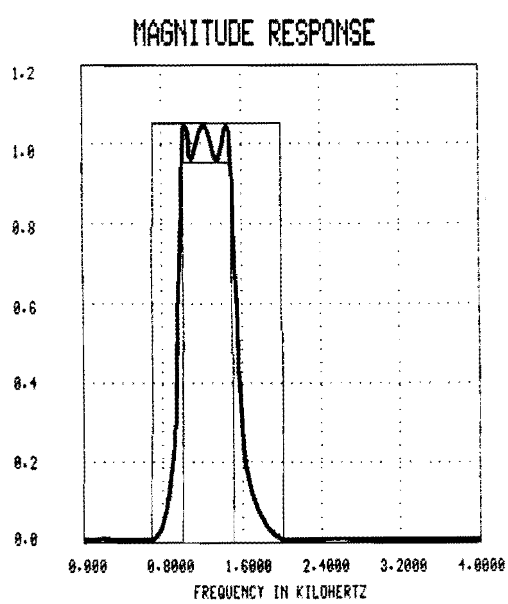
FILTER ORDER = 6
SAMPLING FREQUENCY = 20.000 KILOHERTZ

I	A(I,1)	A(I,2)	B(I,0)	B(I,1)	B(I,2)
1	-1.776184	.916656	.041883	.000000	-.041883
2	-1.743652	.955933	.200073	-.314072	.200073
3	-1.872742	.969666	.642090	-1.253723	.642090

Met de optie PLOT RESPONSES kunnen verschillende kakteristieken van het filter in beeld worden gebracht, zoals bijvoorbeeld de getoonde MAGNITUDE RESPONSE.

```
** PLOT MENU **

1. MAGNITUDE RESPONSE
2. LOG MAGNITUDE RESPONSE
3. PHASE RESPONSE
4. GROUP DELAY
5. POLE-ZERO PLOT
6. IMPULSE RESPONSE
7. DISPLAY ALL PLOTS
8. TERMINATE PLOTTING

OPTION DESIRED = 1
```





Code-generator.

Uitgaande van de berekende overdrachtsfunctie kan men nu een filterstructuur programmeren. Hier zullen we echter gebruik maken van een code-generator.
Als de berekende waarden van het filter zijn opgeslagen in een file kan vanuit deze file de programmacode voor de TMS320C25 worden gegenereerd met behulp van het programma CGEN. Men kan hierbij nog een keuze maken uit verschillende processoren, adresseermodes en precisie.

```
                    ** Digital Filter Design Package **
                        CODE GENERATION PROGRAM

    (C) Copyright (1987): ATLANTA SIGNAL PROCESSORS, INC., version 3.20
                              SN: IBM21274

Filter Specification Filename? (IIR.FLT   )   : PDSP
     Output Code Filename? (PDSP   )          :
   Unique 5 character Identifier? (PDSP )     :
  Generate a fully executable filter? (Y)     : N


Generating code for IIR - 2nd order sections. Target processor(s):
  1 - TMS32010; Direct Paged Memory
  2 - TMS32010; Direct Paged Memory; Increased Precision
  3 - TMS32020; Direct Paged Memory
  4 - TMS32020; Direct Paged Memory; Increased Precision
  5 - TMS32020; Indexed Memory
  6 - TMS32020; Indexed Memory; Increased Precision
  7 - TMS32020; Index Memory; Looping Control
  8 - TMS32020; Indexed Memory; Looping Control; Increased Precision
  9 - TMS320c25; Direct Paged Memory
 10 - TMS320c25; Direct Paged Memory; Increased Precision
 11 - TMS320c25; Indexed Memory
 12 - TMS320c25; Indexed Memory; Increased Precision
 13 - TMS320c25; Indexed Memory; Looping Control
 14 - TMS320c25; Indexed Memory; Looping Control; Increased Precision

Please enter your choice for target processor implementation (1 - 14)? 9
```

Het programma "PDSP" is het resultaat van een code-generatie slag en kan als subroutine worden aangeroepen.

```
          IDT        'PDSP'                *UNIQUE NAME
          DEF        FPDSP                 *Name of filter subroutine
          DEF        IPDSP                 *Name of filter initialization subroutine
**********************************************
*  ASPI TMS320c25 DIGITAL FILTER REALIZATION
**********************************************
*          Direct Paged Memory
*        3-STAGE RECURSIVE FILTER
*         SECOND ORDER SECTIONS
*       FILTER GENERATED FROM FILE
*              pdsp.FLT
*        Fri Feb 22 14:04:22 1991
*
*     Filter type: BANDPASS
* Approximation type: ELLIPTIC
*   Sampling freq: 20000 HZ
*
**********************************************
          PSEG
**********************************************
*        DATA MEMORY DEFINITION
**********************************************
*   CONTAINS:
*        STORAGE FOR FILTER INPUT AND OUTPUT
*        STORAGE FOR COEFFICIENTS
*        STORAGE FOR DELAY ELEMENTS
**********************************************
*
* FILTER INPUT and OUTPUT STORAGE
*
          REF        FILTT,VPDSP
*
COEF
* COEFFICIENT INITIALIZATION STORAGE AREA
*
*
*  SECOND-ORDER SECTION # 01
*
          DATA       21959                 *B0
          DATA       29101                 *A1
          DATA       -30037                *A2
          DATA       -21959                *B2
*
*  SECOND-ORDER SECTION # 02
*
          DATA       13112                 *B0
          DATA       -20583                *B1
          DATA       28568                 *A1
          DATA       -31324                *A2
          DATA       13112                 *B2
*
*  SECOND-ORDER SECTION # 03
*
          DATA       10520                 *B0
          DATA       -20541                *B1
          DATA       30683                 *A1
          DATA       -31774                *A2
          DATA       10520                 *B2
*
          PEND
```

```
*
          CSEG      'XPDSP'
* DELAY STORAGE DATA STORAGE AREA
*
DELAY     EQU       $
Z011      BSS       1
Z012      BSS       1
Z021      BSS       1
Z022      BSS       1
Z031      BSS       1
Z032      BSS       1
*
* COEFFICIENT DATA STORAGE AREA
*
FDATA     EQU       $
*
B010      BSS       1                 *B0
A011      BSS       1                 *A1
A012      BSS       1                 *A2
B012      BSS       1                 *B2
*
B020      BSS       1                 *B0
B021      BSS       1                 *B1
A021      BSS       1                 *A1
A022      BSS       1                 *A2
B022      BSS       1                 *B2
*
B030      BSS       1                 *B0
B031      BSS       1                 *B1
A031      BSS       1                 *A1
A032      BSS       1                 *A2
B032      BSS       1                 *B2
*
          CEND
*
          PSEG
*********************************************
*     FILTER INITIALIZATION SUBROUTINE
*********************************************
IPDSP
          LARP      0                 *POINT TO AR0
          LRLK      0,FDATA           *POINTER TO DATA MEMORY
          RPTK      13                *COUNT FOR NUMBER OF POINTS
          BLKP      COEF,*+           *BLOCK MOVE OF COEF.
          ZAC                         *CLEAR ACCUMULATOR
          LRLK      0,DELAY
          RPTK      5                 *NUMBER OF DELAY POINTS
          SACL      *+                *CLEAR DATA VALUE
          RET                         *INIT RETURN
*********************************************
*         FILTER SUBROUTINE
*********************************************
* ASSUMPTIONS:
*   SATURATION ARITHMETIC MODE IS ON
*   P REGISTER OUTPUT SHIFT = 1
*   PAGE REGISTER IS SET TO 'IOPAGE'
*   SIGN EXTEND MODE IS ON
*
* INPUT:  VPDSP
* OUTPUT:  VPDSP
```

```
*
**********************************************
*
FPDSP
*
* SECOND-ORDER FILTER SECTION # 01
*
        LAC     VPDSP,12        *GET & SCALE INPUT
        SACH    FILTT           *SAVE SCALED INPUT
        LT      FILTT           *GET SCALED INPUT
        MPY     B010            *P = B0 * INPUT
        ZALH    Z011            *AC = Z-1
        APAC                    *AC = Z-1 + (B0 * INPUT)
        SACH    VPDSP           *Save in OUTPUT
        LT      VPDSP           *Get OUTPUT
        ZALH    Z012            *AC = Z-2
        MPY     A011            *P = A1 * OUTPUT
        APAC
        MPYA    A012            *AC = (B1 * INPUT) + (A1 * OUTPUT)
*                               *P = A2 * OUTPUT
        SACH    Z011            *Save in Z-1
        LTP     FILTT           *AC = A2 * OUTPUT
        MPY     B012            *P = B2 * INPUT
        APAC                    *AC = (B2 * INPUT) + (A2 * OUTPUT)
        SACH    Z012            *Save in Z-2
*
* SECOND-ORDER FILTER SECTION # 02
*
        LAC     VPDSP,15        *GET & SCALE INPUT
        SACH    FILTT           *SAVE SCALED INPUT
        LT      FILTT           *GET SCALED INPUT
        MPY     B020            *P = B0 * INPUT
        ZALH    Z021            *AC = Z-1
        MPYA    B021            *AC = Z-1 + (B0 * INPUT)
*                               *P = B1 * INPUT
        SACH    VPDSP           *Save in OUTPUT
        LTP     VPDSP           *AC = B1 * INPUT
        ADDH    Z022            *AC = Z-2 + (B1 * INPUT)
        MPY     A021            *P = A1 * OUTPUT
        APAC
        MPYA    A022            *AC = Z-2 + (B1 * INPUT) + (A1 * OUTPUT)
*                               *P = A2 * OUTPUT
        SACH    Z021            *Save in Z-1
        LTP     FILTT           *AC = A2 * OUTPUT
        MPY     B022            *P = B2 * INPUT
        APAC                    *AC = (B2 * INPUT) + (A2 * OUTPUT)
        SACH    Z022            *Save in Z-2
*
* SECOND-ORDER FILTER SECTION # 03
*
        ZALH    VPDSP           *GET INPUT
        ADDH    VPDSP           *SCALE INPUT
        SACH    FILTT           *SAVE SCALED INPUT
        LT      FILTT           *GET SCALED INPUT
        MPY     B030            *P = B0 * INPUT
        ZALH    Z031            *AC = Z-1
        MPYA    B031            *AC = Z-1 + (B0 * INPUT)
*                               *P = B1 * INPUT
        SACH    VPDSP           *Save in OUTPUT
        LTP     VPDSP           *AC = B1 * INPUT
```

```
        ADDH    Z032        *AC = Z-2 + (B1 * INPUT)
        MPY     A031        *P = A1 * OUTPUT
        APAC
        MPYA    A032        *AC = Z-2 + (B1 * INPUT) + (A1 * OUTPUT)
*                           *P = A2 * OUTPUT
        SACH    Z031        *Save in Z-1
        LTP     FILTT       *AC = A2 * OUTPUT
        MPY     B032        *P = B2 * INPUT
        APAC                *AC = (B2 * INPUT) + (A2 * OUTPUT)
        SACH    Z032        *Save in Z-2
*
        RET                 *RETURN
        PEND
        END
```

```
        IDT     'MAINPDSP'
        REF     FPDSP,IPDSP       * These are the two symbols that
*                                 *  are defined in the code generated
*                                 *  by CGEN
*
        DEF     FILTT,VPDSP       * These symbols are referrenced by
*                                 *  code generated by CGEN
*
IOPAGE  EQU     6                 * Use direct page 6
*
DOUT    EQU     2                 * D/A output port
DIN     EQU     2                 * A/D input port
*
* Define the locations in the direct page
*
        DSEG
*
VPDSP   BSS     1                 * Input sample to filter subroutine
*                                 * and output sample from subroutine
*
FILTT   BSS     1                 * Temporary used by filter subroutine
*
TEMP    BSS     1                 * Temporary register
*
        DEND
*
**********************************************
*                                            *
*       MAIN PROGRAM                         *
*       Bandpass Filter                      *
*                                            *
* This program assumes that the TMS32020/C25 *
* controls a T.I. Analog Interface Board.    *
* The mode of the A/D and D/A conversion     *
* is set by writing a mode-value to I/O      *
* location 0.                                *
* The conversion rate is set by writing a    *
* rate-value to I/O location 1.              *
* The A/D and D/A are addressed at I/O       *
* location 2 (i.e. when the TMS320 reads     *
* I/O location 2 it reads the data from the  *
* A/D and when it writes I/O location 2      *
* it writes data to the D/A).                *
* The sample-clock is starting the A/D and   *
* the end of convert signal sets a flip-flop *
* while the output of the flip-flop is       *
* connected to the BIO pin on the TMS320.    *
* This flip-flop is cleared by a read from   *
* I/O location 2.                            *
* Writing to I/O location 3 resets the A.I.B.*
*                                            *
**********************************************
*
        PSEG
        B       START
        BSS     30
*
**********************************************
*       A.I.B. SETTINGS
**********************************************
```

```
MODE      DATA      >A                  Normal mode
RATE      DATA      249                 20 kHz
*
START
*
********************************************
*         MAIN I/O LOOP
********************************************
*
* Initialize the TMS320C25
*
          LDPK      IOPAGE              * Set the I/O direct page
          SPM       1                   * Product Output shift = 1 bit
          SSXM                          * Set Sign Extend Mode ON
          SOVM                          * Set Overflow Arith. ON
          DINT                          * Disable Interrupt
*
* Initialize the A.I.B.
*
          LALK      MODE
          TBLR      TEMP
          OUT       TEMP,PA0
          LALK      RATE
          TBLR      TEMP
          OUT       TEMP,PA1
          LALK      >FFFF
          SACL      TEMP
          OUT       TEMP,PA3
*
* Initialize the FILTER
*
          CALL      IPDSP
*
*Start of Main I/O Loop
*
IOLOOP
*
* Wait for end of convert.
*
          BIOZ      GET                 * Wait for end of convert
          B         IOLOOP              * Branch if not
* Output filtered sample and get new one.
*
GET       OUT       VPDSP,DOUT
          IN        VPDSP,DIN           * Input sample
*
* Filter sample
*
          CALL      FPDSP
*
* FPDSP expects a two's complement digital sample in the PAGE storage
* location VPDSP. FPDSP will output the filtered sample into the same
* PAGE storage location VPDSP.
*
* Wait for next sample
*
          B         IOLOOP              * Loop
*
          PEND
          END
```

Het voorgaande programma "MAINPDSP" is een hoofdprogramma dat kan worden gebruikt om het Analog Interface Board (AIB) van T.I. aan te sturen.

Linker.

Het gegenereerde programma "PDSP" kan met behulp van een Linker aan dit programma worden vastgeknoopt.
De opdracht-file aan de Linker ziet er dan als volgt uit.

```
TASK     PDSP
PROGRAM  >0
DATA     >300
COMMON   >303 XPDSP
INCLUDE  MAINPDSP.MPO
INCLUDE  PDSP.MPO
END
```

De Linker levert onderstaande object-file, die aan een Software Development Systeem kan worden aangeboden.

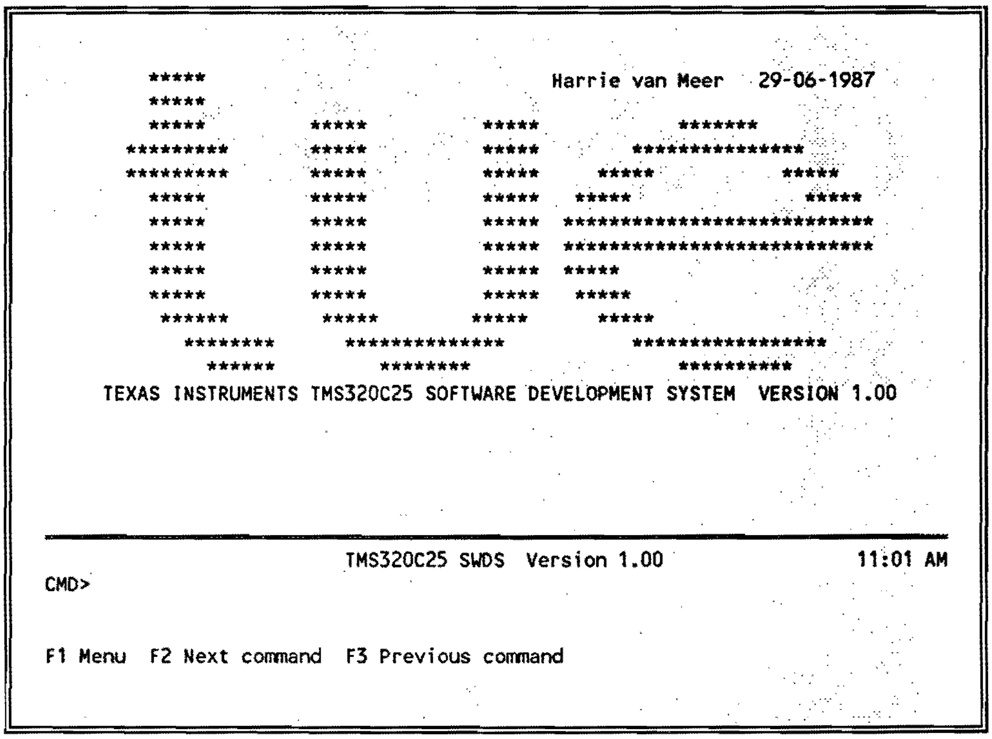
```
K0000PDSP   90000BFF80B002290020B000AB00F9BC806BCE09BCE07BCE037F1EFF
PDSP0001
BCE01BD001B0020B5802BE002BD001B0021B5802BE102BD001BFFFFB6002BE3027F17BF
PDSP0002
BFE80B004DBFA80B0039BFF80B0035BE200B8200BFE80B0059BFF80B00359003F7F13FF
PDSP0003
B55C7B71ADB8AABBAA39B3338BAF99B6F98B85A4B3338B2918BAFC3B77DBB83E27F0A3F
PDSP0004
B29189004DB5588BD000B0309BCB0DBFCA0B003FBCA00BD000B0303BCB05B60A07F147F
PDSP0005
BCE26B2C00B6801B3C01B3809B4003BCE15B6800B3C00B4004B380ABCE15B3A0B7F15FF
PDSP0006
B6803B3E01B380CBCE15B6804B2F00B6801B3C01B380DB4005B3A0EB6800B3E007F172F
PDSP0007
B4806B380FBCE15B3A10B6805B3E01B3811BCE15B6806B4000B4800B6801B3C017F188F
PDSP0008
B3812B4007B3A13B6800B3E00B4808B3814BCE15B3A15B6807B3E01B3816BCE157F17EF
PDSP0009
B6808BCE267FD7FF
PDSP0010
:   PDSP    02/22/91  14:07:17   XLNKPC  v3.1   88.005      PDSP0011
```

De Linker levert tevens een verslag van zijn werkzaamheden in de vorm van de volgende MAP-file.

```
PC/CrossWare Family Linker v3.1 88.005  02/22/91  14:07:17          Page  1
 Copyright (C) 1986, 1987 Texas Instruments Inc.  All Rights Reserved
 Command List

 TASK     PDSP
 PROGRAM  >0
 DATA     >300
 COMMON   >303 XPDSP
 INCLUDE  MAINPDSP.MPO
 INCLUDE  PDSP.MPO
 END

PC/CrossWare Family Linker v3.1 88.005  02/22/91  14:07:17          Page  2
 Copyright (C) 1986, 1987 Texas Instruments Inc.  All Rights Reserved
 Link Map

 Control File = PDSP.CTL

 Linked Output File = PDSP.LOD

 List File = PDSP.MAP

 Output Format = ASCII


PC/CrossWare Family Linker v3.1 88.005  02/22/91  14:07:17          Page  3
 Copyright (C) 1986, 1987 Texas Instruments Inc.  All Rights Reserved

 Phase 0    PDSP     Module Origin = 0000   Length = 0000

 Module    No  Origin  Length    Type      Date      Time     Creator

 MAINPDSP  1   0000*    003F    INCLUDE   02-22-91  14:07:06  ASM320
 $DATA     1   0300*    0003
 PDSP      2   003F*    004F    INCLUDE   02-22-91  14:07:11  ASM320


 Common    No  Origin  Length

 XPDSP     2   0303*    0014


                 D E F I N I T I O N S


  Name   Value No   Name   Value No   Name   Value No   Name   Value No

  FILTT  0301*  1   FPDSP  0059*  2   IPDSP  004D*  2   VPDSP  0300*  1


 Length of Region for Task          = 0000


 Number of Records for Module PDSP  =  11

 Total Records Written              =  11

 **** Linking Completed   02/22/91  14:07:19
```

Software Evaluation.

Het *Software Development System* (SWDS) van T.I. wordt vervolgens ingezet om het ontwikkelde programma te testen.
SWDS kondigt zich aan met een logo en kan worden gebruikt in een command mode of in een menu mode.



Met de opdracht LOAD wordt de object-code file geladen waarna via de opdracht DEBUG de komplete status van de processor op het scherm wordt getoond.
De debugger geeft in aparte windows verschillende soorten van systeeminformatie:

- De inhouden van de processor-registers.
- De status van het systeem.
- De toestand van een aantal inputpinnen.
- De reverse-assembler tekst van het programma.
- De inhoud van het data/program memory.

```
DEBUGGER: Breakpoint, Count, Execute, Go, Halt, Load, Quit, Run, Show, sTep
                          Program state
  PC = 35            AR0 = 309     AR4 = 0       S1 = 0        S5 = 0
 ACC = 10244         AR1 = 0       AR5 = 0       S2 = 0        S6 = 0
   P = FFFD1C50      AR2 = 0       AR6 = 0       S3 = 0        S7 = 0
   T = FFEE          AR3 = 0       AR7 = 0       S4 = 0       IMR = FFC0
 ARP   OV   OVM   INTM   DP   ARB   CNF   TC   SXM   C   HM   FSM   XF   FO   TXM   PM
  0     0    1     1      6    0     0     0    1    1    0    0     0    0    0    1

      Status                           Hardware signal monitor
 Processor running      INT0 = 1 INT1 = 1 INT2 = 1 BIO = 1 XF = 0 RESET = 1

          Reverse assembler                        Data memory
 0022     LDPK  6                   0300 =  FFF9   FFEE   FFFF   FFE7
 0023     SPM   1                   0304 =  001C   FFFA   0002   FFFE
 0024     SSXM                      0308 =  0001   55C7   71AD   8AAB
 0025     SOVM                      030C =  AA39   3338   AF99   6F98
 0026     DINT                      0310 =  85A4   3338   2918   AFC3
 0027     LALK  20,0                0314 =  77DB   83E2   2918   0000
 0029     TBLR  2                   0318 =  0000   0000   0000   0000
 002A     OUT   2,0                 031C =  0000   0000   0000   0000
 002B     LALK  21,0                0320 =  0000   0000   0000   0000
 002D     TBLR  2                   0324 =  0000   0000   0000   0000
F1 Help  F2 Toggle data/program memory  F3 Command card  F4 Modify windows
```

De opdracht DPM toont of print een display van de inhoud van het programmageheugen.

```
TMS320C25 SWDS  Version 1.00  22-Feb-1991    2:49 PM

0000  FF80  0022  5500  5500  5500  5500  5500  5500  ..."U.U.U.U.U.U.
0008  FFFF  FFFF  FFFF  FFFF  FFFF  FFFF  FFFF  FFFF  ................
0010  FFFF  FFFF  FFFF  FFFF  FFFF  FFFF  FFFF  FFFF  ................
0018  5500  5500  5500  5500  5500  5500  5500  5500  U.U.U.U.U.U.U.U.
0020  000A  00F9  C806  CE09  CE07  CE03  CE01  D001  ................
0028  0020  5802  E002  D001  0021  5802  E102  D001  . X......!X.....
0030  FFFF  6002  E302  FE80  004D  FA80  0039  FF80  ..`......M...9..
0038  0035  E200  8200  FE80  0059  FF80  0035  55C7  .5.......Y...5U.
0040  71AD  8AAB  AA39  3338  AF99  6F98  85A4  3338  q....938..o...38
0048  2918  AFC3  77DB  83E2  2918  5588  D000  0309  )...w...).U.....
0050  CB0D  FCA0  003F  CA00  D000  0303  CB05  60A0  .....?.........`.
0058  CE26  2C00  6801  3C01  3809  4003  CE15  6800  .&,.h.<.8.@...h.
0060  3C00  4004  380A  CE15  3A0B  6803  3E01  380C  <.@.8...:.h.>.8.
0068  CE15  6804  2F00  6801  3C01  380D  4005  3A0E  ..h./.h.<.8.@.:.
0070  6800  3E00  4806  380F  CE15  3A10  6805  3E01  h.>.H.8...:.h.>.
0078  3811  CE15  6806  4000  4800  6801  3C01  3812  8...h.@.H.h.<.8.
0080  4007  3A13  6800  3E00  4808  3814  CE15  3A15  @.:.h.>.H.8...:.
0088  6807  3E01  3816  CE15  6808  CE26  5500  5500  h.>.8...h..&U.U.
```

Met de Reverse Assembler (RASM) kunnen de instructies in de debugger worden nagetrokken.

Een korte verklaring van de programmageheugen-indeling:

>00	t/m	>3E	programma MAINPDSP
>3F	t/m	>4C	filtercoëff. in PDSP
>4D	t/m	>8D	programma PDSP

(Geheugenplaatsen >3F t/m >4C worden door de Reverse Assembler omgezet naar mnemonics die hier geen enkele betekenis hebben.)

```
TMS320C25 SWDS  Version 1.00  22-Feb-1991   2:48 PM

0000  B     22            002F  LALK  FFFF,0      0064  MPYA  B
0002  NOP                 0031  SACL  2,0         0065  SACH  3,0
0003  NOP                 0032  OUT   2,3         0066  LTP   1
0004  NOP                 0033  CALL  4D          0067  MPY   C
0005  NOP                 0035  BIOZ  39          0068  APAC
0006  NOP                 0037  B     35          0069  SACH  4,0
0007  NOP                 0039  OUT   0,2         006A  LAC   0,F
0008  DATA  FFFF          003A  IN    0,2         006B  SACH  1,0
0009  DATA  FFFF          003B  CALL  59          006C  LT    1
000A  DATA  FFFF          003D  B     35          006D  MPY   D
000B  DATA  FFFF          003F  MAR   *BR0-       006E  ZALH  5
000C  DATA  FFFF          0040  SAR   1,*+,5      006F  MPYA  E
000D  DATA  FFFF          0041  IN    *+,A,3      0070  SACH  0,0
000E  DATA  FFFF          0042  MPYK  A39         0071  LTP   0
000F  DATA  FFFF          0043  LAR   3,38        0072  ADDH  6
0010  DATA  FFFF          0044  MPYK  F99         0073  MPY   F
0011  DATA  FFFF          0045  SACH  *-,7,0      0074  APAC
0012  DATA  FFFF          0046  IN    *+,5        0075  MPYA  10
0013  DATA  FFFF          0047  LAR   3,38        0076  SACH  5,0
0014  DATA  FFFF          0048  LAC   18,9        0077  LTP   1
0015  DATA  FFFF          0049  MPYK  FC3         0078  MPY   11
0016  DATA  FFFF          004A  SAR   7,*0-,3     0079  APAC
0017  DATA  FFFF          004B  IN    *0+,3       007A  SACH  6,0
0018  NOP                 004C  LAC   18,9        007B  ZALH  0
0019  NOP                 004D  LARP  0           007C  ADDH  0
001A  NOP                 004E  LRLK  0,309       007D  SACH  1,0
001B  NOP                 0050  RPTK  D           007E  LT    1
001C  NOP                 0051  BLKP  3F,*+       007F  MPY   12
001D  NOP                 0053  ZAC               0080  ZALH  7
001E  NOP                 0054  LRLK  0,303       0081  MPYA  13
001F  NOP                 0056  RPTK  5           0082  SACH  0,0
0020  ADD   A,0           0057  SACL  *+,0        0083  LTP   0
0021  ADD   *BR0+,0,1     0058  RET               0084  ADDH  8
0022  LDPK  6             0059  LAC   0,C         0085  MPY   14
0023  SPM   1             005A  SACH  1,0         0086  APAC
0024  SSXM                005B  LT    1           0087  MPYA  15
0025  SOVM                005C  MPY   9           0088  SACH  7,0
0026  DINT                005D  ZALH  3           0089  LTP   1
0027  LALK  20,0          005E  APAC              008A  MPY   16
0029  TBLR  2             005F  SACH  0,0         008B  APAC
002A  OUT   2,0           0060  LT    0           008C  SACH  8,0
002B  LALK  21,0          0061  ZALH  4           008D  RET
002D  TBLR  2             0062  MPY   A
002E  OUT   2,1           0063  APAC
```

Eindtest.

In het algemeen zal men in de testfase moeten terugvallen op het wijzigen van de sourceteksten, waarna opnieuw het traject van assembleren, linken en debuggen zal moeten worden doorlopen.
Nadat men stap voor stap door het programma is gelopen met al of niet gebruikmaking van breakpoints en het programma voldoet aan de gestelde eisen, kan het SWDS worden gekoppeld met het targetboard.
Hiermee is de deze ontwerpfase afgerond en kan de ontwerper tevreden achteroverleunend terugzien op een geslaagde realisatie.

DIGITALE SIGNAALBEWERKING

deel II.2

Speciale Filterstructuren

door

J.H.F. Ritzerfeld

Technische Universiteit Eindhoven

SPECIALE FILTERSTRUCTUREN

INLEIDING

Bij een gegeven overdrachtsfunctie $H(z)$ zijn er vele manieren om een digitaal filter te realiseren. De directe weg heeft daarbij veelal niet de voorkeur. Zo hebben we bijvoorbeeld al gezien dat een filter vaak wordt gerealiseerd als samenstel van tweede (en eerste) orde secties in cascade of parallel. Alle filter-realiseringen die anders zijn dan de directe vorm zullen we aanduiden met *speciale filterstructuren.* Onze interesse in speciale structuren is drieledig:

- Verminderen van coëfficiënt–gevoeligheid.

De vermenigvuldigers in een digitaal filter – in een directe vorm filter de coëfficiënten van de overdrachtsfunctie – worden gerealiseerd met een eindig aantal bits. In het algemeen zal dit betekenen dat de coëfficiënten afwijken van de gewenste waarde. De invloed van de waarden van de vermenigvuldigers op de modulus van de overdrachtsfunctie – de coëfficiënt–gevoeligheid – kan met een speciale filterstructuur echter veel kleiner zijn dan met een directe vorm realisering.

- Verminderen van niet–lineaire effecten.

Naast het kwantiseren van de *waarden* waarmee kan worden vermenigvuldigd, moet in een recursief digitaal filter ook de *uitkomst* van een vermenigvuldiging worden gekwantiseerd om een steeds verder toenemende woordlengte te vermijden. Het eerste effect is lineair en veroorzaakt slechts een (kleine) afwijking van de gewenste overdracht. Het tweede effect is niet–lineair en geeft aanleiding tot ongewenste oscillaties (*limit cycles*) en kwantisatieruis. Daarnaast moet in een digitaal filter ook actie worden genomen indien een signaal de maximaal te representeren waarde – met het gegeven aantal bits – overschrijdt. Deze *overflow correctie* kan eveneens ongewenste oscillaties (*overflow oscillations*) tot gevolg hebben. Beide oscillaties kunnen echter met een speciale filterstructuur geheel worden vermeden, hetgeen niet met een directe vorm realisering (ook niet met een samenstel van tweede orde directe vorm secties) kan worden gegarandeerd. Kwantisatieruis kan als zodanig nooit geheel worden vermeden. De signaal/ruis-verhouding kan echter beduidend worden verbeterd met een geschikt gekozen filterstructuur, zonder het aantal bits waarmee de signalen worden gerepresenteerd groter te kiezen.

- Lineaire stabiliteit met begrensde vermenigvuldigers.

Voor bepaalde toepassingen - met name adaptieve filters - kan het voordelig zijn om een eenvoudig criterium te hebben waarmee lineaire stabiliteit in een recursief filter kan worden gegarandeerd. Een lattice filter van willekeurige orde is lineair stabiel - zoals we zullen zien - als alle vermenigvuldigers in absolute waarde kleiner dan één zijn. Vergelijk dit met een directe vorm realisering, waarbij al in een tweede orde sectie de coëfficiënten in de stabiliteitsdriehoek moeten worden gekozen.

Deze drie punten zijn in feite nauw met elkaar verweven. Zo blijkt een structuur met een lage coëfficiënt–gevoeligheid ook goed te scoren op het punt van de kwantisatieruis en de niet–lineaire stabiliteit - d.i. het niet optreden van ongewenste oscillaties. Ook blijkt het derde punt een uitvloeisel te zijn van het streven om structuren te ontwerpen met een betere performance op de eerste twee punten. Een en ander houdt verband met het feit dat goed filterontwerp een gemeenschappelijke noemer bevat die eigenlijk al bekend is uit de analoge filtertechniek. Deze onderliggende gedachte zullen we eerst proberen te ontwikkelen, waarna achtereenvolgens de belangrijkste speciale structuren aan de orde komen, te weten:

- *Lattice– en Ladderfilters* (tralie– en ladderfilters) of algemener *Orthogonale Filters.*
- *Wave Digital Filters* (WDF's of digitale golffilters).
- *State–Space Filters.*

ALGEMENE AANPAK

In de analoge filtertechniek speelt het probleem van de coëfficiënt–gevoeligheid eveneens een rol. Ook daar willen we filters ontwerpen waarvan de overdracht zo min mogelijk gaat afwijken bij afwijkende waarden van de spoelen (L's), weerstanden (R's) en condensatoren (C's) in de klassieke filters. De oplossing die gevonden werd, berust op het concept van *verliesvrijheid* en *passiviteit.* Deze beide begrippen kunnen we ook invoeren in de digitale filtertheorie om ongevoelige filters te ontwerpen. Om een en ander duidelijk te maken, is in Fig. 1 een typisch analoog laagdoorlaatfilter getekend met een lage coëfficiënt–gevoeligheid. De elementen binnen de omhulling zijn alle verliesvrij (L's en C's) en de afsluitimpedanties aan de beide poorten zijn passief (R's). De bijbehorende

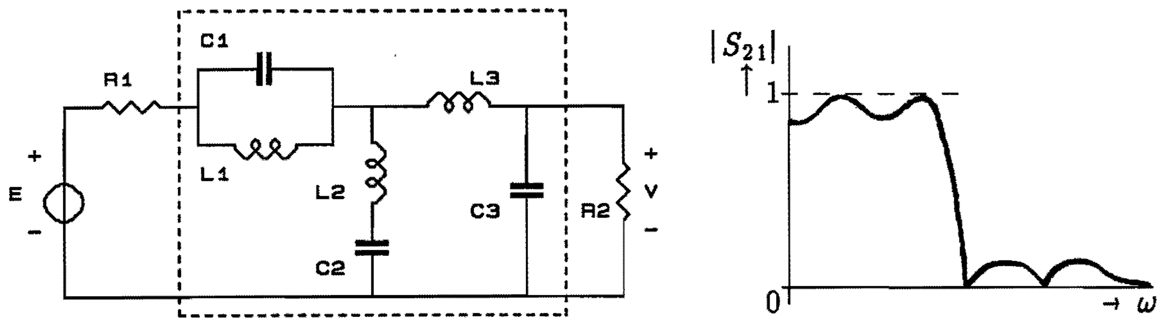


Fig. 1 *Analoog laagdoorlaatfilter met bijbehorende amplituderesponsie*

amplituderesponsie kent in het algemeen een aantal frequenties waar de (vermogens–)overdracht één of nul is. De bijbehorende punten in het complexe p–vlak heten – niet onlogisch – dempingsnulpunten resp. dempingspolen (waar de demping oneindig wordt en de overdracht dus nul). In zowel nulpunten als polen van demping is de amplituderesponsie ongevoelig voor coëfficiënt–variatie op grond van de volgende overwegingen.

- De demping kan niet negatief worden, omdat alle elementen in het filter passief zijn (positieve R's, L's en C's) en negatieve demping zou duiden op aanwezigheid van minstens één actief element. Daardoor zal de amplitude–responsie in de maxima een afgeleide nul hebben naar alle elementwaarden in het filter.

  $$\frac{\partial|H|}{\partial L_i} = \frac{\partial|H|}{\partial C_i} = 0, \quad \text{als} \quad |H(j\omega)| = 1.$$

  Definiëren we de gevoeligheid of *sensitivity* S_c van het filter voor variatie van een coëfficiënt c op de gebruikelijke manier als de relatieve verandering van de modulus van H als gevolg van de relatieve verandering van c, dan kunnen we schrijven $S_c = \lim\limits_{\Delta c \to 0} \frac{\Delta|H|/|H|}{\Delta c/c} = 0$ op grond van bovenstaande relatie.

- Alle elementen binnen de omhulling – een verliesvrije *tweepoort* – hebben een imaginaire impedantie. Daardoor zullen de polen (en nulpunten) van demping op de imaginaire as in het p–vlak optreden, en – wat belangrijker is – de imaginaire as niet kunnen verlaten. Het gevolg is, dat de nulpunten in de amplituderesponsie ook echt nulpunten blijven bij coëfficiënt–variatie en hooguit een beetje op de frequentie–as kunnen schuiven. In een plaatje van modulus H op logarithmische schaal komt dit tot uiting, doordat de nulpunten (oneindig diepe) putten zijn, die bij afwijkende elementwaarden horizontaal kunnen verschuiven, echter niet kunnen worden opgevuld.

Soortgelijke redeneringen zouden we ook in het digitale domein kunnen voeren, met dien verstande dat dan de eenheidscirkel in het complexe z-vlak de rol van de imaginaire as in het p-vlak vervult. Daartoe moeten we echter eerst de begrippen verliesvrijheid en passiviteit invoeren in de digitale filtertheorie. Definiëren we het *momentane vermogen* van een tijddiscreet signaal $x[n]$ als $p[n] = x^2[n]$ en het *gemiddelde vermogen* of kortweg het *vermogen P* als het tijdgemiddelde van $p[n]$, dan kunnen we stellen:

> Een digitale bouwsteen is verliesvrij als het totale vermogen dat de bouwsteen instroomt via ingaande signalen gelijk is aan het totale vermogen dat de bouwsteen verlaat via uitgaande signalen, $P_u = P_i$.
> Een digitale bouwsteen is passief als het uitgaande vermogen niet groter is dan het ingaande vermogen, $P_u \leq P_i$.

Beschouwen we twee eenvoudige digitale bouwstenen, een *digital one-pair* - met één ingangssignaal en één uitgangssignaal - en een *digital two-pair* - met twee ingangssignalen en twee uitgangssignalen. Overeenkomstig kennen we in de analoge wereld de algemene impedantie of eenpoort resp. de tweepoort. In Fig. 2 zijn de beide digitale bouwstenen getekend.

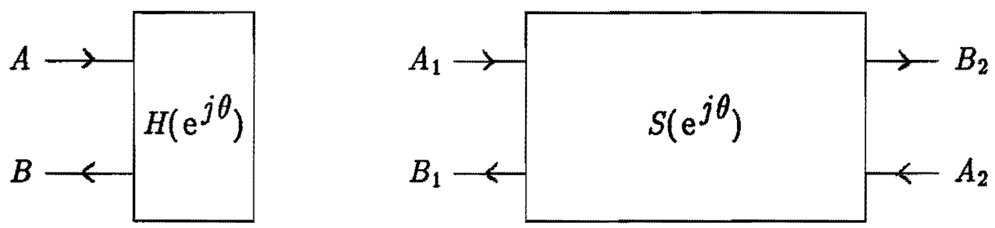


Fig. 2 *Digital one-pair* *en* *digital two-pair*

Een one-pair kunnen we algemeen beschrijven met een overdrachtsfunctie $H(z)$, een two-pair met een matrix van functies van z, die om historische reden S wordt genoemd (we komen daarop later nog terug).

$$\begin{bmatrix} B_1(z) \\ B_2(z) \end{bmatrix} = \begin{bmatrix} S_{11}(z) & S_{12}(z) \\ S_{21}(z) & S_{22}(z) \end{bmatrix} \begin{bmatrix} A_1(z) \\ A_2(z) \end{bmatrix} \quad \text{of in matrix-notatie} \quad \underline{B} = S\,\underline{A}$$

De A's en B's zijn z-getransformeerden van de in- en uitgangssignalen, of, zoals in Fig. 2, *complexe amplituden* van deze signalen in het geval van harmonische tijdafhankelijkheid: $a_i[n] = \mathrm{Re}(A_i\, e^{j\theta n})$ resp. $b_i[n] = \mathrm{Re}(B_i\, e^{j\theta n})$.
Het vermogen van een harmonisch signaal met complexe amplitude A is $|A|^2/2$.

Een digital one-pair is *verliesvrij* indien voor elke frequentie θ geldt: $|B|=|A|$. Voor de frequentieresponsie betekent dit: $|H(e^{j\theta})| = 1$ voor alle θ.

Een dergelijke overdracht heet een *all-pass* filter of fase-draaier, omdat alle frequenties in amplitude worden doorgelaten en slechts in fase worden aangetast. Blijkbaar bestaat de verzameling van verliesvrije overdrachtsfuncties uit de all-pass filters. In termen van $H(z)$ kunnen we deze verzameling ook karakteriseren met $H(z)H(z^{-1}) = 1$. Eenvoudigste voorbeelden van verliesvrije overdrachtssystemen zijn

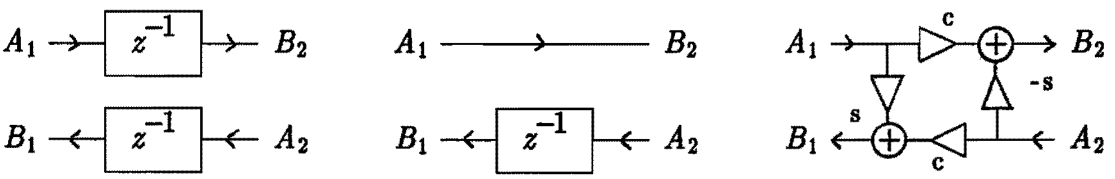
de tijdvertrager met $H(z) = z^{-1}$,
de doorverbinding met $H(z) = 1$ en
de inverter (de vermenigvuldiging met –1) met $H(z) = -1$.

Opm.: De functie $H(z)H(z^{-1})$ heet de *analytische voorzetting* of *uitbreiding* van $|H(e^{j\theta})|^2$ naar het hele z-vlak. Op de eenheidscirkel geldt de identiteit $H(z)H(z^{-1})\Big|_{z=e^{j\theta}} = H(e^{j\theta})H(e^{-j\theta}) = H(e^{j\theta})H^*(e^{j\theta}) = |H(e^{j\theta})|^2$.

Een digital two-pair is op dezelfde manier verliesvrij indien $S^+(e^{j\theta})S(e^{j\theta}) = \mathcal{I}$, voor alle θ. Hierin staat S^+ voor de *getransponeerde* en *geconjugeerde* matrix $(S^t)^*$, die ontstaat uit S door verwisseling van rijen en kolommen en conjugatie van de elementen. $\mathcal{I}$ is de eenheidsmatrix in $\mathbb{R}_2$, dus $\mathcal{I}=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$.

De eis $S^+S=\mathcal{I}$ volgt eenvoudig uit de voorwaarde voor verliesvrijheid $P_u=P_i$, dus $|B_1|^2+|B_2|^2=|A_1|^2+|A_2|^2$, ofwel $\underline{B}^+\underline{B} = \underline{A}^+\underline{A}$, met $\underline{B} = S\cdot\underline{A}$. In termen van $S(z)$ kunnen we ook schrijven $\boxed{S^t(z^{-1})S(z)= \mathcal{I}}$. S heet dan *para-unitair*.

Drie elementaire verliesvrije two-pairs met hun bijbehorende S-matrix zijn getekend in Fig. 3, de vertragingslijn, de semi-vertragingslijn en de *rotator*.



$$S = \begin{bmatrix} 0 & z^{-1} \\ z^{-1} & 0 \end{bmatrix} \qquad S = \begin{bmatrix} 0 & z^{-1} \\ 1 & 0 \end{bmatrix} \qquad S = \begin{bmatrix} \sin(\phi) & \cos(\phi) \\ \cos(\phi) & -\sin(\phi) \end{bmatrix}$$

Fig. 3 *Elementaire verliesvrije digital two-pairs*

Opm.: De rotator heeft een S-matrix met reële elementen, die daarmee voldoet aan de relatie $S^tS = \mathcal{I}$ Een reële unitaire matrix heet ook wel een orthogonale matrix. Vandaar de naam *orthogonale filters* voor structuren die zijn opgebouwd met verliesvrije one– en two–pairs. In ruimere zin vallen zowel de lattice– en ladderfilters als de wave digital filters onder deze klasse.

Een verliesvrije overdrachtsfunctie of all–pass functie wordt ook wel een LBR–functie genoemd. De afkorting staat voor *L*ossless *B*ounded *R*eal. Een *B*ounded *R*eal (BR–functie) is daarbij een *passieve* overdracht, waarvoor het uitgangsvermogen naar boven begrensd ('bounded') is door het ingangsvermogen. In geval van gelijke vermogens aan in– en uitgang voor een willekeurig ingangssignaal spreekt men van 'lossless' (verliesvrij). De aanduiding 'real' slaat op het reëel zijn van $H(z)$ voor reële z. $H(z)$ is een zgn. reële rationale functie, d.w.z. een quotiënt van twee polynomen in z met reële coëfficiënten.

Willen we digitale filters ontwerpen op de 'analoge' manier (Fig. 1), dan moeten we naast verliesvrije bouwblokken ook passieve afsluitingen kennen. De meest voor de hand liggende passieve afsluiting is daarbij een one–pair met B=0. Dit komt heel simpel neer op A_2=0 voor de two–pair die op deze manier passief wordt afgesloten. B_2 is dan het uitgangssignaal van het digitale filter (Fig. 2).

Het ontwerpen van digitale filters met verliesvrije en passieve bouwstenen is ook de meest effectieve manier om de storende invloed van niet–lineariteiten (kwantisatie en overflow correctie) in het filter te verminderen. Dit geldt zowel voor het onderdukken van limit cycles en overflow oscillaties als voor het verbeteren van de signaal/ruis–verhouding.

- In eerste benadering kunnen we het afronden van de *uitkomst* van een vermenigvuldiging modelleren als ruis op de *waarde* van de betrokken vermenigvuldiger. Een lage coëfficiënt–gevoeligheid gaat daardoor altijd gepaard met een verminderde kwantisatieruis aan de uitgang van het filter.
- Een voldoende voorwaarde voor niet–lineaire stabiliteit van een recursief digitaal filter is gegeven door de eis dat een geschikt gekozen *pseudo–energie* afneemt door de niet–lineaire operatie(s) en niet toeneemt door de lineaire bewerkingen in het filter. Met het voorgestelde filterontwerp voldoen we aan deze eis indien we voor de pseudo–energie het eerder gedefinieerde vermogen kiezen en het afronden in het filter vervangen door afbreken. Deze laatste manier van kwantiseren verkleint de absolute waarde van een signaal (2.6 → 2)

in tegenstelling tot het normale afronden (2.6 → 3). Limit cycles zijn daarmee uitgesloten, daar elk signaal dat in een lus (*loop*) van het filter rondloopt slechts energie kan verliezen (elke loop bevat onherroepelijk een kwantisator). Hetzelfde geldt voor overflow oscillaties, omdat elke vorm van overflow correctie energie–verminderend werkt.

LATTICE– EN LADDERFILTERS

We hebben nu alle middelen om een orthogonaal filter op te bouwen met elementaire verliesvrije two–pairs in cascade, zoals getekend in Fig. 4.

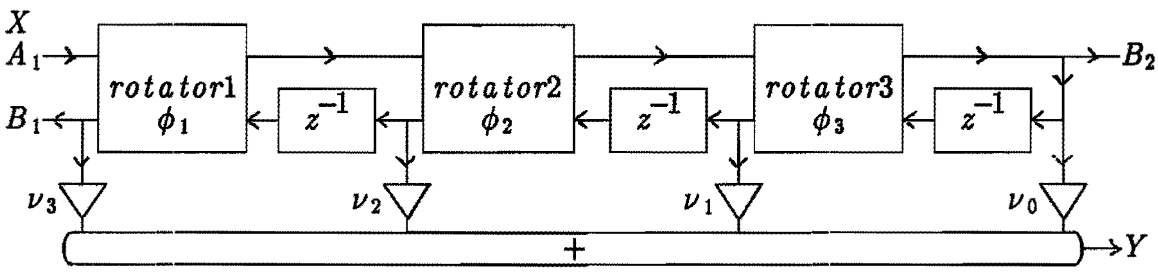


Fig. 4 *Orthogonaal ladder filter*

De elementen in de cascade zijn de semi–vertragingslijn en de rotator uit Fig. 3. Voor de laatste two–pair in de cascade geldt $A_2 = B_2$, zodat er sprake is van een verliesvrije afsluiting (een doorverbinding). De overdracht van A_1 naar B_1 zal daardoor een LBR–functie zijn en dus een all–pass karakter hebben. De overdracht $A_1 \rightarrow B_2$ heeft dezelfde polen als $A_1 \rightarrow B_1$, echter alle nulpunten in de oorsprong, zoals door berekening kan worden geverifieerd. Een willekeurige nulpunten–ligging en daarmee een algemene overdracht $X \rightarrow Y$ komt tot stand door de getekende lineaire combinatie van uitgangssignalen van de two–pairs.

Opm.: Door deze realisering van de nulpunten gaat er een deel van de lage coëfficiënt–gevoeligheid verloren, althans voor de nulpunten. Voor de meeste toepassingen van recursieve filters is echter een gunstige pool–gevoeligheid van groter belang. Als we FIR–filters willen ontwerpen – die zoals bekend slechts nulpunten hebben (en polen in de oorsprong) – dan kunnen we toch een lage coëfficiënt–gevoeligheid bereiken met de in Fig. 5 getekende structuur. Deze lijkt sterk op een omkering van Fig. 4, met dat verschil, dat de rotators nu met hun twee ingangen links en hun twee uitgangen rechts zijn getekend.

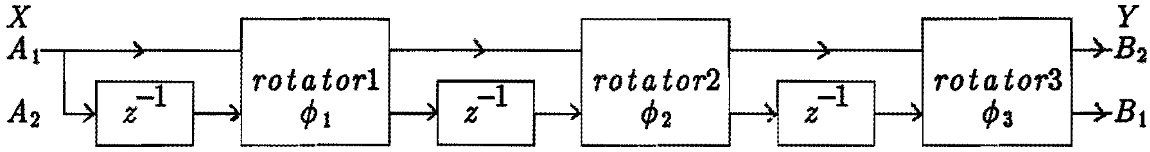
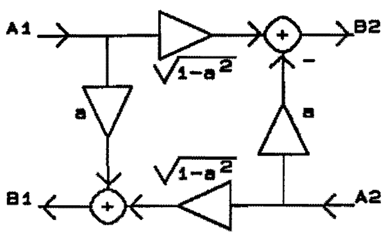


Fig. 5 *Orthogonaal FIR–filter*

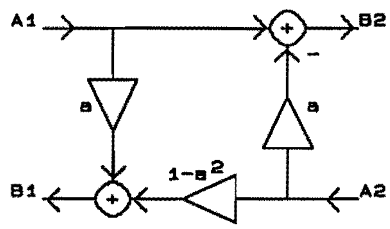
Het grote nadeel van deze realiseringen is, dat er veel te kwistig wordt omgesprongen met vermenigvuldigers. In Fig. 4 realiseren we slechts drie polen met de 12 vermenigvuldigers in de drie rotators. Hetzelfde geldt voor de drie nulpunten in Fig. 5. Wanneer we op zoek gaan naar orthogonale two–pairs met minder vermenigvuldigingen, komen we vanzelf uit op *lattice* structuren. In Fig. 6 is de overgang aangegeven via het zgn. *three–multiplier* ladder element naar de *two–multiplier* lattice en *one–multiplier* lattice elementen. De typische kruisende takken geven de beide laatste hun naam (ook wel tralie–element).



$$S = \begin{bmatrix} a & \sqrt{1-a^2} \\ \sqrt{1-a^2} & -a \end{bmatrix}$$

$$S^t S = \mathcal{I}$$

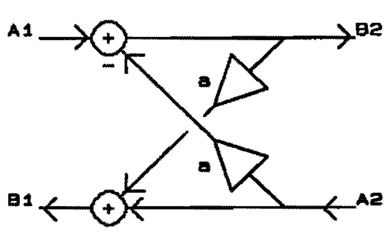
Fig. 6a *Rotator (four–multiplier ladder element)*



$$S = \begin{bmatrix} a & 1-a^2 \\ 1 & -a \end{bmatrix}$$

$$S^t DS = D \quad \text{met} \quad D = \begin{bmatrix} 1 & 0 \\ 0 & 1-a^2 \end{bmatrix}$$

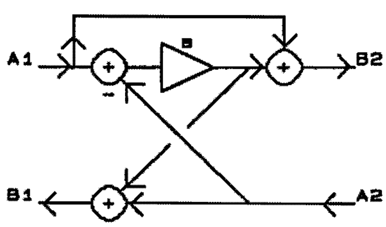
Fig. 6b *Three–multiplier ladder element*



$$S = \begin{bmatrix} a & 1-a^2 \\ 1 & -a \end{bmatrix}$$

$$S^t DS = D \quad \text{met} \quad D = \begin{bmatrix} 1 & 0 \\ 0 & 1-a^2 \end{bmatrix}$$

Fig. 6c *Two–multiplier lattice element*



$$S = \begin{bmatrix} a & 1-a \\ 1+a & -a \end{bmatrix}$$

$$S^t DS = D \quad \text{met} \quad D = \begin{bmatrix} 1+a & 0 \\ 0 & 1-a \end{bmatrix}$$

Fig. 6d *One–multiplier lattice element*

Uiteraard betalen we een prijs voor het besparen op vermenigvuldigers. De nieuwe elementen zijn namelijk niet zuiver verliesvrij, in die zin dat ze niet voldoen aan de relatie $S^tS=\mathcal{I}$. Wel geldt een verwante relatie $S^tDS=D$, zoals ook in Fig. 6 is aangegeven. Daarbij is D een *diagonale* matrix met *positieve* elementen (bedenk dat $|a| < 1$). Deze afwijking van het ideale geval heeft een toename van de coëfficiënt–gevoeligheid tot gevolg, die echter frequentie-onafhankelijk is en in de praktijk weinig problemen geeft.

Opm.: Het verschil tussen de condities $S^tS=\mathcal{I}$ en $S^tDS=D$ heeft alles te maken met *scaling*, d.w.z. het schalen of aanpassen van signaalniveau's in een digitaal filter om maximale uitsturing te bereiken met een gegeven (zeer kleine) kans op overflow. Dopen we $D^{\frac{1}{2}}SD^{-\frac{1}{2}}$ even S', dan is de relatie $S^tDS=D$ identiek met $(S')^tS'=\mathcal{I}$. Hierin zijn $D^{\frac{1}{2}}$ en $D^{-\frac{1}{2}}$ diagonale matrices met de wortels, resp. de reciproke waarden van de wortels, van de elementen van D op hun diagonaal. De two–pair S' verschilt van de two–pair S alleen in het ontbreken van twee *scalers* (lees: vermenigvuldigers) aan de beide ingangen ($\sqrt{D_{11}}$ en $\sqrt{D_{22}}$) en twee reciproke scalers aan de beide uitgangen.

De vier elementen in Fig. 6 zijn identiek wat betreft de overdracht $A_1 \rightarrow B_1$ als ze verliesvrij worden afgesloten, b.v. met een tijdvertrager: $B_2=z^{-1}A_2$. In alle vier de gevallen geldt dan

$$H(z) = \frac{B_1}{A_1} = \frac{az+1}{z+a}$$

d.w.z. een algemene eerste orde all–pass of LBR–functie, met $|H(e^{j\theta})| = 1$.

Een begrensde coëfficiënt $-1 < a < 1$ levert een stabiel filter. Ook in een samenstel van lattice secties hoeven we slechts elke coëfficiënt in absolute waarde kleiner dan 1 te kiezen om een stabiel filter te verkrijgen. Deze eenvoudige conditie maakt de lattice–filters aantrekkelijk voor toepassing in recursieve adaptieve filters, waar automatisch aanpassen van coëfficiënten niet tot instabiliteit mag leiden.

Bij een gegeven overdrachtsfunctie moeten in een lattice–filter nog de waarden van de vermenigvuldigers in de lattice elementen en de ν's in Fig. 4 worden bepaald, hetgeen niet triviaal is. Voor de lattice–filters bestaat er echter een eenvoudig algorithme om software–matig de coëfficiënten te bepalen, gebaseerd op het recursief afsplitsen (extractie) van eerste orde LBR–functies van de totale overdrachtsfunctie.

WAVE DIGITAL FILTERS

Bij de vertaling van het concept 'verliesvrijheid' naar digitale filters zijn we vrij geruisloos overgegaan van het p-domein naar het z-domein en zijn we tot de conclusie gekomen dat verliesvrije eenpoorten uit de analoge filters overeenkomen met LBR digital one-pairs. We kunnen deze overgang ook explicieter maken met de bilineaire transformatie en rechtstreeks de verliesvrije analoge elementen - zoals de spoel en de condensator - vertalen naar overeenkomstige digitale elementen. Het voordeel van deze werkwijze is, dat we dan meteen de volledige kennis omtrent het ontwerp van analoge filters ter beschikking hebben in het digitale domein. We kunnen dan b.v. een zesde orde elliptisch filter of een tiende orde Chebyshev filter opzoeken in een tabellenboek van analoge filters en de elementen één-op-één vertalen naar het overeenkomstige digitale filter.

Om tot een goeie vertaling te komen, keren we even terug naar de beschrijving van een two-pair met een matrix S. In de analoge filtertheorie is deze manier van beschrijven van een tweepoort reeds lang bekend. De matrix S heet dan de *scattering-* of *verstrooiïngsmatrix*, die voor een verliesvrije tweepoort unitair blijkt te zijn: $S^+S=\mathcal{I}$ De naamgeving voor deze matrix is afkomstig uit de microgolftheorie, waar de signaaloverdracht door een lange leiding of via een antenne-verbinding beschouwd wordt als een verstrooiïng van golven. Een naar rechts lopende golf A_1 - in b.v. een coax-kabel - wordt met een factor S_{21} doorgelaten en met een factor S_{11} gereflecteerd. Op dezelfde manier wordt een naar links lopende golf A_2 met een factor S_{12} doorgelaten en met een factor S_{22} gereflecteerd, zoals is aangegeven in Fig. 7, waarin $\underline{B}=S\underline{A}$.

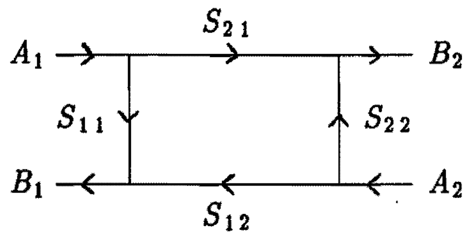


Fig. 7 *De verstrooiïngs- of scattering-matrix*

Formeel kunnen we de signalen in een digitaal filter als golven beschouwen met een richting en een meegevoerd vermogen $\frac{1}{2}|A|^2$ in een golf $a[n]=\mathrm{Re}(Ae^{j\theta n})$. Vandaar ook de naam *wave digital filters.*

In een analoog filter worden twee golven geassocieerd met een poort of klemmenpaar (spanning V, stroom I), en wel een golf $A=V+RI$ in de richting van de stroom aan de plus-klem en een golf $B=V-RI$ in de richting van de stroom aan de min-klem. Hierin is R een geschikt te kiezen *poortweerstand*, of aan de poort toegevoegd *kengetal*. Een en ander wordt duidelijk aan de hand van de golfvoorstelling van de vier belangrijkste analoge filter-elementen.

- Een spanningbron E met inwendige weerstand R levert een klemspanning $V=E-RI$, zodat $V+RI=E$. In een golfvoorstelling is een spanningsbron dus een *golfbron* met $A=E$, onafhankelijk van $B=V-RI$.

- Een weerstand R wordt aan de klemmen beschreven door $V=RI$, zodat $V-RI=0$. Een weerstand R is daarmee een *golfput* met $B=0$. Het vermogen van de inkomende golf A wordt volledig gedissipeerd en er wordt niets gereflecteerd. Definiëren we de reflectie-coëfficiënt ρ als de verhouding B/A, dan geldt
$$\rho = \frac{B}{A} = \frac{V-RI}{V+RI} = 0.$$

- Een spoel L wordt aan de klemmen beschreven door $V=j\omega L\cdot I$. Voor de modulus van de reflectie-coëfficiënt geldt dan
$$|\rho| = \left|\frac{B}{A}\right| = \left|\frac{V-LI}{V+LI}\right| = \left|\frac{j\omega - 1}{j\omega + 1}\right| = 1.$$
Voor het gemak hebben we L als kengetal genomen. Blijkbaar wordt het vermogen van de inkomende golf A volledig gereflecteerd.

- Een condensator C wordt aan de klemmen beschreven door $I=j\omega C\cdot V$. Voor de modulus van de reflectie-coëfficiënt geldt dan
$$|\rho| = \left|\frac{B}{A}\right| = \left|\frac{V-I/C}{V+I/C}\right| = \left|\frac{1 - j\omega}{1 + j\omega}\right| = 1.$$
Voor het gemak hebben we $1/C$ als kengetal genomen. Het vermogen van de inkomende golf A wordt weer volledig gereflecteerd.

De reflectie-coëfficiënt van de beide verliesvrije elementen L en C is dus - zoals het hoort - een LBR-functie van de Laplace-variabele p, met modulus 1 voor $p=j\omega$. We hebben nu echter expliciet een reflectie $(p-1)/(p+1)$ voor de spoel en $(1-p)/(1+p)$ voor de condensator. Met de bilineaire transformatie $p=(z-1)/(z+1)$, als vertaling van een LBR-functie in het p-domein naar een LBR-functie in het z-domein, volgt dan

$$B = -z^{-1}A \quad \text{voor de spoel} \quad \text{en}$$
$$B = z^{-1}A \quad \text{voor de condensator.}$$

Opm.: We hebben hier de conventie overgenomen dat p een op ω_0 genormeerde frequentie–variabele is en daardoor geen dimensie heeft. Hierin is ω_0 de afsnijfrequentie (in rad/s) van een prototype analoog laagdoorlaatfilter. Op dezelfde manier hebben ook de elementwaarden L en $1/C$ de dimensie van een weerstand, doordat spoelen en condensatoren worden gekarakteriseerd door de grote van hun impedantie bij frequentie ω_0. Na keuze van de afsnijfrequentie volgen de waarden van spoelen (in Henry) en condensatoren (in Farad) uit L/ω_0 resp. C/ω_0. Het analoge filter dat het uitgangspunt vormt voor het ontwerp van een wave digital filter heet het *referentie–filter.*

Opm.: De scattering–matrix van een analoge *tweepoort* beschrijft de verstrooiïng van de inkomende golven $A_1 = V_1 + I_1 R_1$ en $A_2 = V_2 + I_2 R_2$. Een verliesvrije tweepoort voldoet aan

$$S^+ \mathscr{R}^{-1} S = \mathscr{R}^{-1} \qquad \text{met} \quad \mathscr{R} = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}.$$

De reden waarom niet de relatie $S^+ S = \mathscr{I}$ geldt, is gelegen in het feit dat golven van het type $V+RI$ *spanningsgolven* zijn in tegenstelling tot *vermogensgolven* van het type $V/\sqrt{R} + \sqrt{R}I$. De keuze van golftype heeft geen invloed op reflectie–coëfficiënten (S_{11}, S_{22} en ρ), maar alleen op overdrachtscoëfficiënten (S_{21} en S_{12}). Dit effect zagen we al eerder optreden bij de lattice–filters (Fig. 6). Ook voor de wave digital filters geldt dat er wordt bespaard op het aantal vermenigvuldigers, wanneer we uitgaan van spanningsgolven. Het gevolg is ook nu weer een kleine frequentie–onafhankelijke toename van de coëfficiënt–gevoeligheid.

We hebben nu dus zeer simpele vertalingen van de analoge filter–elementen ter beschikking. Een spanningsbron drukt een ingangssignaal op, een weerstand onttrekt een uitgangssignaal, een spoel is een tijdvertrager met een inverter en een condensator is een tijdvertrager. De eenvoud van de overeenkomstige digitale elementen is in zoverre bedrieglijk, dat steeds een gunstige poortweerstand is gekozen, afhankelijk van de waarde van de analoge elementen. Op het moment dat we de digitale elementen met elkaar willen verbinden, vergelijkbaar met 'analoge' serie– en parallelschakeling, komen de poortweerstanden en daarmee de waarden van de analoge elementen in het spel in de vorm van vermenigvuldigers in de *adapters* die de verbindingen in een wave digital filter verzorgen. De eenvoudigste adapter is de two–pair adapter, die een verandering van poortweerstand $R_1 \to R_2$ kan realiseren met gebruikmaking van één vermenigvuldiger.

In termen van V en I zoeken we dan een tweepoort die de spanning behoudt ($V_2 = V_1$) en de stroom doorgeeft ($I_2 = -I_1$, waarin het min-teken afkomstig is van de conventie om de stroomrichting aan de plus-klemmen naar de tweepoort toe te kiezen). In de golfvoorstelling moeten we dan voldoen aan de vergelijkingen

$$\tfrac{1}{2}(A_2 + B_2) = \tfrac{1}{2}(A_1 + B_1)$$

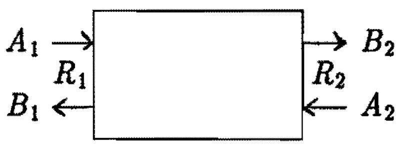
$$\frac{A_2 - B_2}{2R_2} = -\frac{A_1 - B_1}{2R_1}.$$

Om de verstrooiïngsmatrix te bepalen, drukken we B_1 en B_2 uit in A_1 en A_2.

$$B_1 = A_2 + \frac{R_2 - R_1}{R_2 + R_1} \cdot (A_1 - A_2)$$

$$B_2 = A_1 + \frac{R_2 - R_1}{R_2 + R_1} \cdot (A_1 - A_2)$$

In Fig. 8 is de two-pair adapter weergegeven met bijbehorende S-matrix.



$$S = \begin{bmatrix} a & 1-a \\ 1+a & -a \end{bmatrix} \qquad \text{met} \quad a = \frac{R_2 - R_1}{R_2 + R_1}$$

Fig. 8 *Two-pair adapter voor verandering van poortweerstand*

De two-pair adapter is dus niets anders dan het one-multiplier lattice element (Fig. 6d). Nemen we b.v. een condensator met poortweerstand $1/C$ als afsluiting (zie ook blz. 9), dan geldt $B_2 = z^{-1} A_2$, met $R_2 = 1/C$. De overdracht $A_1 \rightarrow B_1$ is dan

$$H(z) = \frac{B_1}{A_1} = \frac{az+1}{z+a} \qquad \text{met} \quad a = \frac{1 - R_1 C}{1 + R_1 C}.$$

Gezien de vrijheid van keuze van R_1 is dit een algemene eerste orde all-pass of LBR-functie met $|H(e^{j\theta})| = 1$.

In een analoog filter kunnen we elementen (of eenpoorten) in serie of parallel schakelen. In het overeenkomstige wave digital filter hebben we daartoe adapters nodig die algemeen n poorten met verschillende poortweerstanden $R_1 \ldots R_n$ met elkaar kunnen verbinden, de n-pair serie-adapter en de n-pair parallel-adapter. In termen van V en I betekent serieschakelen van n poorten:

$$V_1 + V_2 + V_3 + \ldots + V_n = 0$$

$$I_1 = I_2 = I_3 = \ldots = I_n .$$

De n-pair serie-adapter dient dan de volgende n relaties $\underline{B} = S_{serie}\underline{A}$ te realiseren.

$$B_i = A_i - \beta_i \cdot \sum_{k=1}^{n} A_k \quad (i=1...n) \qquad \text{waarin } \beta_i = \frac{2R_i}{R_0} \text{ met } R_0 = \sum_{k=1}^{n} R_k$$

Er zijn in totaal n vermenigvuldigers β_i nodig om de n uitgangssignalen B_i van de n-pair serie–adapter te berekenen uit de n ingangssignalen A_i. Merk op dat alle vermenigvuldigers positief zijn als het referentie–filter slechts passieve elementen bevat. Door gebruik te maken van de relatie $\Sigma\beta_i=2$ kunnen we nog één vermenigvuldiger besparen, door b.v. B_n te berekenen met $-\sum_{1}^{n} A_k - \sum_{1}^{n-1} B_k$.

In termen van V en I betekent parallelschakelen van n poorten:

$$V_1 = V_2 = V_3 = ... = V_n$$

$$I_1 + I_2 + I_3 + ... + I_n = 0.$$

De n-pair parallel–adapter dient dan n relaties $\underline{B}=S_{\text{parallel}}\underline{A}$ te realiseren.

$$B_i = \sum_{k=1}^{n} \alpha_k \cdot A_k - A_i \qquad \text{waarin } \alpha_i = \frac{2G_i}{G_0} \text{ met } G_0 = \sum_{k=1}^{n} G_k \text{ en } G_k = 1/R_k$$

Er zijn weer n vermenigvuldigers α_i nodig om de n uitgangssignalen B_i van de n-pair parallel–adapter te berekenen uit de n ingangssignalen A_i. Merk op dat ook deze vermenigvuldigers alle positief zijn als het referentie–filter slechts passieve elementen bevat. Door gebruik te maken van de relatie $\Sigma\alpha_i=2$ kunnen we ook nu nog één vermenigvuldiger besparen.

De algemene n-pair adapters gebruiken elk minimaal n–1 vermenigvuldigers en hebben dus n–1 vrijheidsgraden. In de uiteindelijke realisering heeft een wave digital filter hetzelfde totale aantal vrijheidsgraden als het analoge referentie–filter. In Fig. 9 zijn de symbolen van de beide n-pair adapters getekend.

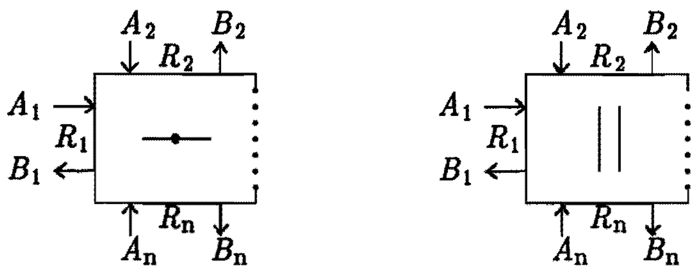


Fig. 9 *n-pair serie–adapter* *en* *n-pair parallel–adapter*

Opm.: We kunnen de adapters ook karakteriseren door een $n \times n$ matrix S. Met bovenstaande relaties vinden we dan voor S_{serie} en S_{parallel}:

S_{serie} $\quad S_{ii} = 1-\beta_i$ en $S_{ij} = -\beta_i$ voor $i \neq j$ $(i,j = 1...n)$

S_{parallel} $\quad S_{ii} = \alpha_i-1$ en $S_{ij} = \alpha_j$ voor $i \neq j$ $(i,j = 1...n)$.

De poortweerstanden van de adapters worden bepaald door de aangesloten elementen. In het in Fig. 10 getekende voorbeeld worden de poortweerstanden bepaald door de bronweerstand R_1, de afsluitweerstand R_2, de spoel'weerstand' L en de condensator'weerstand' $1/C$. Het gaat hier om een notchfilter met een overdrachtsnulpunt (een dempingspool) bij $\omega=1/\sqrt{LC}$. Een 3-pair parallel-adapter realiseert de parallelschakeling van de bron (met inwendige weerstand), de afsluitweerstand en de serieschakeling van L en C. Voor de serieschakeling hebben we nog een 3-pair serie-adapter nodig.

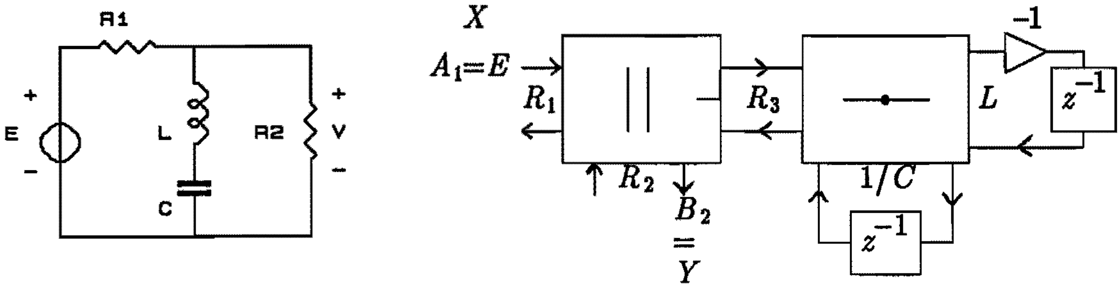


Fig. 10 *Voorbeeld van een wave digital filter met bijbehorend referentiefilter*

Bij de verbinding *tussen* adapters wordt de gemeenschappelijke poortweerstand niet bepaald door een element (R_3 in Fig. 10). Toch is deze niet vrij te kiezen. Bij het verbinden van adapters ontstaat er in het algemeen namelijk een *vertragingsvrije lus* of *delay-free loop*. Dit is een gesloten pad in het filter dat geen vertragingselement bevat. Een vertragingsvrije lus mag nooit voorkomen, omdat het onmogelijk is om een rekenvolgorde in de lus aan te geven. Slechts door geschikte keuze van de gemeenschappelijke poortweerstand kan bij het verbinden van adapters een vertragingsvrije lus worden vermeden. Bij één van de adapters mag dan de directe weg van ingangs- naar uitgangssignaal van de verbindende poort niet voorkomen. In Fig. 10 is gekozen voor de parallel-adapter, hetgeen is aangegeven met een dwarsstreepje bij de betrokken poort. Er moet gelden dat B_3 onafhankelijk is van A_3, ofwel $S_{\text{parallel}33} = \alpha_3-1 = 0$. Aan de eis $\alpha_3=1$ wordt voldaan met $G_3 = G_1+G_2$, of $1/R_3 = 1/R_1+1/R_2$.

Algemeen geldt voor een parallel-adapter dat wanneer één poort*geleiding* G_i gelijk is aan de som van de andere poortgeleidingen, dat dan S_{ii} gelijk is aan nul. De betrokken poort heet dan *reflectievrij* of *aangepast*. Voor een serie-adapter geldt hetzelfde met betrekking tot de poort*weerstanden*. In Fig. 10 hadden we dus ook $R_3=L+1/C$ kunnen kiezen om $S_{\text{serie}33}=1-\beta_3=0$ te realiseren. Aanpassing van een poort bespaart weer een vermenigvuldiger in de adapters, zodat een n-pair adapter met n-2 vermenigvuldigers kan worden gerealiseerd.

STATE–SPACE FILTERS

In deel I van de cursus is al kort aan de orde geweest dat een digitaal filter kan worden voorgesteld als een 'black box' met L poorten (ingangs–/uitgangs–klemmenparen) waarin algebraïsche bewerkingen worden uitgevoerd (optellingen en vermenigvuldigingen) en waaraan uitwendig L vertragingselementen zijn aangesloten. In Fig. 11 is deze *state–space* voorstelling nog een keer getekend, samen met een ingangssignaal $x[n]$ en een uitgangssignaal $y[n]$ tussen welke een gewenste overdracht $H(z)$ moet worden gerealiseerd.

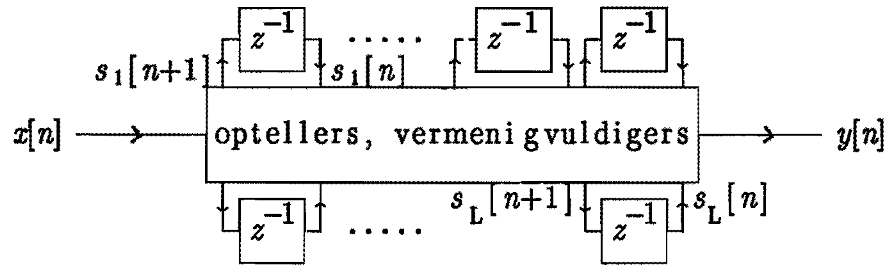


Fig. 11 *State–space voorstelling van een digitaal filter*

De *toestanden* of *states* $s_i[n]$ zijn de inhouden van de geheugen–elementen die op elk tijdstip n kunnen worden uitgelezen. De nieuwe *state–* of *toestandsvector* $\underline{s}[n+1]$ wordt op elk moment aan de ingangen van de tijdvertragers aangeboden. De dimensie L van de *toestandsruimte* of *state space* zegt daarbij iets over de orde van het filter. Met L geheugenplaatsen kunnen we algemeen een L–de orde overdrachtsfunctie realiseren.

De verandering van de state in de tijd wordt vastgelegd door de toestandsvergelijkingen

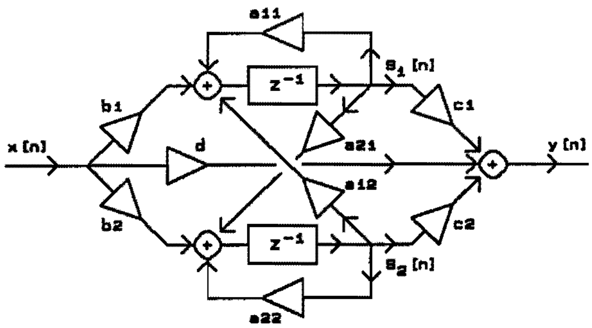
$$\underline{s}[n+1] = A\ \underline{s}[n] + \underline{b}\ x[n]$$

$$y[n] = \underline{c}^t\ \underline{s}[n] + d\ x[n].$$

Hierin is A een $L \times L$ matrix, $\underline{b}$ en $\underline{c}$ zijn (kolom–)vectoren met L elementen – en $\underline{c}^t$ dus een $1 \times L$ rijvector – en d is een scalar. De *toestandsmatrix* A – niet te verwarren met een golf – beschrijft hoe de nieuwe state afhangt van de huidige en bepaalt daarmee de stabiliteit van het systeem. De vector $\underline{b}$ geeft aan hoe de state mede wordt gestuurd door het ingangssignaal en $\underline{c}$ en d geven aan hoe het uitgangssignaal ontstaat door lineaire combinatie van state en ingangssignaal.

Opm.: Men zou de toestandsmatrix ook kunnen zien als de verstrooiïngsmatrix van een L–pair adapter die de interconnectie van L elementen realiseert.

Deze beschrijving in de toestandsruimte kan voor *elk* digitaal filter worden gegeven. We noemen een realisering van een gegeven $H(z)$ een ***state–space filter***, wanneer de toestandsvergelijkingen rechtstreeks zijn vertaald naar een filter–structuur. De elementen van A, $\underline{b}$ en $\underline{c}$ en de constante d komen daarbij alle voor als vermenigvuldigers in de realisering. In Fig. 12 is een tweede orde state–space filter weergegeven, waarin in totaal 9 vermenigvuldigers worden gebruikt. Bedenken we dat een algemene tweede orde overdrachtsfunctie slechts 5 vrijheidsgraden kent, dan hebben we blijkbaar 4 vrijheden over. Voor een L–de orde state–space filter hebben we $(L+1)^2$ vrijheidsgraden ter beschikking, waarvan er $2L+1$ nodig zijn voor de realisering van de gewenste overdracht.



$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad \underline{b} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \qquad \underline{c} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

Fig. 12 *Tweede orde state–space filter*

De resterende L^2 vrijheidsgraden hebben enerzijds tot gevolg dat er vele state–space filters zijn die dezelfde overdrachtsfunctie realiseren, anderzijds kunnen we deze vrijheid zo goed mogelijk benutten om ongewenste niet–lineaire effecten te verminderen. Daarbij beperkt men zich dan meestal tot een realisering met een cascade van tweede orde state–space secties, om het aantal vermenigvuldigingen niet onnodig groot te laten worden. Een L–de orde filter gebruikt dan $4L+1$ vermenigvuldigers.

Om na te gaan hoe de verschillende state–space filters met dezelfde $H(z)$ met elkaar samenhangen, bepalen we eerst hoe de overdrachtsfunctie volgt uit de toestandsvergelijkingen. Daartoe schrijven we deze in z–getransformeerde vorm

$$z\,\underline{S}(z) = A\,\underline{S}(z) + \underline{b}\,X(z)$$
$$Y(z) = \underline{c}^t\,\underline{S}(z) + d\,X(z).$$

Elimineren we de getransformeerde state–vector $\underline{S}(z)$ – niet te verwarren met de verstrooiïngsmatrix S – dan vinden we het verband tussen X en Y.

$$H(z) = \frac{Y(z)}{X(z)} = d + \underline{c}^t(z\mathcal{I}-A)^{-1}\underline{b}$$

Hierin komt de inverse van de matrix $z\mathcal{I}-A$ voor, waarin $\mathcal{I}$ de $L{\times}L$ eenheidsmatrix is. Het noemerpolynoom van $H(z)$ wordt daarmee gelijk aan de determinant $\det(z\mathcal{I}-A)$, zodat de *polen* van $H(z)$ gelijk zijn aan de *eigenwaarden* van A. De eigenwaarden van A moeten dus binnen de eenheidscirkel in het z-vlak liggen.

State–space filters die met elkaar samenhangen via een *state–transformatie* $\underline{s}=T\underline{s}'$ blijken nu dezelfde overdrachtsfunctie te realiseren. Hierin is T een *reguliere* matrix, d.w.z. een matrix waarvan de inverse bestaat. Wanneer we de relatie $\underline{s}=T\underline{s}'$ namelijk invullen in de originele toestandsvergelijkingen, dan zien we dat de parameters van het getransformeerde filter worden gegeven door

$$A' = T^{-1}AT \qquad \underline{b}' = T^{-1}\underline{b} \qquad \underline{c}' = T^{t}\underline{c} \qquad d' = d.$$

A' ontstaat uit A via een *gelijkvormigheidstransformatie*, die de eigenwaarden onveranderd laat. Voor het getransformeerde filter geldt dan ook $H'(z) = H(z)$.

Een geschikte state–transformatie kan nu dienen om een state–space filter met een gegeven $H(z)$ te optimaliseren met betrekking tot coëfficiënt–gevoeligheid, limit cycles en kwantisatieruis. We kunnen b.v. zodanig transformeren dat de L–pair met verstrooiïngsmatrix A *passief* wordt (zie Fig. 11 met weglating van de signalen $x[n]$ en $y[n]$). Verliesvrijheid van deze L–pair zou zijn gegeven door de conditie $A^tA=\mathcal{I}$. Overeenkomstig kunnen we de voorwaarde voor passiviteit verkort noteren als "$A^tA < \mathcal{I}$". Daarmee wordt dan bedoeld dat de matrix $\mathcal{I}-A^tA$ *positief definiet* is, hetgeen betekent dat $\underline{s}^t(\mathcal{I}-A^tA)\underline{s} > 0$ voor een *willekeurige* state $\underline{s}\neq\underline{0}$. Naast een lage coëfficiënt–gevoeligheid garandeert deze voorwaarde dat zowel limit cycles als overflow oscillaties niet kunnen optreden.

Opm.: De eis $\underline{s}^t(\mathcal{I}-A^tA)\underline{s} > 0$ zegt in feite dat het momentane vermogen van de huidige state $\underline{s}^t\underline{s}[n]$ groter is dan dat van de nieuwe state $\underline{s}^t\underline{s}[n+1]$. Kiezen we dit vermogen als de op pag. 6 genoemde pseudo–energie, dan hebben we de gewenste afname van de energie in het filter nadat $x[n]\equiv 0$ is gemaakt. Niet–lineaire stabiliteit wordt bereikt door kwantisatie (afbreken) en overflow correctie uit te voeren op de L toestandsvariabelen, d.w.z. *buiten* de black box. Kwantisatie binnen de black box is overbodig, omdat hierin geen lussen voorkomen. Deze zouden namelijk vertragingsvrij zijn. Het niet kunnen optreden van ongewenste oscillaties in een filter met ingangssignaal $x[n]\equiv 0$ wordt ook wel aangeduid met de term *zero–input stability*.

Anders geformuleerd kunnen we stellen dat een state–space filter *lineair* stabiel is als de eigenwaarden van de state–matrix A in modulus alle kleiner dan 1 zijn en dat een state–space filter *niet–lineair* stabiel is als *bovendien* de eigenwaarden van A^tA alle kleiner dan 1 zijn (de eigenwaarden van A^tA zijn positief reëel).

Bij gegeven eigenwaarden van A – de polen van $H(z)$ – is er altijd een matrix A te vinden waarvoor $A^tA < \mathcal{I}$. State–space filters die aan deze eis voldoen worden *minimum–norm filters* genoemd. Vergelijkbaar met het kwadraat van de norm van een vector $\|\underline{v}\|^2=\underline{v}^t\underline{v}$, is het kwadraat van de *matrix–norm* $\|A\|$ per definitie gelijk aan de grootste eigenwaarde van A^tA. Op grond van het bovenstaande geldt voor minimum–norm filters blijkbaar $\|A\| < 1$.
Het belangrijkste minimum–norm filter is een tweede orde state–space filter met polen $\alpha \pm j\beta$ en toestandsmatrix

$$A = \begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix} \qquad \text{zodat} \quad A^tA = (\alpha^2+\beta^2)\cdot\mathcal{I} = \det(A)\cdot\mathcal{I} < \mathcal{I}$$

Dit wordt een realisering in *normale vorm* genoemd, omdat de state–matrix een normale matrix is, gekarakteriseerd door $A^tA=AA^t$. Een directe vorm filter met $A = \begin{bmatrix} 2\alpha & 1 \\ -\alpha^2-\beta^2 & 0 \end{bmatrix}$ gaat over in de normale vorm via een state–transformatie

$$\underline{s} = T\underline{s}' \quad \text{met } T = \begin{bmatrix} 1 & 0 \\ -\alpha & \beta \end{bmatrix} \qquad \text{zodat} \quad A' = T^{-1}AT = \begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix}.$$

Naast de minimum–norm eis voor de toestandsmatrix kunnen we ook nog eisen opleggen aan de vectoren $\underline{b}$ en $\underline{c}$ in de state–space realisering. Deze eisen worden dan geformuleerd vanuit het oogpunt om de kwantisatieruis aan de uitgang van het filter te verminderen. Een state–space filter met een maximale signaal/ruis–verhouding heet een *minimum–noise filter*. Het blijkt dat de eis $\|A\| < 1$ een nodige voorwaarde is voor een minimum–noise realisering. Zonder afleiding zij vermeld dat een tweede orde minimum–noise filter voldoet aan drie condities.

$$a_{11} = a_{22}$$

$$b_1c_1 = b_2c_2$$

Gelijke en maximale uitsturing van de beide state–variabelen.

De laatste voorwaarde heeft te maken met *scaling*. Het schalen in een state–space filter is niets anders dan een state–transformatie met een diagonale matrix en vergt L vrijheidsgraden in een L–de orde filter. Aan bovenstaande voorwaarden kan precies worden voldaan met de 4 resterende vrijheden bij L=2.

Opm.: Kiezen we een positieve diagonale matrix D en passen we scaling toe met $\underline{s}=D^{\frac{1}{2}}\underline{s}'$, dan wordt de minimum-norm voorwaarde $A^tA < \mathcal{I}$ getransformeerd in $A'^tDA' < D$. Deze laatste voorwaarde is ook voldoende om niet-lineaire stabiliteit te garanderen. We kunnen nl. als geschikte pseudo-energie het getransformeerde vermogen $\underline{s}'^tD\underline{s}'$ gebruiken. Voor tweede orde state-space filters kunnen we de conditie $A^tDA < D$ ook expliciet uitdrukken in een voorwaarde waaraan de elementen van de state-matrix moet voldoen. Er geldt dan de volgende stelling.

Een tweede orde state-space filter is niet-lineair (zero-input) stabiel met elke vorm van overflow correctie en kwantisatie in de vorm van afbreken op de beide state-variabelen als de elementen van de toestandsmatrix voldoen aan

$$|a_{11}-a_{22}| < 1-\det(A).$$

LITERATUUR

Algemeen

[1] P.P. Vaidyanathan and S.K. Mitra, "Low passband sensitivity digital filters: A generalized viewpoint and synthesis procedures," *Proc. IEEE*, vol. 72, pp. 404–423, 1984.
[2] P.P. Vaidyanathan, "A unified approach to orthogonal digital filters and wave digital filters, based on LBR two-pair extraction," *IEEE Trans. Circuits & Syst.*, vol. CAS-32, pp. 673–686, 1985.
[3] H.J. Butterweck, J.H.F. Ritzerfeld and M.J. Werter, "Finite wordlength effects in digital filters," *Arch. Elektron. & Übertragungstech.*, vol. AEÜ-43, pp. 76–89, 1989.

Lattice and Ladder Filters, Orthogonal Filters

[4] A.H. Gray, Jr. and J.D. Markel, "Digital lattice and ladder filter synthesis," *IEEE Trans. Audio & Elektroacoust.*, vol. AU-21, pp. 491–500, 1973.
[5] E. Deprettere and P. Dewilde, "Orthogonal cascade realization of real multiport digital filters," *Int. J. Circuit Theory & Appl.*, vol. 8, pp. 245–272, 1980.

Wave Digital Filters

[6] A. Fettweis, "Digital filters related to classical filter networks," *Arch. Elektron. & Übertragungstech.*, vol. AEÜ-25, pp.79–89, 1971.
[7] A. Fettweis, "Wave digital filters: Theory and practice," *Proc. IEEE*, vol. 74, pp. 270–327, 1986.

State–Space Filters

[8] C.T. Mullis and R.A. Roberts, "Synthesis of minimum roundoff noise fixed point digital filters," *IEEE Trans. Circuits & Syst.*, vol. CAS–23, pp. 551–562, 1976.
[9] C.W. Barnes and A.T. Fam, "Minimum norm recursive digital filters that are free of overflow limit cycles," *IEEE Trans. Circuits & Syst.*, vol. CAS–24, pp. 569–574, 1977.
[10] W.L. Mills, C.T. Mullis and R.A. Roberts, "Digital filter realizations without overflow oscillations," *IEEE Trans. Acoust., Speech & Signal Process.*, vol. ASSP–26, pp. 334–338, 1978.
[11] L.B. Jackson, A.G. Lindgren and Y. Kim, "Optimal synthesis of second–order state–space structures for digital filters," *IEEE Trans. Circuits & Syst.*, vol. CAS–26, pp.149–153, 1979.
[12] J.H.F. Ritzerfeld, "A condition for the overflow stability of second–order digital filters that is satisfied by all scaled state–space structures using saturation," *IEEE Trans. Circuits & Syst.*, vol. CAS–36, pp. 1049–1057, 1989.

Overige Speciale Structuren

[13] R.C. Agarwal and C.S. Burrus, "New recursive digital filter structures having very low sensitivity and roundoff noise," *IEEE Trans. Circuits & Syst.*, vol. CAS–22, pp. 921–927, 1975.
[14] L.T. Bruton, "Low sensitivity digital ladder filters," *IEEE Trans. Circuits & Syst.*, vol. CAS–22, pp. 168–176, 1975.
[15] T.L. Chang, "On low roundoff noise and low sensitivity digital filter structures," *IEEE Trans. Acoust., Speech & Signal Process.*, vol. ASSP–29, pp. 1077–1080, 1981.
[16] R.E. Crochière, "Digital ladder structures and coefficient sensitivity," *IEEE Trans. Audio & Elektroacoust.*, vol. AU–20, pp. 240–246, 1972.
[17] D.C. Munson, Jr. and B. Liu, "Low–noise realizations for narrow–band recursive digital filters," *IEEE Trans. Acoust., Speech & Signal Process.*, vol. ASSP–28, pp. 41–54, 1980.

DIGITALE SIGNAALBEWERKING

deel II.3

Digitaal/Analoog- en Analoog/Digitaal omzetting

door

M. van der Veen

Philips I & E Breda

DIGITAAL/ANALOOG- EN ANALOOG/DIGITAAL OMZETTING

Ir. M. van der Veen   Philips Natuurkundig Laboratorium

INHOUD

1 INLEIDING

Binnen een cursus tijddiscrete signaalbewerking is enige aandacht aan Analoog / Digitaal- en Digitaal / Analoog omzetters (A/D- en D/A omzetters, A/D- en D/A converters, ADC en DAC) voor de hand liggend. De natuur gedraagt zich veelal analoog (fysische grootheden zijn meestal continu in tijd en amplitude), zodat we verplicht zijn om deze signalen te digitaliseren, willen we tijddiscrete signaalbewerkingen kunnen loslaten op deze informatiedragers. Zeker binnen de elektrotechniek zien we een toenemende digitalisering van veel (voorheen analoge) functies. Ook het inburgeren van het werken met microcomputers heeft het gebruik van A/D en D/A omzetters sterk in positieve zin beinvloed. We zien heden ten dage digitale signaalbe-werkingen bij: audio (compact disc), video (digitalisering van beeld en geluid), radar, sonar, telecommunicatie (digitale telefooncentrale, sateliet), medische diagnostiek, procesbesturingen en ga zo maar door. In welk van die velden u zich ook gaat begeven, als u een stukje digitalisering in wilt voeren is het gebruik van A/D- en D/A-converters niet meer weg te denken. De producenten van electronische componenten zijn volop ingesprongen op deze toegenomen vraag. Er is een enorm scala van A/D- en D/A-converters te koop, elk met zijn eigen mogelijkheden en beperkingen.

Het is de bedoeling van dit onderdeel van de cursus om de idealiteit van A/D en D/A converters, zoals die in voorafgaande delen van de cursus wordt gebruikt enigszins te ontkrachten. Dit gebeurt door in dit gedeelte enig inzicht te geven in problemen en moeilijkheden die kunnen ontstaan bij het toepassen van deze bouwstenen.

Allereerst wordt er een stukje theorie over bemonsteren en filteren gegeven, gevolgd door een overzicht van diverse configuraties van A/D- en D/A-converters. Het maken van een weloverwogen keus voor uw specifieke toepassing wordt voorbereid door dieper in te gaan op de diverse specificaties die van een converter kunnen worden gegeven. Ook het "hervormen" van het spectrum van de quantisatieruis (noise shaping) zal een bescheiden plaats krijgen.

Het is geenszins de bedoeling dat de hier gegeven presentatie een volledig overzicht geeft van de omzettings problematiek omdat het gebied hiervan veel te uitgebreid is. Er is geprobeerd om voor diegenen die dieper op de materie willen ingaan een tamelijk uitgebreide literatuurverwijzing bij te voegen.

2 EFFECTEN VAN BEMONSTEREN

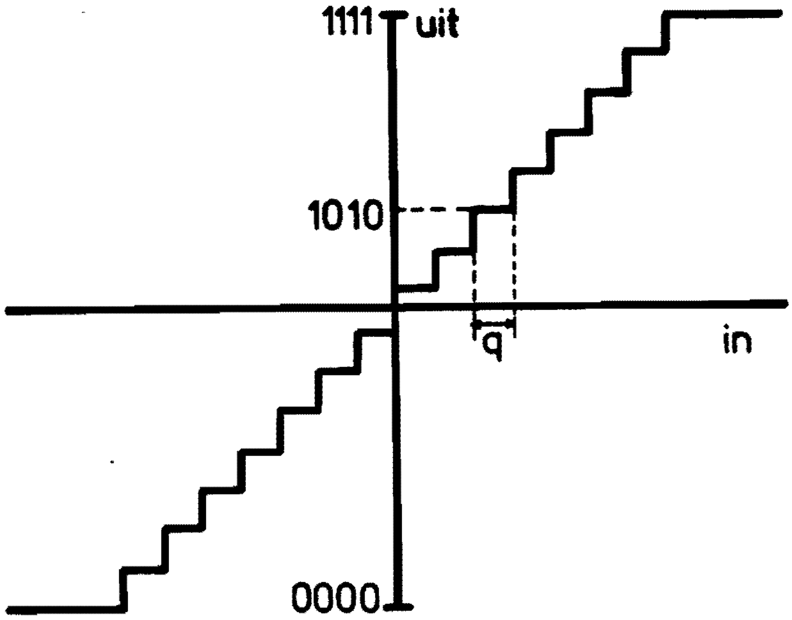
Als we spreken van systemen waarin A/D en D/A conversie voorkomt, dan zijn er signaalvormen zowel continu als gequantiseerd in tijd en amplitude in het susteem aanwezig.

De meestvoorkomende vorm in de natuur is die van continue amplitude en continue tijd. Alle electronische analoge schakelingen verwerken dit type signalen. Systemen waarin een continue amplitude en een discrete tijd voorkomen zijn bv. sample-

and-hold schakelingen. Hierin wordt op vaste tijdstippen (discreet) de amplitude vastgehouden, die elke waarde kan aannemen (continu). Gaan we de amplitude nu ook discretiseren, en weergeven met een digitaal woord, dan zijn we aangekomen bij digitale signalen.
Bij quantiseren voegen we dus inherent zelf afwijkingen van het oorspronkelijke signaal toe. Deze afwijkingen noemen we quantisatieruis.

* quantisering van de amplitude

Allereerst zullen we een schatting maken van het vermogen van het foutsignaal dat door amplitude-quantisatie aan het signaal wordt toegevoegd. Uitgaande van een uniforme quantisatie-karakteristiek (figuur 1) met N quantisatie-niveaus (N=2, voor een binaire representatie van B bits) en een quantisatie-eenheid q dan, zal een



Figuur 1: Quantisatie-karakteristiek voor uniforme quantisatie met N=16 (B=4).

signaalamplitude x worden gerepresenteerd door het [x/q]-de quantisatie niveau. Hier is [.] de afronding op een geheel getal.
Als [x/q] groter is dan N dan zal er begrenzingsvervorming gaan optreden omdat deze waarden van x door het N-de niveau gerepresenteerd worden. We veronderstellen nu dat een signaal amplitude die ligt in het gebied:

$$(i - 1/2)q < x < (i + 1/2)q$$

wordt gerepresenteerd door de gequantiseerde waarde x[i]=iq. Dus de instantane fout is $(x - x[i])$. Veronderstel verder dat de waarschijnlijkheids-dichtheid van de amplitude-verdeling gegeven wordt door p(x)dx. Als nu q klein is in vergelijking met de veranderingen in de signaalamplitudes dan zal p(x) weinig varieren binnen een quantisatie-interval. Het quantisatie-foutvermogen veroorzaakt door signalen welke binnen het interval i liggen wordt dan gegeven door:

$$E_i^2 = \int_{(i-1/2)q}^{(i+1/2)q} (x - iq)^2 p(x)dx \approx p(x[i]) \int_{-q/2}^{q/2} z^2 dz = (1/12)p(x[i])q^3$$

De waarschijnlijkheid dat de signaalamplitude in het interval i valt is gelijk aan:

$$p(x[i]) = \int_{(i-1/2)q}^{(i+1/2)q} p(x)dx \approx p_i q$$

Substitutie van dit resultaat in de vergelijking voor E_i^2 levert dan:

$$E_i^2 = (1/12)p_i q^2$$

De totale gekwadrateerde fout is gelijk aan de som van de resultaten van elk der intervallen i en volgt dus uit de vergelijking voor E_i^2:

$$E^2 = \sum_{i=1}^{N} E_i^2 = (1/12)\sum_{i=1}^{N} p_i q^2 = q^2/12$$

De laatste stap in deze vergelijking volgt uit het feit dat de som van alle p_i per definitie gelijk is aan 1 zolang de amplitude verdeling binnen de schaal van de quantisatiekarakteristiek valt. Het resultaat dat het foutvermogen ten gevolge van quantisatie gelijk is aan $q^2/12$, wordt vaak gebruikt om een signaal/ruis verhouding te bepalen voor het quantisatie-proces. We bekijken nu wat het resultaat is van het quantiseren van een sinusvormig signaal met amplitude A. Verondersteld wordt dat er geen begrenzingsvervorming optreedt dus dat geldt: $A < 2^N \cdot q/2$, met N= aantal bits. De amplitude A wordt dan gerepresenteerd door een quantisatie niveau k zodanig dat [A/q]=k en het signaal vermogen is dan $A^2/2 \approx (kq)^2/2$, waarbij de benadering geldt zolang A >> q. De signaal/ruis verhouding of beter gezegd de signaal/quantisatievervorming verhouding (SQVV) is dan gelijk aan:

$$SQVV = 10log((k^2q^2/2)/(q^2/12)) = 20log(k) + 7.78(dB)$$

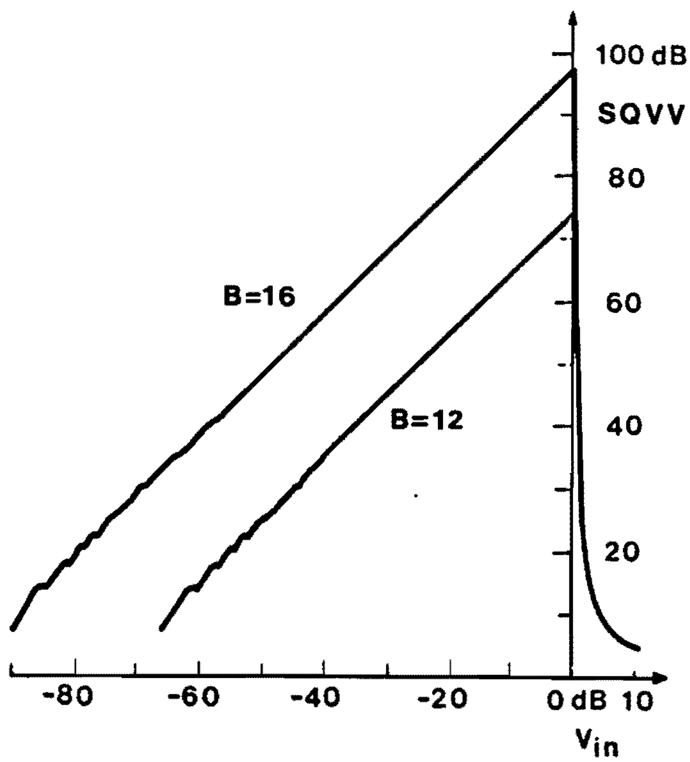
Is de quantisatie-karakteristiek zodanig uitgestuurd dat er nog juist geen begrenzingsvervorming optreedt, dus $k = 2^N$ dan vinden we voor de

$$SQVV = 20log(2^N) + 7.78 \approx 6.N + 1.76(dB)$$

Dit is de maximaal haalbare SQVV.
Treedt er wel begrenzingsvervorming op dan geldt:

$$E^2 = q^2/12 + \int_{-\infty}^{-V} (x + 2^N q/2)^2 p(x)dx + \int_{\infty}^{V} (x - 2^N q/2)^2 p(x)dx$$

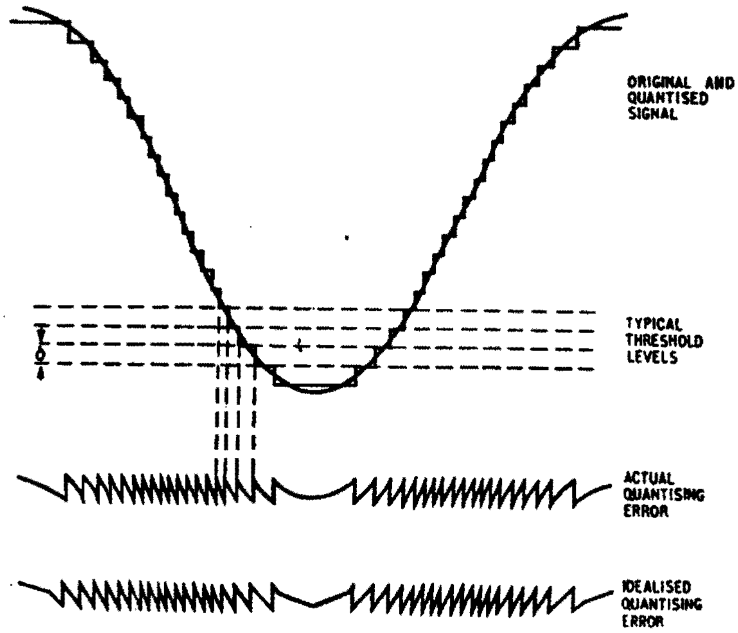
In figuur 2 is de SQVV als functie van het niveau van het sinusvormige signaal gegeven voor de quantisatie karakteristiek van figuur 1. De onderste kromme geldt voor een representatie met 12 bits per signaalmonster de bovenste voor een 16



Figuur 2: SQVV als functie van het niveau van een sinusvormig signaal.

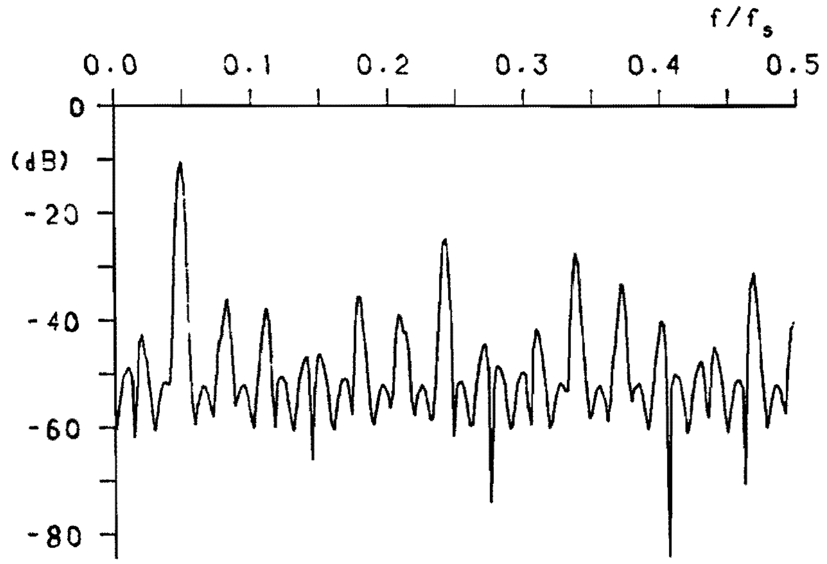
bits representatie. Het 0 dB niveau is zodanig gekozen dat er juist geen begrenzingsvervorming optreedt. Dit geldt bv. voor een sinusvormige uitsturing. Als de kansverdeling van het ingangssignaal echter bv. een gaussische verdeling heeft, dan is er altijd wel een stukje signaal te bedenken dat buiten het bereik van de converter valt. Voor dit type signaal treedt begrenzingsvervorming op. Het is dan belangrijk om het instelpunt van het gebruik van de converter zodanig te kiezen dat de afrondingsvervormingen een dominante rol spelen en niet de begrenzingsvervormingen.

Zoals uit figuur 2 kan worden gezien, zijn de benaderingen welke gebruikt zijn in de afleiding van het quantisatie-foutvermogen niet meer correct voor lage waarden van de amplitudes. Bij kleine amplitudes zal de waarschijnlijkheids-dichtheid functie van de amplitude-verdeling niet langer constant zijn over de quantisatie-intervallen. Als gevolg hiervan hangt de SQVV sterk af van de amplitude van de sinus. Uit figuur 3 blijkt ook dat het foutsignaal zich periodiek gaat herhalen met de periode van de sinus waardoor de afrondings-vervorming zich gaat manifesteren als een harmonische vervorming. Vallen deze vervormings- produkten buiten de band van 0 tot $f_s/2$ dan zullen ze t.g.v. het effect wat beschreven wordt door het theorema van Nyquist in deze band terugvouwen en aanleiding geven tot niet met het signaal gerelateerde frequenties. Een voorbeeld hiervan is het spectrum van een gequantiseerde sinus met amplitude q zoals getoond in figuur 4. Ter vergelijking is in figuur 5 het spectrum van een gequantiseerde sinus met amplitude 127q afgebeeld. Het is van belang dit soort vervormingen te onderscheiden van de ver-



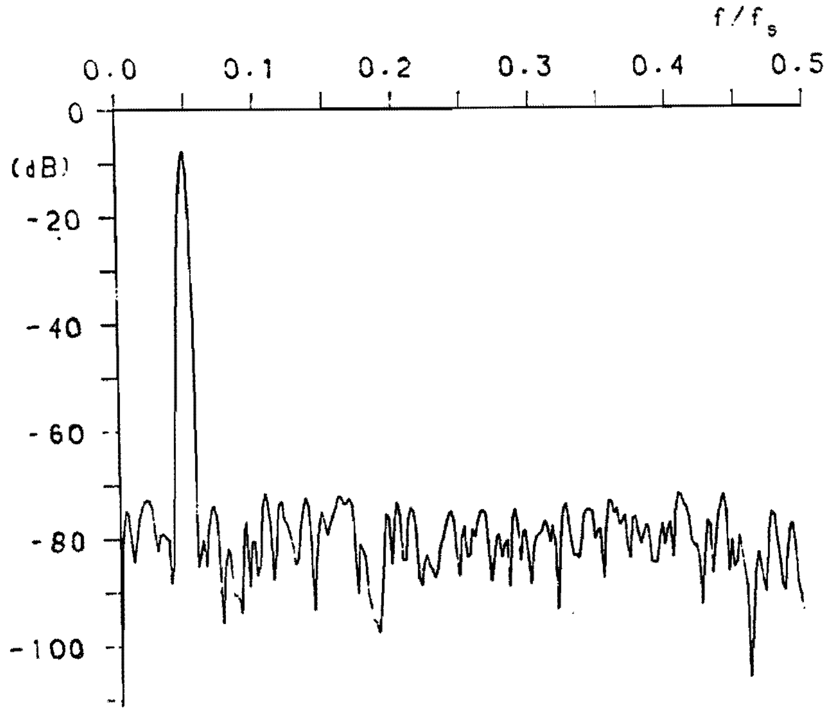
Figuur 3: Uniforme quantisatie van een sinusvormig signaal.

vormingen veroorzaakt door niet-lineairiteiten van de ADC omdat ze het gevolg zijn van quantisatie en zelfs bij een ideale ADC optreden.



Figuur 4: Spectrum van gequantiseerde sinus met amplitude q.

We merken hier tevens op dat quantisatie-vervorming ook belangrijk is in het geval dat D/A omzetters getest worden m.b.v. digitaal gegenereerde signalen omdat ook dan een ideale DAC aan de uitgang een spectrum vertoont zoals dat van figuur 4. Ook hier dient dit onderkend te worden om een verkeerde inschatting van de

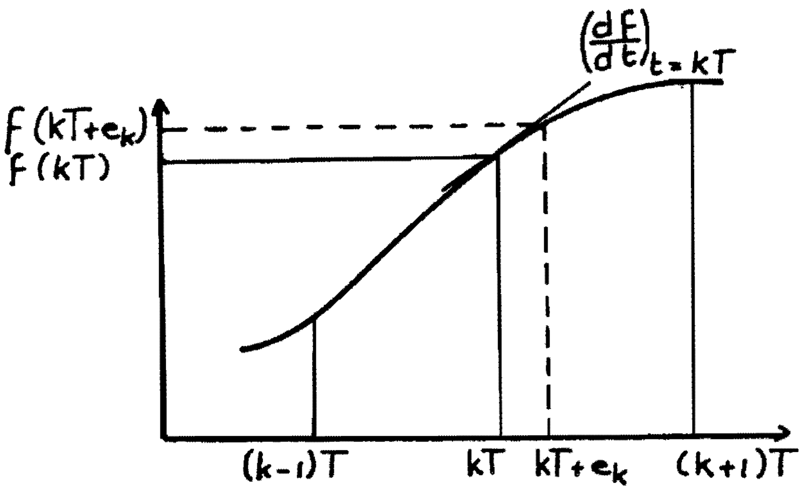


Figuur 5: Spectrum van een gequantiseerde sinus met amplitude 127 q.

kwaliteit van de omzetter te vermijden.

* quantisering van de tijd

Bij het omzetten van analoge d.w.z. tijd- en amplitude-continue signalen naar digitale d.w.z. tijd- en amplitude-discrete signalen is niet alleen het proces van



Figuur 6: Fouten in de bemonsteringstijdstippen.

amplitude-discretisatie (via A/D omzetting) van belang maar dienen we ook het tijd-discreet maken van het signaal te beschouwen. Het tijd-discreet maken van een signaal geschied door periodiek monsters van dit signaal te nemen, gewoonlijk op equidistante tijdstippen.
Bekijken we het bemonsterproces dan zien we dat een fout in het tijdstip waarop er een signaalmonster genomen wordt, vertaald kan worden naar een fout in de amplitude (zie figuur 6). We meten namelijk f(kT+e[k]) op het tijdstip kT+e[k] terwijl we f(kT) behoren te meten wanneer we op het juiste tijdstip kT bemonsteren. Het verschil tussen beide waarden is:

$$f(kT + e[k]) - f(kT) \approx e[k](df/dt)_{t=kT}$$

We eisen nu dat de gemaakte amplitudefout kleiner is dan 1/2 LSB van de ADC. Voor een n-bits ADC met een maximale ingangsspanning van +V volt is een 1/2 LSB gelijk aan V . 2^{-n}. Wanneer aan de bemonsteraar een sinusvormig signaal met een amplitude van V volt wordt aangeboden dan is de gemaakte amplitudefout:

$$f(kT + e[k]) - f(kT) \approx e[k].V.2\pi.f.cos(2\pi fkT) < +/- V.2^{-n}$$

Hieruit volgt dat de maximale tijdsfout e[k] kleiner moet zijn dan:

$$e[k] < 2^{-n}/(2\pi f)$$

om ervoor te zorgen dat de amplitudefout binnen de gestelde grenzen blijft. Voor een 16 bits ADC en ingangssignalen met frequenties tot maximaal 20 kHz (eisen bij digitaliseren van audiosignalen) is er dus een bemonsteraar vereist waarvan de bemonstertijdstippen binnen 125 picoseconde vastliggen. Een zelfde nauwkeurigheidseis (121 picoseconde) geldt voor 8 bits ADC met ingangsfrequenties tot maximaal 5 MHz (digitalisering van videosignalen) Het is dus duidelijk dat de kloksignalen welke de bemonstertijdstippen vastleggen in deze gevallen gegenereerd moeten worden m.b.v. nauwkeurige oscilatoren.

* bandbreedte begrenzing bij A/D's

Willen we het oorspronkelijke signaal getrouw reconstrueren uit een bemonsterde versie hiervan dan moet dit signaal in frequentie begrensd zijn (bemonsteringstheorema van Nyquist). Deze frequentieband-begrenzing wordt gerealiseerd door het te bemonsteren signaal eerst door een laagdoorlaat-filter te sturen zodat er geen frequentiecomponenten in het signaal aanwezig zijn boven f = $f_s/2$. De sperdemping van dit laagdoorlaat filter moet zo groot zijn dat de vervorming ten gevolge van het terugvouwen van de verzwakte frequenties boven $f_s/2$, kleiner is dan de vervorming veroorzaakt door het amplitude discretiseren (quantisatie). De vervorming t.g.v. quantisatie wordt meestal beschreven m.b.v. een witte ruis met een vermogen van $q^2/12$,zoals hiervoor is afgeleid.

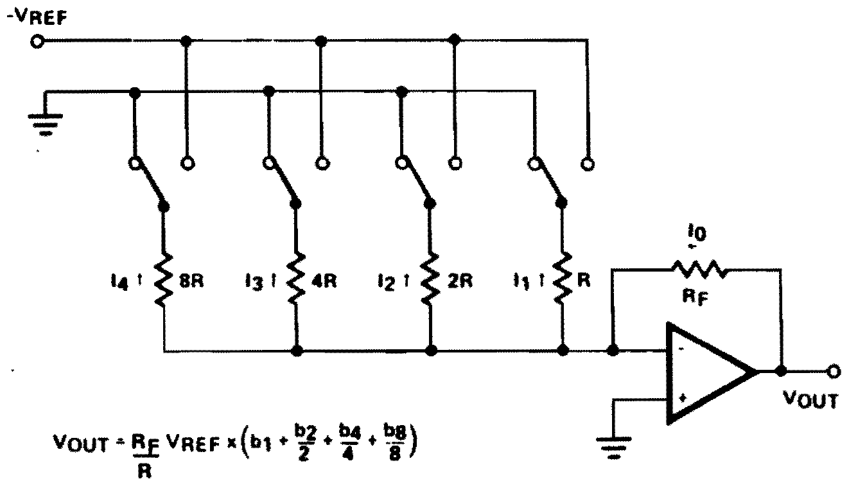
* bandbreedte begrenzing bij D/A's

Na D/A omzetting is voor de reconstructie van het analoge signaal een laagdoorlaat filter nodig. Dit laagdoorlaat filter fungeert als interpolatie filter voor het tijd-discrete signaal dat uit de D/A omzetter komt. De eisen welke aan dit filter gesteld worden hangen sterk af van het totale systeem waarin de DAC zich bevindt. Het filter moet zodanig ontworpen worden dat de schakelingen na de DAC geen last hebben van de spectrale herhalingen van de basisband ($0 < f < f_s/2$) op veelvouden van f_s. We merken hier ook op dat aan de uitgang van de DAC een houd-effect optreedt waardoor er reeds een bepaalde filterwerking (sinx/x) wordt gerealiseerd. In sommige gevallen kan het nodig zijn om voor dit houd-effect een compensatie in te bouwen. Dit kan zowel analoog als digitaal gebeuren. We gaan hier niet verder op in.

Een laatste punt waar we de aandacht op willen vestigen is de vervorming van laagdoorlaat filters. Zelfs bij gebruik van passieve LC filters treedt er harmonische vervorming op aan de uitgang van het filter als gevolg van niet-lineairiteiten van de gebruikte spoelen. Afhankelijk van de gewenste signaal/vervormings verhouding zal de grootte van deze vervorming gecontroleerd moeten worden.

3 Digitaal/Analoog Omzetting, enige basis principes

Digitaal/Analoog omzetters worden gebruikt voor het omzetten van een digitale representatie (een getal) in een analoge stroom of spanning. In het algemeen is deze stroom of spanning recht evenredig met de grootte van het digitale getal. Een eenvoudige schakeling om binaire getallen in een aan dit getal evenredige stroom of spanning om te zetten wordt getoond in figuur 7. De standen van de schakelaars



Figuur 7: Digitaal-Analoog omzetter met binair gewogen weerstanden.

worden bepaald door het optreden van een logische "nul" of een logische "een" voor het desbetreffende bit. Als de schakelaar aan de referentie geschakeld wordt zal er een stroom gaan lopen die bepaald wordt door de grootte van de bijbehorende weerstand. Door de tegenkoppeling via de weerstand R_f blijft het sommatiepunt van de operationele versterker op de nulpotentiaal. De uitgangsspanning van de omzetter wordt nu gegeven door:

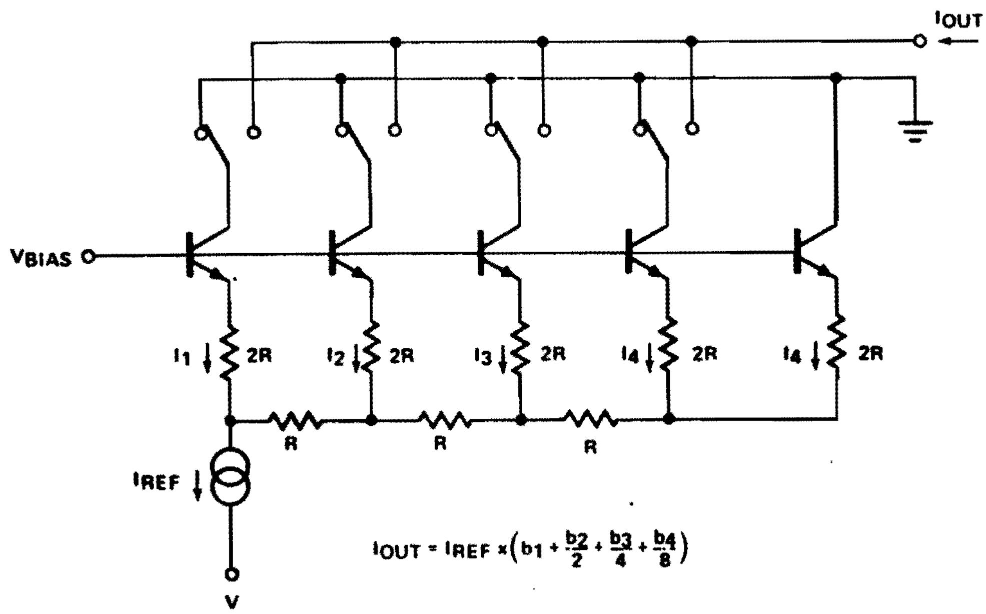
$$V_{out} = (R_f/8R)V_{ref}(8b_1 + 4b_2 + 2b_3 + b_4)$$

waarbij $b_i = 0$ of 1. Als $b_i = 0$ dan loopt er geen stroom door de betreffende weerstand, dit is wel het geval als $b_i = 1$. De nadelen van de schakeling van figuur 7 zijn:

i) de benodigde weerstanden in het weeg netwerk worden steeds een factor 2 groter (of kleiner) en dit heeft tot gevolg dat voor een n bits omzetter de grootste en kleinste weerstand in het netwerk een factor 2^n uit elkaar liggen. Hierbij geldt ook nog dat de onnauwkeurigheid van de grootste weerstand bepaald wordt door de helft van de kleinste weerstand in het netwerk en is gelijk aan 2^{-n-1}. Voor een 16 bits omzetter is dit een nauwkeurigheid van 7.5 10^{-6}. Dit is met de huidige integratie-technieken onhaalbaar voor omzetters met $N > 6$.

ii) Parasitaire capaciteiten in de knooppunten worden tijdens de omzetting geladen of ontladen waardoor ongewenste schakelverschijnselen optreden aan de uitgang. Dergelijke verschijnselen worden "glitches" genoemd en hun aanwezigheid beinvloedt de kwaliteit van het uitgangssignaal sterk. Wij komen hier later op terug. Door in plaats van spanningen stromen te schakelen kunnen deze "glitches" verminderd worden. Dit wordt gerealiseerd door de schakelaars tussen de weerstanden en het sommatiepunt van de operationele versterker te plaatsen. Een andere mogelijkheid is om de referentie spanning te vervangen door een referentie stroom.

Voor integratie wordt het nauwkeurigheidsprobleem omzeilt door te kiezen voor een zogenaamd R-2R laddernetwerk. Figuur 8 laat zien hoe het principe schema van een 4 bits DAC eruit ziet bij het toepassen van een dergelijk netwerk, waarbij

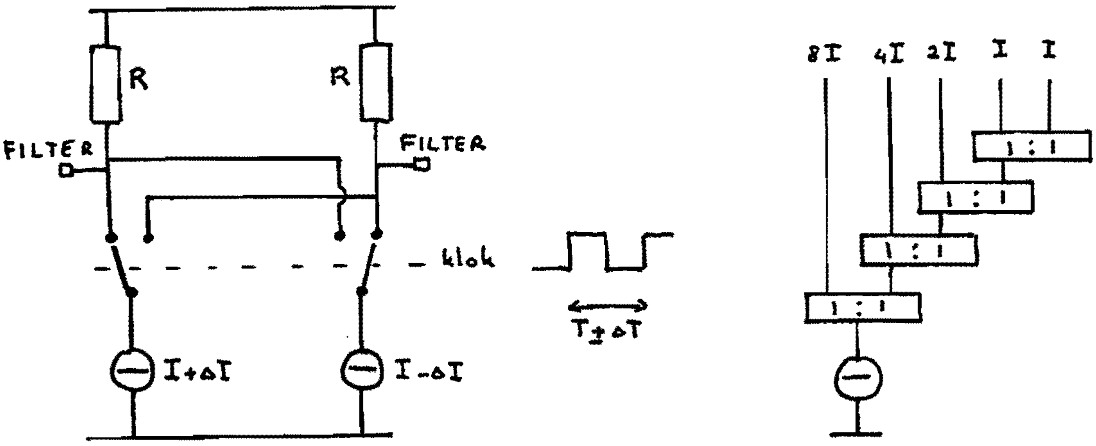


Figuur 8: Digitaal-Analoog omzetter met R-2R ladder netwerk.

tevens gebruik wordt gemaakt van het schakelen van stromen i.p.v. spanningen. Door de keuze van de verhouding van de weerstanden t.w. R en 2R ontstaat een stroomde-ling met een factor 2 voor iedere R-2R sectie. In dit geval wordt de uitgangsstroom gegeven door:

$$I_{out} = (I_{ref}/8)(8b_1 + 4b_2 + 2b_3 + b_4)$$

Binnen bovengenoemde types D/A converters blijft het hoofdprobleem de nauwkeurigheid van de te sommeren stromen. Voor D/A omzetters op relatief lage klokfrequenties is het principe van de "dynamic element matching" een bruikbare manier om de nauwkeurigheid van de stromen te vergroten. De stromen in fig. 9 (bijvoorbeeld verkregen uit een stroomspiegel) hebben de waarde I met een onnauwkeurigheid van +/- dI. Het signaal dat de schakelaar bestuurt heeft een peri-



Figuur 9: Principeschema dynamic element matching

odeduur van T en een duty-cycle van 50 % met een onnauwkeurigheid dT. Omdat in de linker tak gedurende het hoog zijn van de klok I+dI loopt en gedurende het laag zijn van de klok I-dI, en voor de rechter tak het omgekeerde geldt, kan worden afgeleid dat:

$$I_{out}T = (T/2 + dT/2)(I + dI) + (T/2 - dT/2)(I - dI) = I.T + dT.dI$$

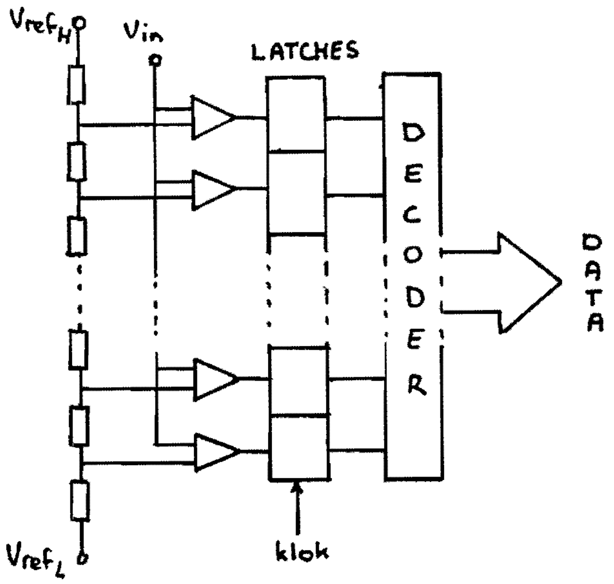
De fout op de gemiddelde stroom I is dus het produkt van de onnauwkeurigheid van stroom en klokperiode, resp. dI/I en dT/T. Als men dus in staat is om een stroom op 10 % nauwkeurig te maken en een klokperiode ook op 10 %, dan heeft de op deze wijze verkregen stroom een onnauwkeurigheid van slechts 1 %. Voor de noodzakelijke nauwkeurigheid voor het CD-systeem (16 bit) van 0,0008 % blijkt dit principe goed bruikbaar.

4 Analoog/Digitaal Omzetting, enige basis principes

Analoog/Digitaal omzetting is de functie welke nodig is om een analoge grootheid om te zetten in een aan de grootheid gerelateerd getal. Dit gebeurt door de waarde van de grootheid te meten en de gevonden waarde af te ronden op een veelvoud van een gekozen discretisatie- of quantisatie-eenheid. De consequentie van de grootte van de quantisatie-eenheid is dat een beperking wordt aangebracht voor wat betreft de getrouwheid waarmee het continue amplitude signaal gereconstrueerd kan worden. Deze getrouwheid wordt verder bepaald door de nauwkeurigheid waarmee de veelvouden van de quantisatie eenheid, de zg. quantisatie-niveaus, worden vastgelegd.

*Full parallel flash comparatoren

Een methode van A/D omzetting waarbij op een vast tijdstip (het klokmoment) de ingangsspanning wordt omgezet in een digitale representatie, is die waarbij gebruik wordt gemaakt van 2^n comparatoren. Voor 3 bits omzetting is het schema gegeven in figuur 10. Door middel van een spanningsdeler worden van een referentiespanning de benodigde referentiespanningen afgedeeld. Deze worden toege-



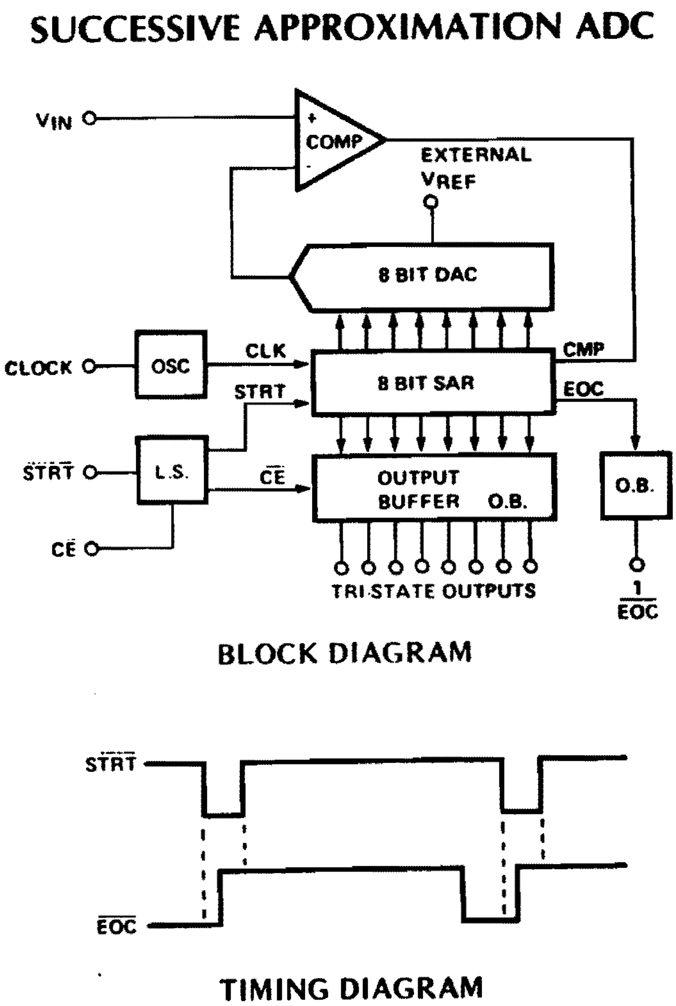
Figuur 10: A/D omzetting m.b.v. parallele comparatoren.

voerd aan de inverterende ingang van de comparatoren. De ingangsspanning wordt toegevoerd aan de niet-inverterende ingangen. Op een klokpuls worden de uitgangen van de comparatoren overgenomen in een aantal sample-latches. Dit resultaat geeft direct aan tussen welke waarden van de afgedeelde referentiespanning de ingangsspanning ligt. Hieruit kan nu de binaire representatie worden afgeleid.

Deze methode van A/D omzetting is bijzonder geschikt wanneer korte conversietijden vereist zijn. Dit is bijvoorbeeld het geval wanneer men videosignalen wil digitaliseren omdat hierbij conversie-frequenties groter dan 10 MHz nodig zijn. Een nadeel is uiteraard dat er voor elk quantisatie-niveau dat gerealiseerd moet worden een comparator en een sample-latch vereist is. Voor een 8 bits ADC zijn dit 256 comparatoren en latches terwijl ook de codering steeds complexere logische schakelingen vraagt naarmate het aantal comparatoren toeneemt. Recentelijk is door "vouw- en interpolatietechnieken" een enorme reductie in aantallen comparators en latches verkregen.

*Successive Approximation

Bij de successive approximation methode wordt een DAC opgenomen in de terugkoppeling van een comparator. Deze comparator vergelijkt nu het analoge ingangssignaal met de uitgang van de DAC (zie figuur 11). Het uitgangssignaal van de comparator wordt nu gebruikt voor het aansturen van een zg. Successive



Figuur 11: A/D omzetter volgens Succesive Approximation principe.

Approximation Register (SAR). Het SAR verzorgt de aansturing van de DAC en ontleent zijn naam aan het feit dat er bit voor bit, te beginnen met het MSB (MSB = Most Significant Bit), gekeken wordt of het omzetten van een bepaald bit resulteert in een uitgangsspanning van de DAC welke groter dan wel kleiner is dan de ingangsspanning van de ADC. Is het resultaat van het op "een" zetten van een bepaald bit dat de uitgangsspanning van de DAC groter is dan de ingangsspanning van de ADC dan zal dat bit op "nul" worden terug gezet, is dit niet het geval dan zal het betreffende bit "een" blijven. Dit vergelijken wordt successievelijk voor alle bits gedaan waarna het resultaat beschikbaar is in het uitgangs-register. De voordelen van deze manier van omzetten zijn o.a.:

i) hoge resolutie en snelheid van de omzetting en

ii) iedere omzetting vindt plaats binnen dezelfde tijd, onafhankelijk van de grootte van het ingangssignaal.

Hier staat tegenover dat gedurende de omzettingstijd (conversion time) het ingangssignaal niet mag veranderen. Om dit te bereiken zal voor de ADC een houd-circuit geschakeld moeten worden. Gewoonlijk wordt dit gerealiseerd m.b.v. een bemonster- en houdschakeling waardoor het signaal tevens tijd-discreet wordt gemaakt. De omzettingstijd wordt nu bepaald door de insteltijd van de bemonster- en houdschakeling plus n maal de insteltijd van de DAC. Hierbij is n het aantal bits van de representatie. Hieruit ziet men dat de insteltijd van de ADC bijna geheel bepaald wordt door de insteltijd van de DAC. Voor een 16 bits ADC met een insteltijd van maximaal 20 microseconde (50 kHz) is een DAC vereist met een insteltijd van maximaal 1.25 microseconde (800 kHz). Hierbij is de insteltijd van de bemonster- en houdschakeling buiten beschouwing gelaten.

* delta modulator

Een andere vorm van A/D conversie is gebaseerd op het quantiseren van de verschillen van opeenvolgende signaalmonsters. Een dergelijke quantisatiemethode wordt differentiele quantisatie genoemd. Differentiele quantisatiemethoden kunnen worden toegepast indien er sprake is van correlatie tussen opeenvolgende signaalmonsters. Dit is het geval als de te quantiseren signalen zijn bemonsterd met een frequentie die vele malen groter is dan de door het bemonsteringstheorema voorgeschreven bemonsterfrequentie. De eenvoudigste toepassing van het concept van differentiele quantisatie is Delta Modulatie. Het eenvoudigste Delta modulatie systeem is gegeven in figuur 12a. In dit geval heeft de quantisator Q slechts twee niveaus met een vaste stapgrootte. Het positieve quantisatieniveau wordt gerepresenteerd door c(n)=0 en het negatieve door c(n)=1. $\hat{d}(n)$ is dan:

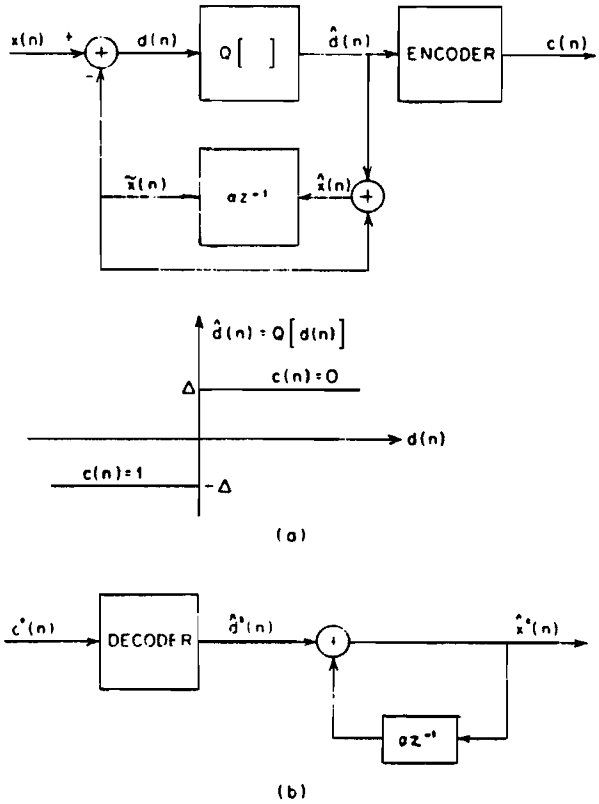
$$\hat{d}(n) = \delta \qquad als \quad c(n) = 0$$

$$\hat{d}(n) = -\delta \qquad als \quad c(n) = 1$$

De schakeling in de terugkoppeling van de quantisator is een eerste orde filter waarvan de overdrachtsfunctie gegeven wordt door:

$$H(z) = a.z^{-1}/(1 - az^{-1})$$

Voor a = 1 is dit filter het digitale equivalent van een integrator met dien verstande dat het de optelling van positieve en negatieve incrementen ter grootte delta verte-



Figuur 12: Blokscema van Delta-Modulatie a) Coder, b) Decoder.

genwoordigt. Voor a < 1 hebben we te maken met een "lekkende" integrator. Door nu de uitgang van deze integrator af te trekken van het ingangssignaal wordt alleen het verschil tussen de geaccumuleerde uitgang van de integrator en het ingangssignaal gequantiseerd. Met behulp van figuur 12a kan worden aangetoond dat $\hat{x}(n)$ de gequantiseerde waarde van x(n) is. Er geldt namelijk:

$$d(n) = x(n) - \tilde{x}(n)$$

$$\hat{d}(n) = d(n) + e(n)$$

$$\hat{x}(n) = \tilde{x}(n) + \hat{d}(n)$$

Hieruit kan direct worden afgeleid dat:

$$\hat{X}(n) = x(n) + e(n)$$

Het signaal $\hat{x}(n)$ kan dus uit de samples $\hat{d}(n)$ worden gereconstrueerd d.m.v. de integrator (1+H(z)). De decoder heeft dan ook de structuur zoals afgebeeld in figuur 12b. Voor a $\approx$ 1 volgt uit bovenstaande vergelijkingen ook:

$$d(n) = x(n) - x(n-1) - e(n-1)$$

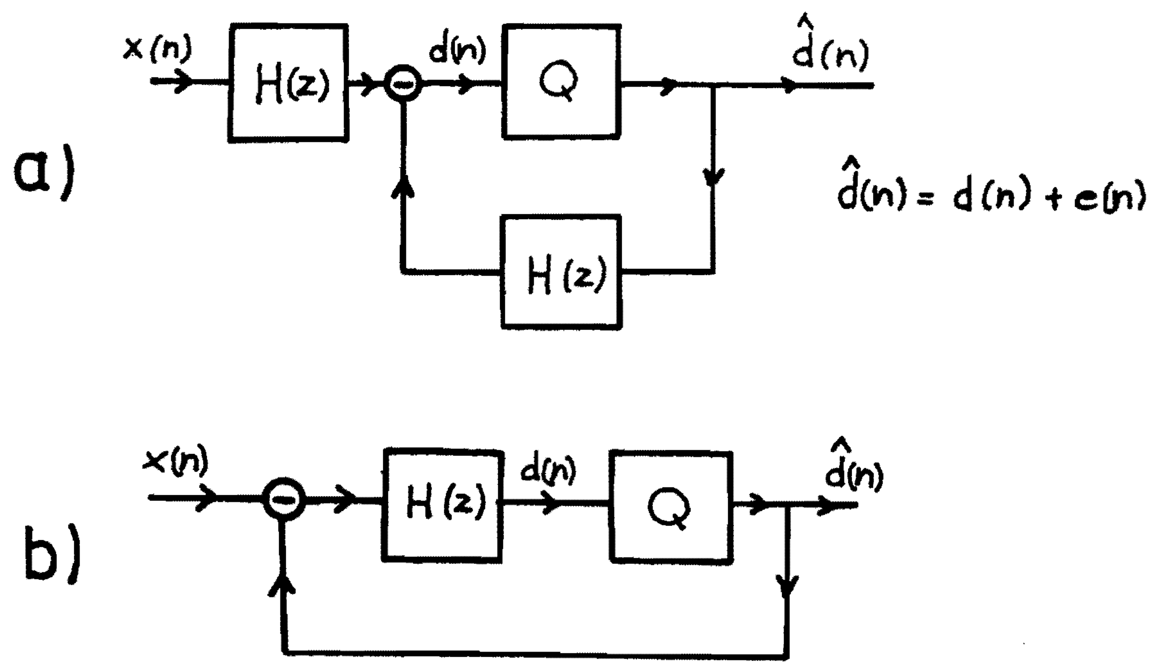
Dus afgezien van de quantisatiefout in $\hat{x}$(n-1) is d(n) het eerste orde verschil van x(n) en x(n-1), wat gezien kan worden als de digitale benadering van de afgeleide van het ingangssignaal x(n). Dit is het inverse proces van de integratie in de terugkoppeling. Consequentie hiervan is dat reeks [$\hat{x}$(n)], een maximale helling in de reeks [x(n)] slechts goed kan weergeven als voldaan is aan de relatie:

$$\delta/T > max|dx_a(t)/dt|$$

waarbij $x_a(t)$ het analoge signaal is waaruit door bemonstering de reeks x(n) wordt gegenereerd. We merken hier op dat de bemonstering van het analoge signaal in het hier gegeven schema voor delta modulatie, voor de delta modulator plaats vindt. Dit in tegenstelling tot de meer gebruikelijke bemonstering tussen de opteller aan de ingang en de quantisator. Wordt aan boven gegeven relatie niet voldaan dan treedt er de zg. hellingsbegrenzing op, hetgeen aanzienlijke vervorming van het signaal inhoudt We merken hierbij op dat aangezien de maximale helling in $\hat{x}$(n) vastgelegd wordt door de stapgrootte delta, de toe- en afname van de reeks $\hat{x}$(n) lineair gebeurt. Hierom wordt deze vorm van quantisatie ook wel Lineaire Delta Modulatie genoemd.

4 sigma delta modulatie

We hebben laten zien dat bij delta modulatie volgens het schema van figuur 12a de monsters d(n), afgezien van een quantisatie fout, de digitale benadering van de



Figuur 13: Coder structuren voor Sigma Delta-Modulatie.

afgeleide van x(n) representeren. Door nu aan de ingang van de delta modulator een integrator op te nemen vormen de samples d(n) een representatie van x(n). Deze vorm van delta modulatie, welke is afgebeeld in figuur 13a wordt Sigma Delta Modulatie genoemd. Door het toevoegen van deze integrator behoeft de decoder nog slechts uit een laagdoorlaat filter te bestaan en bevat geen integrator meer zoals de schakeling van figuur 12b.

De schakeling van figuur 13a kan nog vereenvoudigd worden door de integrator in de lus in het directe pad op te nemen (zie figuur 13b). We kunnen eenvoudig bewijzen dat beide schakelingen identiek zijn.

Beschouwen we nu de z-getransformeerde van het uitgangssignaal van de sigma delta modulator:

$$\hat{D}(z) = [H(z).X(z) + E(z)]/[1 + H(z)]$$

dan kan deze uitdrukking herschreven worden als:

$$[1 + H^{-1}(z)]\hat{D}(z) = X(z) + H^{-1}(z).E(z)$$

Kiezen we voor H(z) het filter van figuur 12 dan krijgen we:

$$\hat{D}(z) = a.z^{-1}.X(z) + (1 - a.z^{-1}).E(z)$$

We zien dat het uitgangssignaal van de sigma delta modulator een vertraagde versie is van het ingangssignaal plus een foutsignaal. In tegenstelling tot de gewone delta modulator van figuur 12, is hier het spectrum van de fout gefilterd met de functie $(1 - a.z^{-1})$, waardoor het foutspectrum van het uitgangssignaal niet langer wit is maar een oploopt naar hogere frequenties. Het totale foutvermogen is toegenomen met een factor $1 + a^2$, maar het spectrum is dusdanig gevormd dat het vermogen bij lage frequenties is afgenomen (noise shaping). Het voordeel hiervan is dat het ruisvermogen in de signaalband sterk is verminderd (bedenk hierbij dat het ingangssignaal van een delta modulator sterk is overbemonsterd). Door nu aan de ontvangzijde (decoder) een laagdoorlaat filter te gebruiken dat alleen de signaalband doorlaat kan bij sigma delta modulatie de signaal/quantisatie ruis verhouding in de signaalband sterk worden verbeterd t.o.v. lineaire delta modulatie.

5 Specificaties van A/D's en D/A's

Om A/D's en D/A's enigszins te kunnen beoordelen voor gebruik in hun directe toepassingsgebied, is enige kennis nodig van de (niet) idealiteit van deze componenten. In de nu navolgende regels volgen de belangrijkste kenmerken van A/D en D/A converters.

* Resolutie (A/D,D/A)

Dit is de kleinste stapvormige verandering in het uitgangsignaal van een DAC. Een omzetter met n schakelaars heeft een resolutie van 1 op 2^n. Een 12 bits DAC heeft een resolutie van 1 op 2^{12} (een deel op 4096) ofwel 0,000245. Is de volle schaalwaarde van de DAC 10 volt (piek-piek) dan is de resolutie 2,45 mV. De resolutie van een DAC of een ADC is slechts een ontwerp parameter en beschrijft alleen het aantal datalijnen. Of deze datalijnen relevante informatie bevatten (A/D) danwel een significante bijdrage geeft aan het uitgangssignaal (D/A) zit niet in deze specificatie. Alleen bij ideale converters, bij geen enkele praktische A/D of D/A bevatten alle datalijnen relevante informatie.

* Klokfrequentie (A/D,D/A)

Dit is de frequentie waarop de converter in staat is informatie te vertalen. Of de converter dit tot die frequentie correct doet is niet uit deze specificatie af te leiden. Bijvoorbeeld hoeft het geenszins het geval te wezen dat een A/D converter met een klokfrequentie van 200 MHz een analoog signaal tot de helft van deze frequentie (Nyquist) correct kan vertalen. Er komen wel nullen en enen op die snelheid uit maar of deze codes enige correlatie vertonen met het ingangssignaal is niet uit deze specificatie te halen, en in de praktijk blijkt dit meestal niet het geval.

*Offset of nulfout (D/A)

Dit is de uitgangsspanning van de DAC als de nul-code aan de ingang wordt aangeboden. Deze fout wordt meestal veroorzaakt door een niet correct op nul afgeregelde versterker. Deze fout kan m.b.v. een externe offset potentiometer meestal tot nul worden gereduceerd.

* Gain error of schaalfout (D/A)

Het verschil tussen de verwachte maximale uitgangsspanning (maximale code * V_{ref}) en de werkelijke maximale uitgangsspanning is de schaalfout. Schaalfouten kunnen ontstaan door fouten in de referentie spanning, de waarden van de ladderweerstanden, versterkingsfactoren etc. In het algemeen kunnen schaal- fouten gecorrigeerd worden door het aanpassen van de versterking van de DAC.

* Leariteit (A/D,D/A)

-Integrale niet leairiteit

De integrale niet leairiteit geeft de afwijking aan van de lineaire overdrachtskarakteristiek. Deze lineaire overdrachtskarakteristiek is de meest ideale rechte lijn door de werkelijke overdrachtscurve waarbij de ofsett- en gainerrors apart bekeken worden. Een specificatie van de integrale niet leairiteit als +/- 1/2 LSB betekent dus een extra fout naast de quantisatie- of resolutiefout.

-Differentiele niet leairiteit

De differentiele niet-leairiteit geeft voor iedere willekeurige code- verandering het verschil aan tussen de werkelijke verandering en de ideale verandering van 1 LSB. Als de uitgang van de DAC bijvoorbeeld een stap maakt van 2 LSB bij een codeverandering dan komt dit overeen met een differentiele niet-leairiteit van 1 LSB. De differentiele leairiteit vormt als het ware een maat voor de geleidelijkheid waarmee de karakteristiek verandert.

* Monotoniciteit (A/D,D/A)

Een monotone kromme is een continue niet-dalende (of niet-stijgende) kromme. Monotoniciteit betekent dat veranderingen in de code altijd een verandering van de uitgangsspanning veroorzaken in dezelfde richting als de codeverandering. Dus wanneer de codeveranderingen positief zijn dient de uitgangsspanning met een bedrag groter of gelijk aan nul te veranderen, wanneer negatieve codeveranderingen optreden dan moet de uitgangsspanning met een bedrag kleiner of gelijk aan nul veranderen.
Een integrale niet lineairiteits specificatie van 1/2 LSB garandeert een monotone karakteristiek. Is de niet-lineairiteit groter dan 1/2 LSB dan kan de karakteristiek niet-monotoon zijn maar dit hoeft niet het geval te zijn zolang de differentiele niet-lineairiteit kleiner is dan 1/2 LSB.

* Insteltijd (D/A)

De insteltijd (of settling time) is de tijd die verloopt tussen de codeverandering aan de ingang en het moment waarop de D/A uitgang zijn uiteindelijke waarde binnen gespecificeerde grenzen bereikt. Deze grenzen zijn meestal +/- 1/2 LSB. Naast de insteltijd wordt ook vaak de stijgtijd opgegeven. Dit is de tijd die nodig is om voor de eerste keer in bovengenoemd interval te komen. Als de sprongkarakteristiek van de codeverandering weinig oscillaties vertoont, is de stijgtijd en de insteltijd veelal gelijk.

* Glitches (D/A)

Glitches zijn ongewenste piekvormige signalen welke optreden bij een codeverandering in DA converters. (Fig 14) Ze ontstaan door het op- en ontladen van parasitaire capaciteiten. De grootte is afhankelijk van de codeverandering. (bij de

Figuur 14: Glitches in D/A omzetters.

gewone binaire code heeft de overgang van het MSB het grootste effect omdat dan alle andere bitten van 1 naar 0 gaan en het MSB van 0 naar 1) Glitches worden meestal gespecificeerd met een glitchenergie bij maximale klokfrequentie. Dit kan betekenen dat de glitch verscheidene LSB's hoog kan zijn maar een niet al te lange duur heeft. Als de schakeling niet tegen dit soort hoge pieksignalen kan, is het nodig een Sample/Hold schakeling toe te voegen teneinde de waarde van de uitgang vast te houden na de glitch.

* Effectieve bits (A/D,D/A)

Om een indruk te krijgen hoeveel datalijnen werkelijk bruikbare informatie bevatten is het begrip effectieve bits ingevoerd. Het wordt afgeleid van de gemeten signaal/ruis verhouding.
In het voorafgaande is voor een ideale converter afgeleid dat de signaal/ruis verhouding ten gevolge van quantisatie gelijk is aan : $S/N = 6.02n + 1.76dB$ waarbij n het aantal bits is. Als we alle afwijkingen van het ideale ruis noemen (quantisatie, harmonische vervormingen, niet lineariteiten enz.) dan kunnen we zeggen: $n = (S/N - 1.76)/6.02$. Hierbij is n het aantal effectieve bits. Dit is dan direct een maat voor de bruikbaarheid van de datalijnen.

* Differential gain and phase (A/D)

De begrippen differential gain and phase komen voort uit de klassieke analoge wereld. Ze vinden hun definitie in een continue overdrachtskarakteristiek. Voor een A/D en een D/A geldt de continuiteit van de overdrachtskarakteristiek niet. Het hele quantisatiegebeuren is discontinu. Op grond daarvan staat de betekenis van specificaties als differential gain and phase voor A/D en D/A converters volledig ter discussie.

6 Referenties

[1] Gordon B.M., "Linear Electronic Analog/Digital Conversion Architectures, Their Origins, Parameters, Limitations and Applications"
IEEE Trans on Circuits and Systems, Vol. CAS-25, No. 7, July 1978, pp. 391-418
[2] Plassche van de R.J., "Is there live beyond 14-bits in A/D and D/A conversion?"
ESSIRC Digest of Technical Papers, September 1982, pp. 49-65
[3] Hoeschele D.F. Jr, "Analog-to-Digital/Digital-to-Analog Conversion Techniques"
John Willey and Sons, Inc., New York, 1968
[4] Tewksbury S.K., et.al., "Terminology Related to the Performance of S/H, A/D and D/A Circuits"
IEEE Trans. on Circuits and Systems, Vol. CAS-25, No. 7 July 1978, pp. 419-426
[5] Maio K., et.al., "A 500 MHz 8-bit D/A Converter"
IEEE J. of Solid State Circuits, Vol. SC-20, No.6, December 1985, pp. 1133-1137
[6] Watson D. ,et.al. "12-bit DAC Chip races at Video speeds, but foregoes Deglitching circuitry"
Electronic Design, June 13, 1985, pp. 111-118
[7] Singh S.P., et.al., "Modified C-2C ladder voltage divider for application in PCM A/D Convertors"
Electronic Letters, 15 September 1983, Vol. 19, No. 19, pp. 788-789

[8] Sugawara T., et.al., "A monolithic 14 bit/20us dual channel A/D converter."
IEEE J. of Solid State Circuits, Vol. SC-18, No.6, December 1983, pp. 723-729
[9] Saul P.H., et.al., "Techniques and technology for high speed D-A conversion."
February 1984, pp. 62-68
[10] Petschacher R., et.al., "New methods improving static and dynamic performance of an 8-bit/120-MHz A/D converter."
ESSCIRC Digest of technical papers, September 1985, pp. 40-44
[11] Plassche van de R.J., "A sigma delta modulator as an A/D converter"
IEEE Trans on Circuits and Systems, Vol. CAS-25, No.7, July 1978, pp.510-514
[12] Grift van de R.E.J.,et.al., "An 8-bit video ADC incorporating folding and interpolation techniques"
IEEE J. of Solid state circuits, Vol. SC-22, No.6, December 1987, pp. 944-953
[13] Saul P.H., "successive aproximation analogue to digital conversion at video rates."
ESSCIRC Digest of technical papers, September 1980, pp. 350-352
[14] Jamal H., et.al., "A digital dual slope analogue to digital converter"
IEE Proceedings, Vol.132, Pt.G., No.4, August 1985, pp. 149-152
[15] Adams R.W., "Design and implementation of an audio 18-bit Analog-to-Digital converter using oversampling techniques"
J. of Audio Engineering Society, Vol. 34, No.3, March 1986, pp. 153-166
[16] Naus P.J.A, et.al., "A CMOS stereo 16 bit D/A converter for digital audio"
ESSCIRC Digest of technical papers, September 1986, pp 152-154.

DIGITALE SIGNAALBEWERKING

deel II.4

Adaptieve Digitale Signaalbewerking

door

P.C.W. Sommen

Technische Universiteit Eindhoven

Adaptieve Digitale Signaalbewerking
Een korte inleiding

P.C.W. Sommen

September 1988

Samenvatting

Als onderdeel van de PATO–cursus "Digitale Signaalbewerking" wordt hier een korte inleiding gegeven van de adaptieve digitale signaalbewerking.

De bedoeling van dit college is dat een aantal verschillende aspecten, die voor adaptieve digitale signaalbewerking van belang zijn, aan de orde komen. Dit zal gedaan worden aan de hand van een aantal praktische voorbeelden afgewisseld met stukjes theorie. Dit overzicht is zeker niet kompleet, er zal derhalve daar waar mogelijk naar literatuur verwezen worden.

Dit diktaat is als volgt ingedeeld:

In de inleiding wordt een motivatie gegeven voor het gebruik van adaptieve digitale filters, waarna een voorbeeld zal worden behandeld dat bekendheid heeft verworven in de datatransmissie. Aan de hand van dit voorbeeld zal het bekendste algoritme op het gebied van adaptieve digitale filters worden afgeleid, het "Least Mean Square" (LMS) algoritme.

Hierna worden drie verschillende toepassingen van adaptieve digitale filters, met enkele praktische voorbeelden, behandeld.

Uitgaande van al de voorgaande kennis kunnen nu een viertal fundamentele kenmerken van adaptieve algoritmen worden afgeleid.

Tot slot zullen nog enkele andere algoritmen en structuren in het kort worden behandeld.

Inhoudsopgave

1 Inleiding

Adaptieve technieken zijn de laatste jaren in opkomst op het gebied van digitale signaalbewerking. Dit komt door de ontwikkeling van goede adaptieve algoritmen maar voornamelijk door het feit dat de hardware, zoals die in de digitale signaalbewerking wordt toegepast, zeer goed geschikt is om adaptieve algoritmen te realiseren.

We zullen ons in dit college met name richten op enkele basisprincipes van adaptieve systemen.

Om de motivatie voor het gebruik van adaptieve signaalbewerking toe te lichten gebruiken we het transmissie kanaal zoals we dat bijvoorbeeld bij het verzenden van data signalen tegenkomen (in de volgende paragraaf wordt dit voorbeeld verder behandeld). Als het transmissie kanaal constante fysische eigenschappen heeft die bekend zijn aan de systeemontwerper, en als de te verzenden of ontvangen signalen goed gedefinieerd en stationair zijn, dan stelt deze a priori kennis de systeemontwerper om het optimale systeem te verkrijgen bij een gegeven foutencriterium. Deze methode kan worden geïmplementeerd in een vast systeem en kan ten alle tijden worden gebruikt. Vaste filters in telefonie zijn een voorbeeld van dit type.

In het algemeen echter, zijn de karakteristieken van het transmissie kanaal redelijk <u>onbekend</u> terwijl ze bovendien nog kunnen <u>variëren</u> in de tijd. In zo'n situatie zijn er twee manieren om het systeem te ontwerpen:

1. Ontwerp een systeem waarvan de parameters van te voren worden vastgesteld. Doe dit zodanig dat op het "gemiddelde" kanaal het beste resultaat wordt verkregen. We hebben dan een soort "compromis" systeem.

2. Ontwerp een systeem dat zichzelf kan aanpassen ("adapteren") aan een <u>onbekend</u> transmissie kanaal. De adaptiviteit zorgt er voor dat het systeem beter past bij het <u>variërende</u> transmissie kanaal dan het "compromis" systeem.

Uit het voorgaande kunnen we concluderen dat bij toepassing van adaptieve digitale signaalbewerking het onbekende systeem eigenschappen heeft die "geleerd" moeten worden. Dit leerproces willen we meestal met een bepaalde <u>snelheid</u> en <u>nauwkeurigheid</u> uitvoeren. Verder is aangegeven dat parameters van het onbekende systeem kunnen variëren op een vooraf onbekende manier. Deze variaties zullen door het adaptief filter gevolgd moeten kunnen worden, en we spreken in dit verband dan van '<u>tracking</u>'. Tenslotte willen we dat het adaptief filter realiseerbaar moet zijn.

Conclusie:
Een adaptief filter wordt vaak toegepast in een omgeving die onbekend is en kan variëren.
De kwaliteitseisen die bij een adaptief filter een belangrijke rol spelen zijn:

1. snelheid en nauwkeurigheid waarmee geregeld kan worden,
2. tracking eigenschappen en
3. realiseerbaarheid van het adaptief filter.

In dit college zal, vanwege het korte tijdsbestek, geen aandacht worden besteedt aan de tracking eigenschappen van adaptieve filters.
Tot slot merken we in deze inleiding nog op dat de notatie in dit college zoveel mogelijk aansluit bij de notatie die gebruikt wordt in [1].

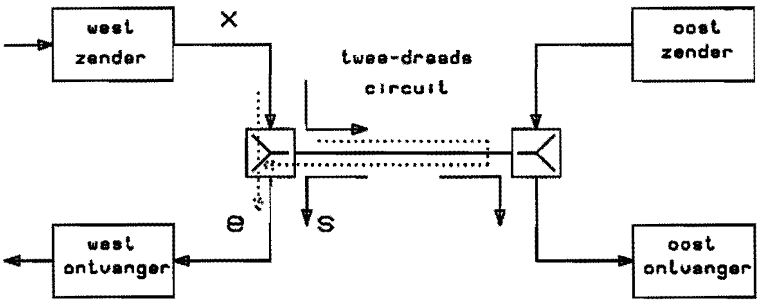
2 De adaptieve echo canceller: een praktisch voorbeeld

In deze paragraaf zullen we een voorbeeld geven van een adaptief filter dat veel in de praktijk wordt toegepast. Communicatie van data over bestaande telefoonkabels vereist de simultane transmissie van data in tegengestelde richtingen ("full–duplex" bewerking). In het verleden kon full–duplex service verkregen worden door de toepassing van vier draads circuits, of door de helft van de bandbreedte op te offeren van een twee draads circuit voor de transmissie in de tegengestelde richting.

Door toepassing van digitale signaalbewerkingtechnieken kan nu een elegantere en meer economische oplossing worden toegepast: digitale echo compensatie door een adaptief filter (= adaptieve echo canceller. Deze methode staat een full–duplex service toe op een twee draads circuit met de beschikking over de volledige bandbreedte voor beide richtingen van transmissie. Enkele eigenschappen die er voor zorgen dat digitale bewerking voor deze toepassing de voorkeur geniet boven analoge bewerking zijn: groot dynamisch bereik, integreerbaarheid en dus lage vermogens dissipatie en kleine afmetingen.

Bij data transmissie willen we dus data verzenden over (bestaande) telefoonkabels van de ene abonnee (zender) naar de ander (ontvanger) en omgekeerd. Vanwege de hoge kosten willen we hiervoor gebruik maken van één aderpaar. Full–duplex data transmissie over een twee draads circuit kan in principe bereikt worden door de toepassing van vorkschakelingen. Deze vormen een interface tussen vier draads– en twee draads–verbindingen. Vorkschakelingen worden veel toegepast in telefonie netwerken. Ze brengen echter wel een imperfecte isolatie tussen zender en ontvanger teweeg aan iedere zijde ("west" en "oost") van de verbinding (zie Fig. 1).

Een deel van het verzonden west signaal x lekt direct in de west ontvanger

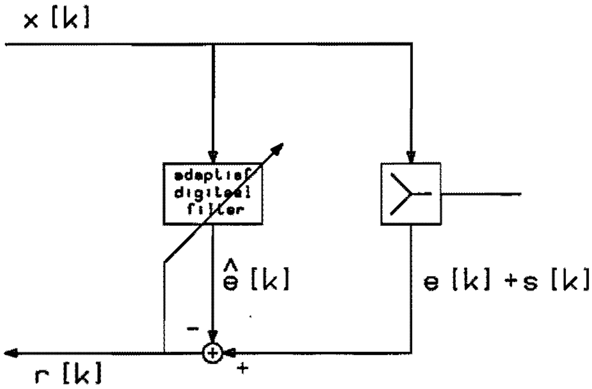


Figuur 1: *Full–duplex data transmissie op een twee–draads circuit door middel van vorkschakeling.*

als gevolg van de onbalans binnen het circuit van de vorkschakeling. Een ander deel van het verzonden signaal wordt gereflecteerd vanwege impedantie onregelmatigheden in het twee draads circuit en dit komt ook terug in de eigen ontvanger. Van nu af aan beschouwen we de combinatie van deze beide imperfecties als één "echo" en zullen dit signaal aanduiden met het signaal e. Als voorbeeld gelden de volgende praktische gegevens: de lek van het ingangssignaal tussen zender en ontvanger door de vorkschakeling is in de orde grootte van 10 dB terwijl voor lange verbindingen het ontvangen signaal van de andere zijde, signaal s, verzwakt is met ongeveer 40 dB. Dit betekent dat de echo zo'n 30 dB sterker is dan het gewenste signaal, terwijl deze verhouding zo'n -20dB moet zijn om in staat te zijn op een acceptabele manier het gewenste signaal te detecteren in de ontvanger. Echo compensatie door middel van een adaptief digitaal filter kan gemakkelijk de 50 dB echo reductie van bovengenoemd voorbeeld halen.

Het echo signaal e kan gemodelleerd worden als het ingangssignaal x dat door een ongewenst transmissie pad ("echo pad") tussen de zender en ontvanger lekt. Voor de eenvoud beschouwen we hier dit echo pad als lineair.

Verder varieert het slechts langzaam met de tijd onder andere ten gevolge van veranderingen in de temperatuur. Om die reden kan echo compensatie verkregen worden door een parallel pad te maken tussen zender en ontvanger waar een kopie $\hat{e}[k]$ van de ontvanger echo $e[k]$ wordt gemaakt door middel van een adaptief digitaal filter (Fig. 2). Door aftrekking van de kopie van het



Figuur 2: *Adaptieve echo canceller in data transmissie*

inkomend signaal kan de echo worden onderdrukt. Na aftrekking verkrijgen we een residu signaal $r[k]$ wat gegeven wordt door:

$$r[k] = s[k] + e[k] - \hat{e}[k]. \tag{1}$$

Hierin is $s[k]$ het ontvangen oost signaal. Het residu signaal $r[k]$ wordt zowel toegevoerd aan de ontvanger van de westzijde voor detectie als aan het digitaal filter om zijn coëfficiënten te adapteren.

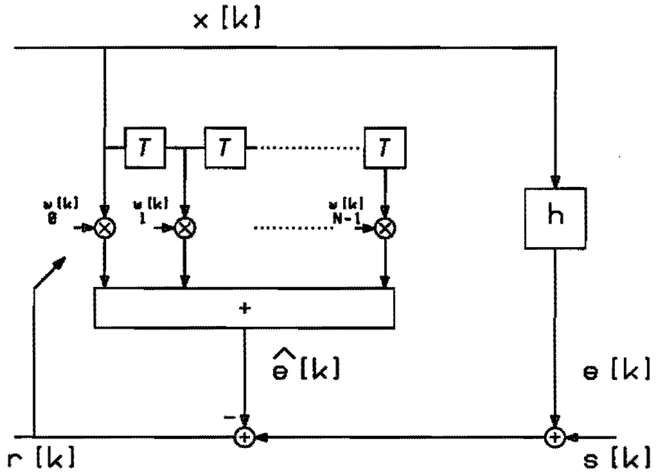
In Fig. 2 zijn alle signalen gerepresenteerd in discrete tijd. In de praktijk echter zal het signaal tussen de vorkschakeling en het aftrekpunt een analoog signaal zijn terwijl men in het algemeen ook een analoog signaal wenst bij de ontvanger zelf. We moeten daarom in de praktijk bij de implementatie van een digitale echo canceller A/D en D/A conversies uitvoeren. Dit valt echter buiten het bestek van dit college (zie hiervoor bijvoorbeeld [2]) en we zullen ons hier verder alleen nog richten op digitale signalen.

Conclusie:

Bij datatransmissie kan de imperfectie van een vorkschakeling, die vooraf onbekend is en kan variëren in de tijd, gecompenseerd worden door een adaptief filter parallel te schakelen: de echo canceller.

3 Het LMS algoritme

Het "Least Mean Square" (LMS) algoritme is een robuust en eenvoudig te implementeren algoritme en wordt om die redenen vaak in de praktijk toegepast. Aan de hand van de echo canceller voor datatransmissie toepassingen, zoals die is beschreven in de vorige paragraaf, zullen we dit algoritme in paragraaf 3.1 afleiden. We gaan daarbij uit van een model van het echo pad zoals dat is beschreven in de voorgaande paragraaf. Dit model is getekend in Fig. 3. We geven hierbij de beschouwing voor de west zijde, de oost zijde gaat immers op dezelfde manier. In paragraaf 3.2



Figuur 3: *Basis schema van adaptieve echo canceller uitgevoerd als transversaal filter*

wordt een afleiding gegeven van een differentievergelijking die een volledige beschrijving geeft van het dynamisch gedrag van een echo canceller gebaseerd op het LMS algoritme voor data transmissie signalen. Deze paragraaf is alleen voor de volledigheid gegeven en kan in principe worden overgeslagen. De conclusies die we uit deze differentievergelijking kunnen trekken geven in paragraaf 3.3 een beschrijving van het convergentie gedrag van het LMS algoritme.

3.1 Afleiding van het LMS algoritme

We gaan er van uit dat het echo pad beschreven kan worden door een impulsresponsie h_i die van eindige duur (N) is met $h_i = 0$ voor $i < 0$ en $i \geq N$. We kunnen dan een transversaal digitaal filter met N variabele coëfficiënten $w_0[k], w_1[k], \cdots, w_{N-1}[k]$ gebruiken voor de opwekking van de echo kopie $\hat{e}[k]$. Het ingangssignaal wordt aangeduid met $x[k]$.

Om de analyse van de echo compensator mogelijk te maken introduceren we de volgende vectoren:

$$\begin{aligned} \text{west ingangssignaal vector } \underline{\mathbf{x}}[k] &= (x[k], x[k-1], \cdots, x[k-N+1])^T \\ \text{echo pad vector } \underline{\mathbf{h}} &= (h_0, h_1, \cdots, h_{N-1})^T \\ \text{coëfficiënten vector } \underline{\mathbf{w}}[k] &= (w_0[k], w_1[k], \cdots, w_{N-1}[k])^T \end{aligned} \tag{2}$$

waarin $()^T$ vector transpositie voorstelt. We vinden nu voor het echo signaal $e[k]$ en de echo kopie $\hat{e}[k]$ de volgende convolutie sommen:

$$\begin{aligned} e[k] &= \sum_{i=0}^{N-1} x[k-i] h_i \\ \hat{e}[k] &= \sum_{i=0}^{N-1} x[k-i] w_i[k] \end{aligned} \tag{3}$$

welke ook geschreven kunnen worden als de volgende vector producten (merk hierbij op dat het produkt van een getransponeerde vector met een andere vector een scalar oplevert):

$$\begin{aligned} e[k] &= \underline{\mathbf{x}}^T[k] \cdot \underline{\mathbf{h}} \\ \hat{e}[k] &= \underline{\mathbf{x}}^T[k] \cdot \underline{\mathbf{w}}[k]. \end{aligned} \tag{4}$$

We kunnen nu het residu signaal herschrijven als:

$$r[k] = e[k] - \hat{e}[k] + s[k] = \underline{\mathbf{x}}^T[k] \cdot (\underline{\mathbf{h}} - \underline{\mathbf{w}}[k]) + s[k]. \tag{5}$$

Om een eenvoudigere notatie te krijgen introduceren we het differentie kanaal d wat het verschil is tussen het echopad h en de adaptieve coëfficiënten w. Hiervoor geldt dus:

$$d_i[k] = h_i - w_i[k] \quad i = 0, 1, \cdots, N-1 \tag{6}$$

oftewel in vector notatie:

$$\underline{\mathbf{d}}[k] = \underline{\mathbf{h}} - \underline{\mathbf{w}}[k]. \tag{7}$$

Hiermee kunnen we het residu signaal $r[k]$ herschrijven als:

$$r[k] = \underline{\mathbf{x}}^T[k] \cdot \underline{\mathbf{d}}[k] + s[k]. \tag{8}$$

De gemiddelde kwadratische fout hiervan wordt nu verkregen door van dit residu signaal de mathematische verwachting te nemen, welke gegeven wordt door:

$$\rho[k] \doteq E\{r^2[k]\} = E\left\{\left(\underline{\mathbf{x}}^T[k] \cdot \underline{\mathbf{d}}[k] + s[k]\right)^2\right\} \tag{9}$$

waar $E\{\}$ de mathematische verwachting voorstelt.
Deze uitdrukking zullen we nu eerst gaan uitschrijven. Dit levert:

$$\rho[k] = E\left\{\underline{\mathbf{x}}^T[k]\underline{\mathbf{d}}[k]\underline{\mathbf{d}}^T[k]\underline{\mathbf{x}}[k]\right\} + 2E\left\{\underline{\mathbf{x}}^T[k]\underline{\mathbf{d}}[k]s[k]\right\} + E\left\{s^2[k]\right\} \tag{10}$$

Omdat de west en de oost zenders onafhankelijke bronnen zijn kunnen we veronderstellen dat de signalen $s[k]$ en $x[k]$ onafhankelijk zijn, zodat de verwachtingswaarde van het product gelijk is aan het product van de verwachtingswaarden. Bovendien stellen we hier voor de eenvoud dat de verwachtingswaarde (gemiddelde waarde) van het signaal $s[k]$ nul is, dus $E\{s[k]\} = 0$. Als dit niet zo mocht zijn dan moet de volgende beschouwing geschaald worden. Met deze aannames kunnen we bovenstaande uitdrukking voor de gemiddelde kwadratische waarde van het residu signaal op de volgende manier herschrijven:

$$\begin{aligned}
\rho[k] &= E\left\{\underline{\mathbf{x}}^T[k]\underline{\mathbf{d}}[k]\underline{\mathbf{d}}^T[k]\underline{\mathbf{x}}[k]\right\} \\
&\quad +2E\left\{\underline{\mathbf{x}}^T[k]\underline{\mathbf{d}}[k]\right\} \cdot E\left\{s[k]\right\} + E\left\{s^2[k]\right\} \\
&= E\left\{\underline{\mathbf{x}}^T[k]\underline{\mathbf{d}}[k]\underline{\mathbf{d}}^T[k]\underline{\mathbf{x}}[k]\right\} + P_s
\end{aligned} \tag{11}$$

waarin $P_s = E\left\{s^2[k]\right\}$ het vermogen is van het signaal $s[k]$. De eerste uitdrukking aan de rechterzijde van uitdrukking (11) is gelijk aan nul als $\underline{\mathbf{d}}[k] = \underline{\mathbf{0}}$ oftewel als $\underline{\mathbf{w}}[k] = \underline{\mathbf{h}}$. Dan is $\rho[k]$ minimaal en wel gelijk aan P_s. Dat dit zo is voor $\underline{\mathbf{w}}[k] = \underline{\mathbf{h}}$ is niet zo verwonderlijk. Voor deze waarden van de adaptieve coëfficiënten volgt uit formule (4) dat de echo kopie $\hat{e}[k]$ gelijk is aan de echo $e[k]$. Hieruit volgt dan weer dat het residu signaal $r[k]$ gelijk is aan $s[k]$. Uit formule (11) volgt dus dat $\rho[k]$ een kwadratische functie is van de filter coëfficiënten met een absoluut minimum P_s. De afgeleide van $\rho[k]$ naar de adaptieve coëfficiënten $w_i[k]$ geeft aan in welke "richting" deze coëfficiënten aangepast moeten worden om in het minimum te komen. We bepalen dus uit (9) de afgeleide van $\rho[k]$ naar de adaptieve coëfficiënten als volgt:

$$\frac{\delta\rho[k]}{\delta w_i[k]} = 2E\left\{r[k]\frac{\delta r[k]}{\delta w_i[k]}\right\} \text{ voor } i = 0, \cdots N-1 \tag{12}$$

waarin de afgeleide $\frac{\delta r[k]}{\delta w_i[k]}$ volgt uit (5). Dit geeft:

$$\frac{\delta r[k]}{\delta w_i[k]} = -x[k-i] \text{ voor } i = 0, \cdots N-1. \tag{13}$$

Samengevat kunnen we (12) als de volgende vector schrijven:

$$\underline{\nabla}[k] = -2E\{r[k]\underline{\mathbf{x}}[k]\} \tag{14}$$

waarin $\underline{\nabla}[k]$ de gradiënt voorstelt van de gemiddelde kwadratische waarde $\rho[k]$. Deze gradiënt kan dus direct worden uitgedrukt in de meetbare signalen $\underline{\mathbf{x}}[k]$ en $r[k]$, zodat de adaptatie van het digitale filter gebaseerd kan worden op deze gradiënt door iteratieve aanpassing van de filter coëfficiënten volgens de formule:

$$\underline{\mathbf{w}}[k+1] = \underline{\mathbf{w}}[k] - \alpha\underline{\nabla}[k] \tag{15}$$

waarin α de adaptatie constante is. Door iteratieve toepassing van uitdrukking (15) wordt de waarde van $\rho[k]$ geminimaliseerd, hetgeen een minimale gemiddelde kwadratische fout van het residu signaal betekent.
In praktische toepassingen zullen we moeten werken met een schatting van $\underline{\nabla}[k]$, hetgeen altijd slechts een benadering is van de verwachtingswaarde $-2E\{r[k]\underline{\mathbf{x}}[k]\}$. Een hele ruwe schatting voor deze grootheid is om de momentane waarde zelf te nemen, dus:

$$\underline{\nabla}[k] \approx \hat{\underline{\nabla}}_{LMS}[k] = -2r[k]\underline{\mathbf{x}}[k] \tag{16}$$

waarbij we er in feite van uitgaan dat de middeling gebeurt in het adaptief algoritme zelf. Het kan aangetoond worden dat voor kleine waardes van de adaptatie constante α dit ook zo is (het adaptief filter werkt dan als een soort middelaar). Toepassing hiervan levert het zogenaamde stochastisch iteratie algoritme wat in de literatuur bekend staat als het "Least Mean Square" (LMS) algoritme. Hiervoor geldt:

$$\underline{\mathbf{w}}[k+1] = \underline{\mathbf{w}}[k] + 2\alpha r[k]\underline{\mathbf{x}}[k] \tag{17}$$

oftewel voor iedere adaptieve coëfficiënt afzonderlijk:

$$w_i[k+1] = w_i[k] + 2\alpha r[k]x[k-i] \text{ voor } i = 0, \cdots N-1. \tag{18}$$

Met deze formule voor het LMS algoritme worden de coëfficiënten in kleine stapjes veranderd zolang er nog correlatie is tussen het residu signaal $r[k]$ en het ingangssignaal $x[k]$. Deze verandering zal de correlatie verminderen en zodoende de echo compensatie verbeteren.

3.2 Het dynamisch gedrag van het LMS algoritme

Om het dynamisch gedrag van het LMS algoritme nader te bepalen beschouwen we het verschil tussen de ontvangen echo $e[k]$ en de kopie $\hat{e}[k]$ als functie van de tijd. Uit de formules (4) en (7) volgt:

$$e[k] - \hat{e}[k] = \underline{\mathbf{x}}^T[k] \cdot \underline{\mathbf{d}}[k] = \underline{\mathbf{d}}^T[k] \cdot \underline{\mathbf{x}}. \qquad (19)$$

Voor de verwachtingswaarde van het kwadraat hiervan geldt dan:

$$\varepsilon[k] \doteq E\left\{(e[k] - \hat{e}[k])^2\right\} = E\left\{\underline{\mathbf{d}}^T[k] \cdot \underline{\mathbf{x}}[k]\underline{\mathbf{x}}^T[k] \cdot \underline{\mathbf{d}}[k]\right\} \qquad (20)$$

Als we nu veronderstellen dat het adaptief proces niet al te snel verloopt dan kunnen we stellen dat de adaptieve coëfficiënten $w_i[k]$ en het ingangssignaal $x[k]$ onafhankelijk van elkaar zijn, zodat we deze uitdrukking kunnen herschrijven als:

$$\varepsilon[k] = E\left\{\underline{\mathbf{d}}^T[k] \cdot E\left\{\underline{\mathbf{x}}[k]\underline{\mathbf{x}}^T[k]\right\} \cdot \underline{\mathbf{d}}[k]\right\}. \qquad (21)$$

De matrix $E\left\{\underline{\mathbf{x}}[k]\underline{\mathbf{x}}^T[k]\right\}$ staat bekend als de autocorrelatiematrix $\mathbf{R}$ van het ingangssignaal $x[k]$, zodat we voor $\varepsilon[k]$ het volgende krijgen:

$$\varepsilon[k] = E\left\{\underline{\mathbf{d}}^T[k] \cdot \mathbf{R} \cdot \underline{\mathbf{d}}[k]\right\}. \qquad (22)$$

In datatransmissie hebben we vaak te maken met een data ingangssignaal $x[k]$ dat op een bepaalde manier gecodeerd is in bijvoorbeeld +1 of -1 die elk een gelijke kans van optreden hebben. Voor dat geval heeft de autocorrelatiematrix $\mathbf{R}$ een heel eenvoudige structuur. Deze is dan gelijk aan de eenheidsmatrix $\mathbf{I}$. Bovenstaande formule gaat dan over in:

$$\varepsilon[k] = E\left\{\underline{\mathbf{d}}^T[k] \cdot \underline{\mathbf{d}}[k]\right\}. \qquad (23)$$

Door nu de uitdrukking van het residu signaal $r[k-1]$ volgens formule (8) (met index k vervangen door $k-1$) in te vullen in het LMS algoritme volgens (17) (met index k vervangen door $k-1$) krijgen we:

$$\underline{\mathbf{w}}[k] = \underline{\mathbf{w}}[k-1] + 2\alpha\underline{\mathbf{x}}[k-1]\left(\underline{\mathbf{x}}^T[k-1]\underline{\mathbf{d}}[k-1] + s[k-1]\right) \qquad (24)$$

oftewel voor de differentie vector:

$$\begin{aligned}\underline{\mathrm{d}}[k] &= \underline{\mathrm{d}}[k-1] - 2\alpha\underline{\mathrm{x}}[k-1]\left(\underline{\mathrm{x}}^T[k-1]\underline{\mathrm{d}}[k-1] + s[k-1]\right) \\ &= \left(I - 2\alpha\underline{\mathrm{x}}[k-1]\underline{\mathrm{x}}^T[k-1]\right)\underline{\mathrm{d}}[k-1] - 2\alpha\underline{\mathrm{x}}[k-1]s[k-1]. \quad (25)\end{aligned}$$

Substitutie hiervan in uitdrukking (23) levert:

$$\begin{aligned}\varepsilon[k] &= E\left\{\left(\underline{\mathrm{d}}^T[k-1]\left(I - 2\alpha\underline{\mathrm{x}}[k-1]\underline{\mathrm{x}}^T[k-1]\right) - 2\alpha s[k-1]\underline{\mathrm{x}}^T[k-1]\right)\cdot\right. \\ &\left.\left(\left(I - 2\alpha\underline{\mathrm{x}}[k-1]\underline{\mathrm{x}}^T[k-1]\right)\underline{\mathrm{d}}[k-1] - 2\alpha s[k-1]\underline{\mathrm{x}}[k-1]\right)\right\} \quad (26)\end{aligned}$$

De kruisproducten leveren ook hier weer de waarde nul op, omdat x en s onafhankelijk zijn en omdat de gemiddelde waarde van s nul verondersteld wordt, zodat deze formule overgaat in:

$$\begin{aligned}\varepsilon[k] &\approx E\left\{\underline{\mathrm{d}}^T[k-1]\left(I - 4\alpha E\left\{\underline{\mathrm{x}}[k-1]\underline{\mathrm{x}}^T[k-1]\right\}\right.\right. \\ &\left.\left. +4\alpha^2 E\left\{\underline{\mathrm{x}}[k-1]\cdot\underline{\mathrm{x}}^T[k-1]\underline{\mathrm{x}}[k-1]\cdot\underline{\mathrm{x}}^T[k-1]\right\}\right)\underline{\mathrm{d}}[k-1]\right\} \\ &+4\alpha^2 E\left\{s^2[k]\right\}E\left\{\underline{\mathrm{x}}[k-1]\underline{\mathrm{x}}^T[k-1]\right\}. \quad (27)\end{aligned}$$

Voor het eerder genoemd type datasignaal van +1 en -1 geldt dat de autocorrelatiematrix gelijk is aan de eenheidsmatrix, dus:

$$E\left\{\underline{\mathrm{x}}[k-1]\underline{\mathrm{x}}^T[k-1]\right\} = I \quad (28)$$

en verder dat

$$\underline{\mathrm{x}}^T[k-1]\cdot\underline{\mathrm{x}}[k-1] = \sum_{i=0}^{N-1} x^2[k-1-i] = N. \quad (29)$$

Bovendien schrijven we voor het gemiddeld vermogen van het signaal $s[k]$ $E\left\{s^2[k]\right\} = P_s$ zodat bovenstaande formule overgaat in:

$$\varepsilon[k] = \left(1 - 4\alpha + 4\alpha^2 N\right)E\left\{\underline{\mathrm{d}}^T[k-1]\cdot\underline{\mathrm{d}}[k-1]\right\} + 4\alpha^2 N P_s. \quad (30)$$

Combinatie van (23) met deze uitdrukking geeft

$$\varepsilon[k] = \left(1 - 4\alpha + 4\alpha^2 N\right)\varepsilon[k-1] + 4\alpha^2 N P_s. \quad (31)$$

Verder uitschrijven hiervan levert de volgende verhouding van $\varepsilon[k]$ ten opzichte van het vermogen van het signaal $s[k]$:

$$\frac{\varepsilon[k]}{P_s} = \left(1 - 4\alpha + 4\alpha^2 N\right)^k \cdot \frac{\varepsilon[0]}{P_s} + 4\alpha^2 N \sum_{i=0}^{k-1} \left(1 - 4\alpha + 4\alpha^2 N\right)^i \tag{32}$$

Deze uitdrukking beschrijft volledig het dynamisch gedrag van een echo canceller gebaseerd op het LMS algoritme voor data transmissie signalen.

3.3 Het convergentiegedrag van het LMS algoritme

Enkele conclusies die we uit (32) kunnen trekken zijn:

1. Convergentie treedt alleen maar op als voldaan wordt aan:

$$|1 - 4\alpha + 4\alpha^2 N| < 1 \quad \text{oftewel als} \quad 0 < \alpha < \frac{1}{N}. \tag{33}$$

2. Als we in het convergentiegebied zitten dan zal na convergentie, dus voor $k \to \infty$, uitdrukking (32) overgaan in:

$$\frac{\varepsilon[\infty]}{P_s} = 4\alpha^2 N \cdot \frac{1}{1 - 1 + 4\alpha - 4\alpha^2 N} = \frac{\alpha N}{1 - \alpha N} \tag{34}$$

   zodat dan de verhouding van de rest echo $\varepsilon[\infty]$ ten opzichte van het vermogen van ontvangen signaal $s[k]$ wordt gegeven door:

$$\delta \doteq 10\log_{10}\frac{\varepsilon(\infty)}{P_s} = 10\log_{10}\frac{\alpha N}{1 - \alpha N} \text{ [dB]}. \tag{35}$$

   Voor kleine waardes van $\alpha \ll 1/N$ gaat deze uitdrukking over in:

$$\delta \approx 10\log_{10}\alpha N. \tag{36}$$

3. In het begin van het convergentieproces kan de convergentiesnelheid worden gekarakteriseerd door ν_{20} wat het aantal iteraties voorstelt dat nodig is om de rest echo met 20 dB te verminderen. Uit (32) volgt dan:

$$\nu_{20} = \frac{-2}{\log_{10}\left(1 - 4\alpha + 4\alpha^2 N\right)}. \tag{37}$$

   Voor kleine waardes van $\alpha \ll 1/N$ is deze uitdrukking te vereenvoudigen tot:

$$\nu_{20} \approx \frac{1.15}{\alpha}. \tag{38}$$

De uitdrukkingen (36) en (38) geven het belang van de adaptatie constante α aan. Deze parameter bepaalt zowel de snelheid van convergentie als de minimale hoeveelheid rest echo (de zogenaamde bodem). In Fig. 4 is dit nog eens aangegeven. In deze figuur is op dB schaal weergegeven de verhouding $P(e-\hat{e})/P(s)$ als functie van de tijd variabele k. Hierin stelt $P(e-\hat{e})$ het vermogen (gemiddeld over een kort tijdsinterval) voor van het signaal $e[k]-\hat{e}[k]$ en $P(s)$ stelt het vermogen voor van het signaal $s[k]$. De verhouding $P(e-\hat{e})/P(s)$ geeft aan in welke mate het signaal $s[k]$ wordt verstoord door de rest echo $e[k]-\hat{e}[k]$. Na convergentie bereikt deze grootheid een bepaalde bodem δ. In de praktijk is het wenselijk dat deze kleiner is dan -20 dB. Dit is weergegeven in curve 1 uit Fig. 4 (adaptatie constante $\alpha = \alpha_1$). Door nu de adaptatie constante een factor twee te vergroten ($\alpha = \alpha_2 = 2\cdot\alpha_1$) zien we dat de snelheid met een factor twee toeneemt, maar dat de bodem met een factor twee verslechtert namelijk van -20dB naar -17dB. Dit is weergegeven in dezelfde figuur in curve 2. Deze uitruil van hogere snelheid tegen slechtere bodem of omgekeerd volgt ook direct uit de formules (36) en (38) als we veronderstellen dat geldt $\alpha \ll 1/N$.

Conclusie:
Het LMS algoritme wordt gegeven door:

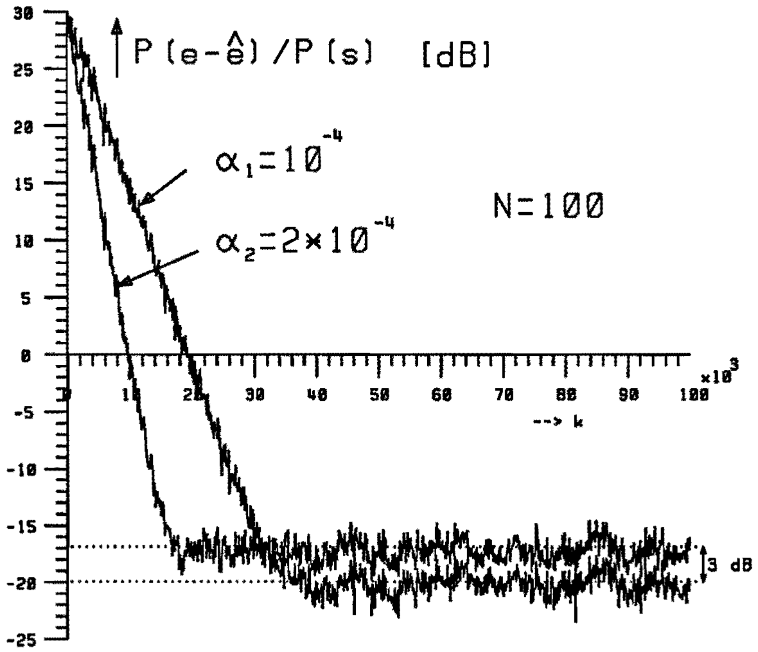
$$w_i[k+1] = w_i[k] + 2\alpha r[k]x[k-i] \quad \text{voor } i = 0, 1, \cdots N-1 \tag{39}$$

en is eenvoudig realiseerbaar. De snelheid en nauwkeurigheid (bodem) voor kleine adaptatie constante ($\alpha \ll 1/N$), worden gegeven door:

$$\nu_{20} \approx \frac{1.15}{\alpha} \qquad \delta \approx \alpha N. \tag{40}$$

4 Verschillende toepassingen van adaptieve filters

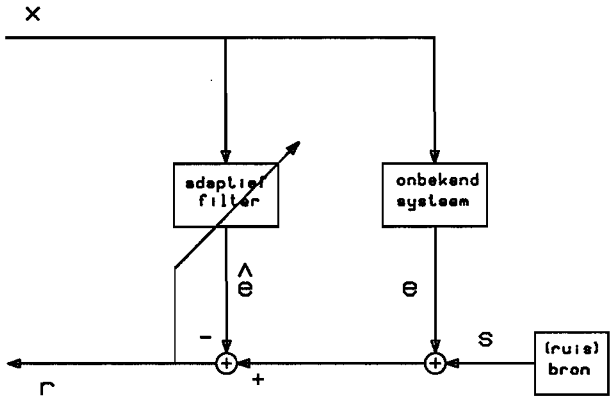
De bedoeling van deze paragraaf is om te laten zien dat er verschillende soorten van adaptieve filters bestaan, die elk weer hun eigen specifieke problemen en oplossingen hebben. We geven hier een (zeker niet volledig) overzicht. Elke subparagraaf behandelt een andere soort, waarbij telkens een of meerdere toepassingen gegeven worden.



Figuur 4: *Convergentie curves van een echo canceller gebaseerd op het LMS algoritme*

4.1 Systeem identificatie

Het principe schema voor de toepassing van een adaptief filter voor systeem identificatie is getekend in Fig. 5. Het signaal x is zowel het ingangssignaal

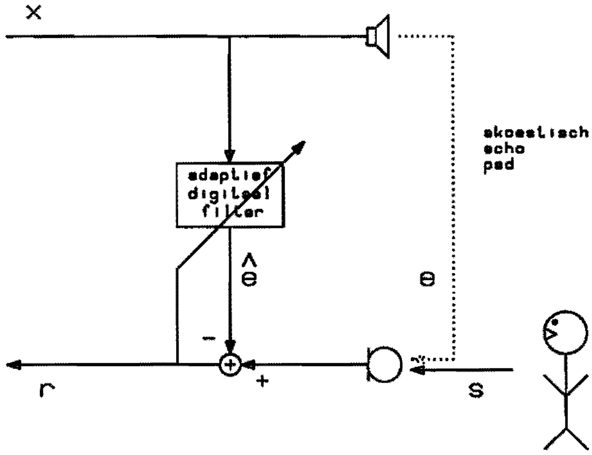


Figuur 5: *Principe schema van systeem identificatie met adaptief filter*

van het onbekend systeem als van het adaptief filter. Om het foutsignaal $e - \hat{e}$ te verkleinen probeert het adaptief filter de impulsresponsie van het onbekende systeem na te maken (te identificeren). Het bijregelen gebeurt door gebruik te maken van het residu signaal $r = e - \hat{e} + s$, waarin s een signaal is dat onafhankelijk is van x. In de eindtoestand, als het adaptief filter is ingeregeld, dan zal gelden: $r \approx s$ omdat $\hat{e} \approx e$.

1. <u>Echo canceller voor datatransmissie</u>
   De adaptieve echo canceller, zoals die is behandeld in paragraaf 2, valt in de klasse van adaptieve filters die bedoeld zijn voor systeem identificatie.

2. <u>akoestische echo canceller</u>
   Het basisschema van een akoestische echo canceller is getekend in Fig. 6. We moeten hierbij denken aan bijvoorbeeld audio teleconferentie. Hierbij voeren twee groepen mensen een bespreking, waarbij iedere groep zich in een andere ruimte bevindt (bijvoorbeeld in een andere plaats of ander land). Een ander voorbeeld is de luidsprekende telefoon, waarbij het de bedoeling is om telefoongesprekken te kunnen



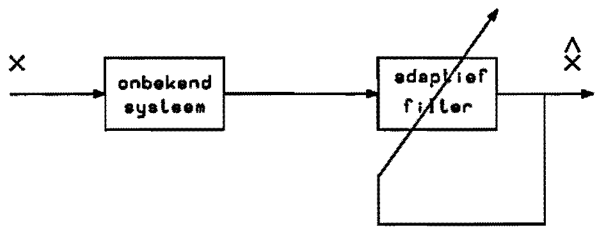
Figuur 6: *Basis schema van een akoestische echo canceller*

voeren zonder daadwerkelijk de telefoon van de haak te hoeven nemen. Dit kan bijvoorbeeld gewenst zijn in de auto.

De werking van de akoestische canceller is als volgt:
Het spraak signaal x van de "veraf" spreker reflecteert via een akoestisch echo pad. Dit resulteert in een echo signaal e. Het akoestisch echo pad, dat in principe onbekend is en kan variëren in de tijd, kan een lengte hebben van een paar honderd milliseconden. Samen met het spraak signaal s, van de spreker in de betreffende ruimte, komt dit echo signaal aan bij de microfoon. Het adaptief filter maakt gebruik van een model van het akoestisch echo pad en maakt een kopie $\hat{e}$ van het echo signaal e. De akoestische canceller onderdrukt dus de echo's van het spraak signaal x van de spreker in de andere ruimte. Theoretisch is het residu signaal $r = s + e - \hat{e}$ in de ingeregelde toestand ongeveer gelijk aan het spraak signaal s.

4.2 Inverse modellering

Het principe schema is getekend in Fig. 7. Het ingangssignaal x wordt vervormd door het onbekend systeem. In de praktijk zal dit vervormde signaal meestal ook nog verstoord worden door een ruissignaal, maar vanwege de eenvoud hebben we dat hier weggelaten. Het adaptief filter probeert een zo goed mogelijke schatting $\hat{x}$ van het ingangssignaal x te maken. In tegenstelling tot bijvoorbeeld de echo canceller hebben we hier

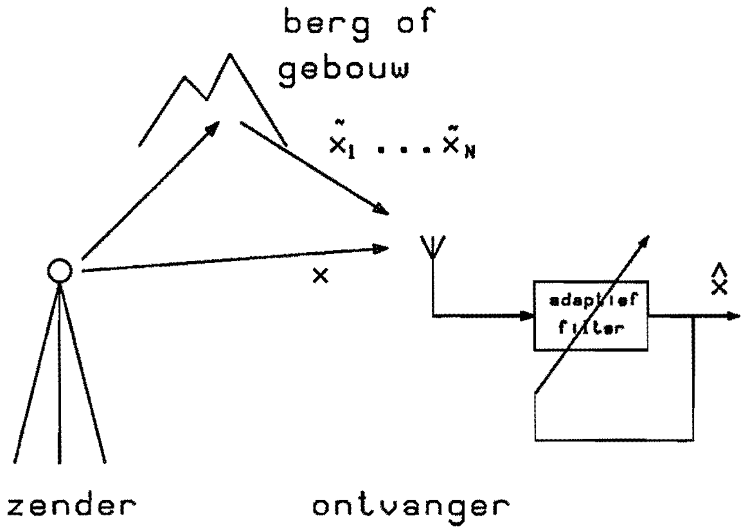


Figuur 7: *Principe schema van inverse modellering*

alleen een verstoord signaal beschikbaar en geen zuiver referentie signaal waarop de regeling gebaseerd kan worden. Het regelsignaal moet dus afgeleid worden uit het ontvangen signaal zelf. Verder hoeft men meestal niet gehele verstoringen te verwijderen, dat zou bovendien onmogelijk zijn (frequentie komponenten die helemaal verzwakt zijn kunnen niet hersteld worden). Het voldoet meestal om die effecten te verwijderen die slecht zijn voor de ontvanger.

1. Adaptieve egalisator bij data transmissie
   Hier hebben we te maken met data transmissie over geschakelde telefoon lijnen waar bij iedere verbinding een ander kanaal gebruikt kan worden. Het kanaal zal het verzonden data signaal verstoren. Dit effect kan verwijderd worden door een adaptieve egalisator. De reden om hier een adaptieve egalisator te kiezen is dat in de meeste gevallen het kanaal totaal onbekend is, terwijl het bovendien nog varieert in de tijd.

2. Multipad
   In Fig. 8 is een schets getekend van multipad effecten bij FM radio uitzending. In FM transmissie wordt multipad voortplanting veroorzaakt door reflecties van het verzonden signaal x door gebouwen of bergen $(\tilde{x}_1, \cdots, \tilde{x}_N)$. Al deze signalen komen bij de ontvanger aan met verschillende vertragingen. Dit effect kan beschreven worden als een soort filterbewerking tussen de uitgang van de zender en de ingang van de ontvanger. De effecten op FM transmissie in de aanwezigheid van ongewenste fase en amplitude modulatie leidt, na detectie, tot hinderlijke niet lineaire vervorming van de audio informatie. In het ideale geval zullen we een invers filter moeten toepassen om deze



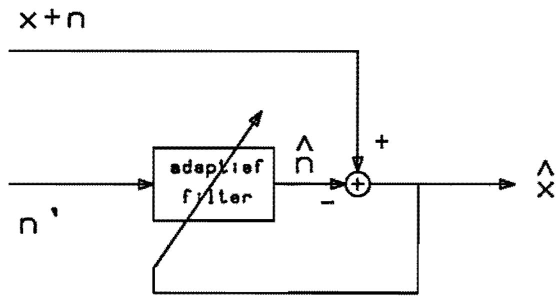
Figuur 8: *Schets van multipad effecten bij FM radio uitzending*

vervorming te verwijderen. We hebben hier een adaptief filter nodig omdat het multipad niet bekend is in de ontvanger, en bovendien kan het variëren in de tijd.

4.3 Interferentie compensatie

Het principe schema is getekend in Fig. 9. Het gewenste signaal x wordt verstoord door een additief ruissignaal n. Verder is een gefilterde versie n' van diezelfde ruisbron n aanwezig. Het doel van het adaptief filter is om met behulp van deze gefilterde versie n' een schatting $\hat{n}$ te maken die zo goed mogelijk lijkt op het ruis signaal n. Het restsignaal $\hat{x}$ zal dan zo goed mogelijk op het ingangssignaal x lijken.

1. <u>ruis compensator</u>
   We gaan hierbij uit van de situatie waarbij een piloot door middel van de radio communiceert vanuit een cockpit van een vliegtuig. Hier is veel achtergrond lawaai (ruis) aanwezig van de machines van het vliegtuig. Deze ruis n wordt door dezelfde microfoon opgevangen waarin de piloot spreekt (spraaksignaal x). Door nu een tweede microfoon in de cockpit op een geschikte positie te plaatsen kan een afgeleide ruiscomponent n' van n verkregen worden waar geen spraak signaal x bij zit. Deze ruis n' kan met behulp van een adaptief filter



Figuur 9: *Principe schema van interferentie compensatie*

gefilterd worden tot $\hat{n}$ en van het ontvangen signaal $x + n$ worden afgetrokken om zodoende de interferentie te verminderen. We hebben hier een adaptief filter nodig omdat de frequentie en intensiteit van de ruis onbekend is en varieert met de snelheid van de machine, de positie van het hoofd van de piloot, enzovoorts.

Conclusie:
In deze paragraaf zijn drie verschillende soorten adaptieve filters behandeld, en wel voor:

1. systeem identificatie
2. inverse modellering en
3. interferentie compensatie.

5 Enkele fundamentele kenmerken

Het zal duidelijk zijn dat de voorgaande paragraaf nog uitgebreid kan worden met talloze voorbeelden. De gegeven voorbeelden volstaan echter om enkele fundamentele kenmerken van adaptieve digitale signaalbewerkings technieken te herkennen. Deze kenmerken zijn:

1. Om in staat te zijn om uit te rekenen welke actie nodig is van het adaptief systeem is het nodig om enige a priori kennis van het onbekende systeem te gebruiken (zonder enige voorkennis kunnen we überhaubt niet adapteren!). Zo maken we bijvoorbeeld bij de echo canceller, die beschreven is in paragraaf 2 en 3, gebruik van het feit

dat het echo pad gemodelleerd kan worden met een transversaal filter, zoals dat in Fig. 3 is aangegeven.

2. Een kwaliteitscriterium (in dit college vanwege de eenvoud altijd een kleinste kwadratische fouten criterium) is nodig. Dit criterium specificeert waarnaar het adaptief systeem moet streven, dat wil zeggen het optimum waarnaar het moet zoeken als de parameters worden berekend.

3. Een algoritme moet gevonden worden dat in staat is om uit de beschikbare signalen de waardes van de parameters te schatten die optimaal voldoen aan het kwaliteitscriterium.

4. Het adaptief systeem moet een signaalbewerkings eenheid hebben die voorzieningen heeft om veranderende parameters op te slaan (dit is vaak de motivatie om voor een digitale implementatie te kiezen). Hierbij is de gebruikte structuur van het adaptief filter van belang.

Als een van deze vier kenmerken verandert dan resulteert dat meestal in een ander adaptief systeem met andere eigenschappen. De eerste twee zijn alleen maar impliciet aanwezig maar hebben een grote invloed op de andere twee. Vandaar dat we in de volgende twee paragrafen nog kort in zullen gaan op andere algoritmen en andere structuren.

6 Andere Algoritmen

In paragraaf 3 is het LMS algoritme behandeld. In deze paragraaf zullen we kort enkele andere algoritmen behandelen.

6.1 Sign–algoritme

In de meeste toepassingen is complexiteit een heel belangrijke factor. Soms is het aantal rekenkundige bewerkingen per tijdseenheid dat nodig is voor het LMS algoritme volgens (17) te complex. In deze toepassingen is men vaak bereid om convergentie eigenschappen van het LMS algoritme uit te ruilen tegen complexiteit.
Een voorbeeld van zo'n algoritme is het Sign–algoritme. Evenals het LMS algoritme wordt hier een schatting van de gradiënt gebruikt voor de update. Echter de schatting is minder nauwkeurig dan bij het LMS algoritme. In plaats van de schatting volgens (16) wordt hier alleen gebruik gemaakt van

het teken ("sign") van het residu signaal, zodat de schatting van de gradiënt wordt gegeven door:

$$\underline{\hat{\nabla}}_{sign}[k] \approx -2\text{sign}(r[k])\underline{\mathbf{x}}[k]. \tag{41}$$

De echte vermenigvuldiging $r[k]x[k-i]$ wordt hier voor elk element van de gradiënt vector vervangen door $\text{sign}(r[k])x[k-i]$, hetgeen een veel eenvoudigere bewerking is dan vermenigvuldigen. Verder kan aangetoond worden dat dit Sign–algoritme (weliswaar onder strengere voorwaarden dan het LMS–algoritme) convergeert, maar dat de convergentie eigenschappen slechter zijn [5].

Conclusie:
Het Sign–algoritme wordt gegeven door:

$$\underline{\mathbf{w}}[k+1] = \underline{\mathbf{w}}[k] + 2\alpha\text{sign}(r[k])\underline{\mathbf{x}}[k]. \tag{42}$$

en is qua complexiteit eenvoudiger dan het LMS–algoritme, maar dit gaat wel ten koste van convergentie eigenschappen.
Voor een uitgebreide vergelijking van het Sign–algoritme met het LMS algoritme verwijzen we naar [5].

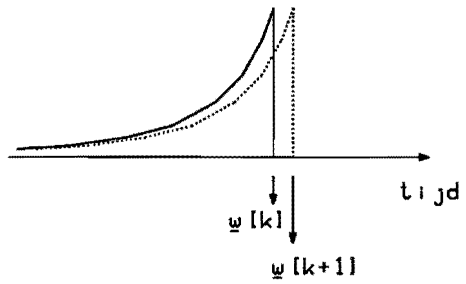
6.2 "Recursive Least Squares" (RLS) methode

In paragraaf 3.3 hebben we de convergentie eigenschappen van het LMS algoritme afgeleid. We zijn hierbij uitgegaan van de differentievergelijking zoals we die in paragraaf 3.2 hebben afgeleid. Hierbij hebben we op dat moment voor de eenvoud gebruik gemaakt van het feit dat in veel datatransmissie toepassingen de autocorrelatiematrix $\mathbf{R} = E\{\underline{\mathbf{x}}[k]\underline{\mathbf{x}}^T[k]\}$ gelijk is aan de eenheidsmatrix $\mathbf{I}$. Het kan aangetoond worden dat de convergentie eigenschappen van het LMS algoritme verslechteren als dit niet meer geldt. Door nu in de update term van het LMS algoritme (17) te vermenigvuldigen met de inverse autocorrelatie matrix $\mathbf{R}^{-1}$ worden de convergentie eigenschappen onafhankelijk van de autocorrelatiefunctie van het ingangssignaal. We krijgen dan het LMS/Newton algoritme:

$$\underline{\mathbf{w}}[k+1] = \underline{\mathbf{w}}[k] + 2\alpha\mathbf{R}^{-1}r[k]\underline{\mathbf{x}}[k]. \tag{43}$$

Bij dit algoritme gaan we er echter wel vanuit dat $\mathbf{R}^{-1}$ bekend is. In het algemeen zal dit echter niet het geval zijn, terwijl bovendien in veel gevallen het ingangssignaal $x[k]$ niet stationair is, zodat $\mathbf{R}$ langzaam zal variëren in

de tijd op een onbekende manier.
Het "Recursive Least Squares" (RLS) algoritme lost dit probleem op door op een recursieve manier de nieuwe matrix $\mathbf{R}[k]$ uit te rekenen vanuit vorige waarden $\mathbf{R}[i]$ met $i < k$. Om niet stationaire effecten te vermijden wordt een negatief exponentieel venster toegepast om data van lang geleden te "vergeten" (zie Fig. 10). In de literatuur zijn implementaties van het



Figuur 10: *Update procedure met RLS methode*

RLS algoritme bekend die een complexiteit hebben die lineair evenredig is met de filterlengte (dit is het zogenaamde "Fast Recursive Least Square" (FRLS) algoritme) [13,15]. In het algemeen kunnen we snellere convergentie verkrijgen met de RLS methode dan bij LMS, hetgeen belangrijk is voor de opstart fase van het adaptief filter, maar ook bij het volgen van variaties in systeem veranderingen. Hét grote probleem van de FRLS algoritmen op dit moment is de numerieke stabiliteit [15].

Conclusie:
Het RLS algoritme is, in tegenstelling tot het LMS algoritme, in staat om de convergentie eigenschappen onafhankelijk te maken van de autocorrelatie functie van het ingangssignaal. Dit betekent in het algemeen dat deze beter zijn. Het RLS algoritme heeft echter een zeer grote (in de praktijk vaak niet toelaatbare) complexiteit.
FRLS algoritmen hebben de mogelijkheid om deze complexiteit te reduceren tot een acceptabel nivo, maar ze hebben echter nog veel numerieke problemen.
Voor een uitgebreidere behandeling van met name FRLS verwijzen we naar [13,15].

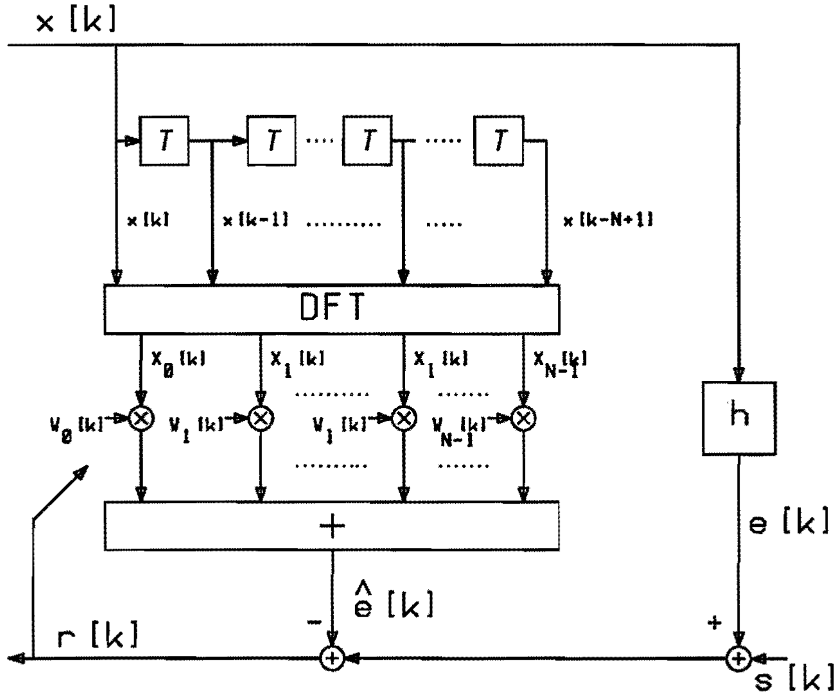
6.3 Frequentie domein adaptief filter

In de vorige subparagraaf 6.2 hebben we aangegeven dat de convergentie eigenschappen van het LMS algoritme verslechteren als het ingangssignaal autocorrelatie heeft. Met andere woorden de de autocorrelatiematrix $\mathbf{R}$ is niet meer gelijk aan de eenheidsmatrix $\mathbf{I}$. Bij de RLS methode hebben we gezien dat we deze autocorrelatie ongedaan konden maken door in de update term te vermenigvuldigen met de inverse autocorrelatiematrix $\mathbf{R}^{-1}$. In deze subparagraaf zullen we een ander algoritme geven waarbij het decorreleren plaatsvindt in het frequentie domein.

In paragraaf 3 hebben we aangenomen dat we te maken hadden met een ingangssignaal waarvan de autocorrelatiematrix gelijk was aan de eenheidsmatrix. Dit houdt in dat alle frequentie componenten van zo'n ingangssignaal evenveel vermogen hebben, oftewel dat het spectrum vlak is. Als we nu te maken hebben met een ingangssignaal waarvan de frequentie componenten niet allemaal hetzelfde vermogen hebben, zodat het spectrum van dat ingangssignaal niet vlak is, dan zou toepassing van het LMS algoritme er toe leiden dat de frequentie componenten met veel vermogen snel naar hun eindwaarde zullen regelen terwijl frequentie componenten met weinig vermogen heel langzaam zullen inregelen. Het convergentie gedrag van het LMS algoritme is afhankelijk van de spectrale verdeling van het ingangssignaal. In de meeste praktische gevallen zal dit er toe leiden dat de convergentie eigenschappen van het LMS algoritme verslechteren.

We zullen nu laten zien dat het decorreleren van het ingangssignaal relatief eenvoudig uitgevoerd kan worden door gebruik te maken van een frequentie domein adaptief filter, waarvan in Fig. 11 het basisschema is getekend. Elk nieuw monster $x[k]$ van het ingangssignaal wordt in een vertragingslijn gezet die N monsters lang is. Daarna wordt van deze N monsters de Discrete Fourier Transformatie (DFT) berekend. Elk van de DFT uitgangen correspondeert met één specifieke frequentie band. De DFT kan beschouwd worden als een aantal banddoorlatende filters van frequenties tussen nul en halverwege de bemonsterings frequentie. Omdat de uitgangen van de DFT allemaal verschillende frequentie bandjes zijn, kunnen we er van uitgaan dat deze frequentie componenten ongeveer orthogonaal (onafhankelijk) van elkaar zijn. (Dit is niet exact zo omdat er altijd enige "leakage" is tussen de verschillende frequentie componenten.) De DFT fungeert hier dus als een soort "Orthogonale Transformatie".

Het niet vlak zijn van het spectrum van het ingangssignaal kan nu



Figuur 11: *Basis schema van frequentie domein adaptief filter*

eenvoudig gecorrigeerd worden met het frequentie domein adaptief filter door elke frequentie component te normeren met zijn eigen vermogen, zoals is weergegeven in het volgende algoritme:

$$W_l[k+1] = W_l[k] + \frac{2\alpha}{\hat{P}_{X_l}[k]} r[k] X_l^*[k] \quad \text{voor } l = 0, 1, \cdots N-1. \tag{44}$$

Hierin is $*$ de complex geconjugeerde omdat we hier te maken hebben met complexe signalen. Verder is $\hat{P}_{X_l}[k]$ een schatting van het vermogen P_{X_l} in elke frequentie component afzonderlijk. Dit vermogen is gedefinieerd als de mathematische verwachting van de absolute waarde in het kwadraat, oftewel in formulevorm:

$$P_{X_l} = E\{|X_l[k]|^2\}. \tag{45}$$

We hebben op deze manier een algoritme gemaakt dat iedere frequentie component zodanig normeert met zijn eigen vermogen dat de convergentiesnelheid voor alle componenten dezelfde zal zijn. Dit ondanks het feit dat we eventueel een gecorreleerd ingangssignaal hebben waarvan het spectrum dus niet vlak is. Het algoritme (44) heeft evenals de RLS methode decorrelerende eigenschappen. Hiermee kunnen de convergentie eigenschappen voor gecorreleerde ingangssignalen verbeterd worden in vergelijking met het LMS algoritme.

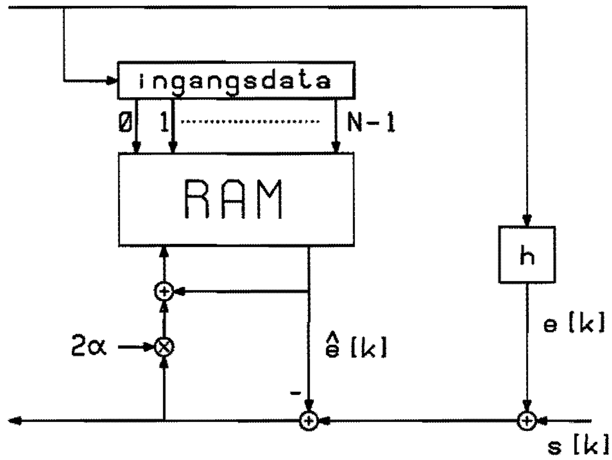
Conclusie:
Door met behulp van een DFT naar het frequentie domein te gaan zijn we in staat om op relatief eenvoudige manier het spectrum van het ingangssignaal te normeren (vlak te maken), hetgeen overeenkomt met het decorreleren van het ingangssignaal.
Voor een uitgebreidere behandeling van dit algoritme wordt verwezen naar [16], waar ook andere orthogonale transformaties worden toegepast.

7 Andere structuren

In het voorgaande zijn we steeds uitgegaan van transversale filter structuren. Hier zullen in het kort enkele andere structuren behandeld worden.

7.1 Table–lookup structuur

In Fig. 12 is een ander type adaptief filter getekend van de zogenaamde "Table–lookup" structuur. Hierbij wordt een combinatie van N binaire



Figuur 12: *Adaptief Table-lookup filter met LMS algoritme.*

ingangsmonsters gebruikt als adres voor een "Random Acces Memory" (RAM). De inhoud van dit adres is het kopiee signaal $\hat{e}[k]$. Als op tijdstip k een bepaald adres wordt gelezen, dan wordt zijn inhoud vervolgens aangepast met een correctieterm, terwijl de inhoud van de overige adressen ongewijzigd blijft. Voor het LMS algoritme is de correctie term $2\alpha \cdot r[k] = 2\alpha \cdot (s[k] + e[k] - \hat{e}[k])$. In de appendix van [3] wordt aangetoond dat dit type filter dezelfde soort van convergentie karakteristieken heeft dan het transversaal filter zoals afgeleid in paragraaf 3.2 . De differentie vergelijking (voor een afleiding wordt verwezen naar genoemde appendix) wordt nu gegeven door:

$$\frac{\varepsilon[k]}{P(s)} = \left(1 - \frac{4\alpha(1-\alpha)}{2^N}\right)^k \cdot \frac{\varepsilon[0]}{P_s} + \frac{4\alpha^2}{2^N} \sum_{i=0}^{k-1} \left(1 - \frac{4\alpha(1-\alpha)}{2^N}\right)^i \tag{46}$$

De bodem δ en snelheid ν_{20} worden nu gegeven door ($\alpha \ll 1$):

$$\begin{aligned} \delta &\approx 10\log_{10}\alpha \quad [\text{dB}] \\ \nu_{20} &= \frac{1.15}{\alpha} 2^N. \end{aligned} \tag{47}$$

Deze uitdrukkingen kunnen als volgt geïnterpreteerd worden. In de Table-lookup structuur wordt elk monster $\hat{e}[k]$ alleen maar verstoord met de onnauwkeurigheid ("ruis") van de inhoud van het adres waar het is opgeslagen. Hiertegenover staat dat bij de transversale structuur N "ruisachtige" coëfficiënten worden gesommeerd, hetgeen een N keer grotere ruis variantie oplevert in de geconvergeerde staat. Van de andere kant wordt bij de Table–lookup structuur elke iteratie slechts de inhoud van een adres opgehoogd, wat resulteert in een convergentie die een factor 2^N trager is in vergelijking met de transversale structuur waar alle N coëfficiënten elke iteratie worden aangepast.

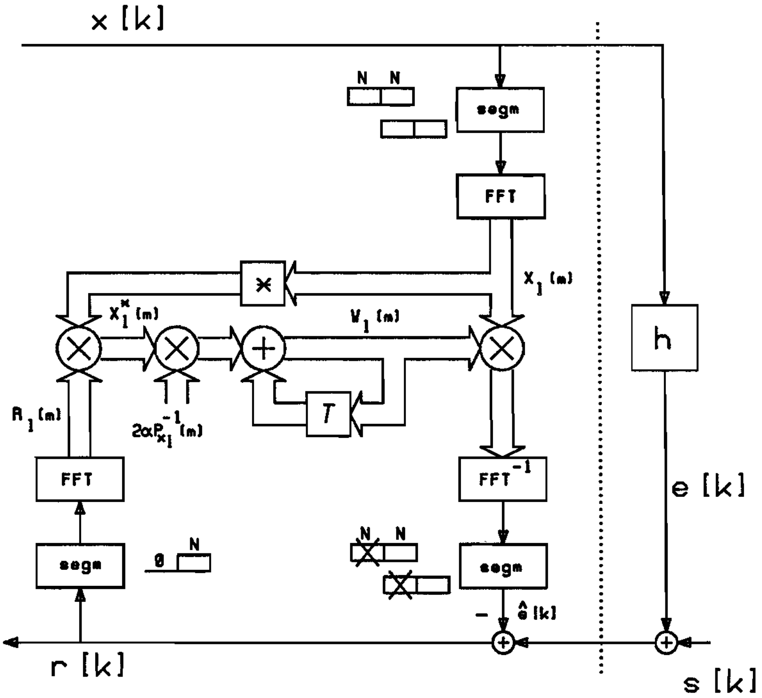
Conclusie:
Toepassing van Table–lookup kan gebeuren als convergentie eigenschappen niet zo kritisch zijn, terwijl geheugens eenvoudig te implementeren zijn. Verder kunnen met Table–lookup, in tegenstelling tot transversale structuren, niet lineaire effecten gecompenseerd worden. Bovendien kan de interne snelheid laag gehouden worden door het gebruik van RAM structuren Voor een verdere beschouwing verwijzen we naar [3].

7.2 Frequentie domein blok adaptief filter

In deze subparagraaf zullen we een adaptief filter behandelen dat, evenals het frequentie domein adaptief filter uit subparagraaf 6.3, het ingangssignaal decorreleerd in het frequentie domein. Verder zal de hier behandelde structuur zodanig zijn dat deze zeer efficient uitgevoerd kan worden. Als mogelijke toepassing voor dit adaptief filter moeten we denken aan de akoestische echo canceller zoals die in subparagraaf 4.1 is behandeld (zie Fig. 6). Uit deze subparagraaf 4.1 kunnen we concluderen dat de twee belangrijkste problemen van adaptieve transversale filters voor dit soort toepassingen zijn:

1. Het aantal coëfficiënten N dat nodig is voor het adaptief filter om het akoestisch echo pad te modelleren is zeer groot. Dat varieert tussen de 500 en 2000. Alleen de convolutie van het ingangssignaal met de coëfficiënten van het adaptief filter (de eigenlijke filterbewerking) zou al leiden tot een zeer hoge complexiteit. Het aantal rekenkundige bewerkingen om N uitgangsmonsters uit te rekenen is evenredig met N^2. Dit geven we aan met het volgende orde van grootte symbool: $O(N^2)$.

2. Het ingangssignaal, wat een spraak signaal is, is een gecorreleerd signaal. Bij toepassing van een LMS algoritme zou dit leiden tot



Figuur 13: *Vereenvoudigd schema van frequentie domein blok adaptief filter*

slechte convergentie eigenschappen.

Deze twee problemen kunnen tegelijkertijd aangepakt worden met een frequentie domein blok adaptief filter, waarvan een vereenvoudigd schema is getekend in Fig. 13.

Het aantal rekenkundige bewerkingen van een convolutie kan worden teruggebracht door deze uit te voeren in het frequentie domein als vermenigvuldiging, waarbij de transformatie wordt gedaan met behulp van de Fast Fourier Transformatie (FFT). We moeten dan wel werken op blok basis, hetgeen in Fig. 13 is aangegeven door de segmenterings bewerkingen ("segm"). Hierbij bewerken we tegelijkertijd een blok van monsters van het ingangssignaal die een blok van uitgangssignaal monsters produceren. Het aantal rekenkundige bewerkingen wordt op deze manier teruggebracht van $O(N^2)$ tot $O(N\log N)$ voor het frequentie domein blok adaptief filter.
Verder kunnen we evenals bij het frequentie domein adaptief filter, zoals behandeld in paragraaf 6.3, elke frequentie component apart normeren met een schatting van het vermogen. In Fig. 13 gebeurt dit door de term $P_{X_l}^{-1}(m)$. Op deze manier kunnen we het ingangssignaal decorreleren.

Conclusie:
Met het FDAF kunnen twee problemen in een keer worden aangepakt, namelijk reductie van de complexiteit en decorrelatie van het ingangssignaal. Voor een uitgebreidere beschouwing wordt verwezen naar [14].

8 Slotopmerking

Zoals reeds in de titel is aangegeven is voorgaande slechts een korte inleiding op het gebied van de adaptieve digitale signaalbewerking. Op dit moment wordt er nog veel onderzoek gedaan op allerlei gebied van dit onderwerp. We zullen hier afsluiten met het refereren naar een paar boeken op dit gebied.

- [7]: Goed leerboek (voor iets meer gevorderden), tamelijk compleet.
- [8]: Goed leerboek (voor beginners), met een nogal lange introductie waarvan het niet altijd even duidelijk is waar het precies voor nodig is.
- [9]: Een boek met heel veel interessante (echter vaak nog niet uitgewerkte) ideëen. Formules kloppen vaak niet.

- [10]: Goed maar ingewikkeld boek. Hoofdmoot wordt gevormd door lattice structuren.
- [11]: Goed maar ingewikkeld boek.
- [12]: Goed maar ingewikkeld boek. Vrij veel formulewerk.

Referenties

[1] "Digitale Signaalbewerking", A.W.M. van den Enden, N.A.M. Verhoeckx, Delta Press bv. 1987.

[2] "Digital Echo Cancellation for Baseband Data Transmission", N.A.M. Verhoeckx, H.C. van den Elzen, F.A.M. Snijders, P.J. van Gerwen, IEEE Trans. on ASSP, vol. ASSP–27, no. 6, dec. 1979, pp768–781

[3] "Design Considerations for a 144 kbit/s Digital Transmission Unit for the Local Telephony Network" P.J. van Gerwen, N.A.M. Verhoeckx and T.A.C.M. Claasen, IEEE Journal on selected areas in Communications, vol. SAC-2, n0.2, march 1984, pp314–323

[4] An integrated echo canceller for baseband data transmission" P.J. van Gerwen, W.A.M. Snijders and N.A.M. Verhoeckx, Philips Techn. Rev. 39, 102–117, 1980, no. 3/4

[5] "Comparison of the convergence of two Algorithms for Adaptive FIR Digital Filters", T.A.C.M. Claasen, W.F.G. Mecklenbräuker, IEEE Trans. on ASSP, vol ASSP–29, no.3, june 1981, pp 670–678.

[6] "Adaptive Techniques for Signal Processing in Communications" T.A.C.M. Claasen and W.F.G. Mecklenbräuker, IEEE Communications Magazine, Nov. 1985, vol. 23, no.11, pp8–19

[7] "Adaptive Signal Processing, theory and applications", S.T. Alexander, Springer–Verlag, 1986.

[8] "Adaptive Signal Processing", B. Widrow, S.D. Stearns, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1985.

[9] "Adaptive Filters", C.F.N. Cowan, P.M. Grant, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1985.

[10] "Adaptive Filters", M.L. Honig, D.G. Messerschmitt, Kluwer Academic Publishers, 1984.

[11] "Adaptive Filter Theory", S.Haykin, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1986.

[12] "Adaptive Filtering Prediction an Control", G.C. Goodwin, K.S. Sin, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1984.

[13] "Adaptive Digital Filtering and Signal Analysis", M.G. Bellanger, Marcel Dekker Inc., New York, 1987.

[14] "Convergence Analysis of a Frequency Domain Adaptive Filter with Exponential Power Averaging and Generalized Window Function", P.C.W. Sommen, P.J. van Gerwen, H.J. Kotmans, A.J.E.M. Janssen, IEEE Trans. on CAS, vol. CAS–34, no. 7, july 1987, pp 788–798.

[15] "Fast Recursive Least Squares Transversal Filters for Adaptive Filtering", J. Cioffi, T. Kailath, IEEE Trans., vol. ASSP–32, no.2, april 1984, pp 304–337.

[16] "Transform Domain LMS Algorithms", S.S. Narayan, A.M. Peterson, M.J. Narasimha, IEEE Trans. vol. ASSP–31, no.3, june 1983, pp 609–615.

DIGITALE SIGNAALBEWERKING

deel II.5

Computer Aided Design (CAD) voor digitale signaalbewerking

door

A.W.M. van den Enden

Philips Nat. Lab. Eindhoven

Computer Aided Design (CAD) voor digitale signaalbewerking

A.W.M. van den Enden
Philips Natuurkundig Laboratorium WAY 5.95
Postbus 80.000
5600 JA Eindhoven
11-09-1989

Contents

Meer informtatie over:

FDAS, HYPERCEPTION, HYPERSIGNAL en SPW

is te verkrijgen via:

Transfer EDS
Institutenweg 16
7521 PK Enschede

1 Inleiding

Dit dictaat bestaat uit twee hoofdstukken. Het eerste hoofdstuk (hoofdstuk 2) zal een overzicht geven van een aantal CAD software pakketten voor tijd-discrete en digitale signaalbewerking die commercieël verkrijgbaar zijn. Van al deze pakketten is illustratiemateriaal in dit hoofdstuk opgenomen. De adressen waar informatie over de beschreven pakketten kan worden verkregen zijn opgenomen in de Appendix van dit dictaat.

Als een demonstratie van wat (bijna) al deze pakketten kunnen, worden in hoofdstuk 3 een vijftal demonstraties uitgebreid beschreven. De resultaten die in dit hoofdstuk worden gepresenteerd zijn berekend met een programmapakket dat uitsluitend intern bij Philips te gebruiken is. Het is echter erg eenvoudig om dezelfde simulaties met de andere software pakketten te berekenen.

2 Commerciëel verkrijgbare software

In dit hoofdstuk zullen we een aantal commerciëel verkrijgbare software pakketten kort bespreken. Deze lijst is zeker niet volledig en het aantal pakketten neemt snel toe. De volgende programmapakketten zullen kort aan de orde komen (een adressenlijst van de genoemde fabrikanten vindt U in de Appendix van dit dictaat):

1. het IEEE pakket;
2. ILS van Signal Technology;
3. DSPlay van Burr-Brown;
4. DFDP van Atlanta Signal Processors;
5. DADISP van DSP Development Corporation;
6. MONARCH van The Athena Group;
7. FDAS van Momentum Data Systems;
8. DFD van Microcraf Corporation;
9. CAT (Computer Aided Teaching) van EPFL in Lausanne;
10. Digital filter design software for the IBM PC (boek);
11. Signal processing algorithms (boek);
12. HYPERCEPTION van de gelijknamige leverancier
13. HYPERSIGNAL van Loughborough Sound Images (LSI)
14. MATLAB van The MATH Works Inc.
15. SYSNOISE van Dynamic Engneering
16. SPW van Comdisco Systems
17. NAGLIB van de Numerical Algorithms Group
18. DSP Processing Experiments (boek)
19. LABWINDOW/LABVIEW van National Instruments
20. MATHCAD van MathSoft
21. MICRO-CAP III van Spectrum Software

Van alle programmapakketten is illustratiemateriaal bijgevoegd. De belangrijkste verschillen tussen al deze pakketten is de manier waarop de verschillende programma's aangesproken moeten worden. Bijna alle pakketten kunnen met een terminal interactief gebruikt worden. Bij veel fabrikanten is het mogelijk om een demodiskette te verkrijgen, waarop de grote voordelen van de betreffende pakketten getoond worden.

2.1 Het IEEE pakket

Het oudste pakket dat reeds in in 1979 beschikbaar kwam is het zogenaamde IEEE pakket. Het is uitgegeven door het "Digital Signal Processing committee" van de IEEE Acoustics Speech en Signal Processing society (ASSP). De beschikbare programma's worden beschreven in het boek:

Programs for digital signal processing,
ISBN 0-471-05962-5 (clothbound) of
ISBN 0-471-05961-7 (paperbound),
John Wiley and Sons, New York, 1979.

Op de volgende pagina's vindt U welke programma's beschreven worden. Tegen tape kosten zijn deze prgramma's bij het IEEE te bestellen:

IEEE press,
345 East 47-th street ,
New York, NY 10017.
USA

Dit pakket bestaat uit een aantal erg goede en betrouwbare programma's. Alleen de dialoog met de programma's is erg moeizaam. Interactief werken is niet mogelijk.

Contents

2.2 ILS van Signal Technology

Overzicht van functies:

Summary of ILS Programs

+ADF	Perform arithmetic operations.
AFP	Compute and display 3-D area functions.
ANA	Analyze using linear prediction or Burg methods.
API	Analyze using linear prediction and cepstral pitch detection.
+ASG	Assign keyboard, printer and graphics devices.
AVG	Calculate moving or exponential averages.
BOP	Perform binary arithmetic operations.
BPA	Perform best pattern matching of test and reference data.
+CEP	Compute and display cepstrum.
CLA	Label sampled data segments for pitch synchronous analysis.
CNV	Convolve time series and filter impulse response using FFTs.
COL	Define colors and assign them to graphic objects.
COR	Correlate two time series using FFTs.
CPT	Compute and store complex cepstrum.
+CST	Compute and print sampled data statistics.
+CTX	Change context (number of sample data points per frame).
+CUR	Mark displayed data with graphics cursor.
+DFI	Design elliptic, Butterworth and Chebychev filters.
DPM	Display time synchronized speech parameters.
DRE	Display real, complex or magnitude/phase data.
+DSP	Display sampled data.
+DTO	Write data from an ILS file to a user ASCII file.
+FDI	Compute and display spectrum.
FFT	Compute and store Fast Fourier Transform.
+FIL	Select, create or delete data files.
+FLT	Convolve time series and filter impulse response.
FPL	Compute and display linear prediction spectra.
FTR	Compute and display formant tracks on spectrogram.
+GRD	Display grid for spectrum or cepstrum.
+HHH	Print help information for a chosen ILS command.
HIL	Compute and store Hilbert Transform.
HIS	Compute and display histogram and fitted curves.
IAE	Edit speech parameters interactively prior to synthesis.
IDC	Initialize common data arrays for parameter conversion.
IFL	Design ideal filter and store impulse response.
+ILS	Create and initialize ILS common data file.
+INA	Initialize sampled data file header.
LBA	Label sampled data segment.
LBF	Select, create or delete label files.
LCM	List ILS common data file.
+LFI	Design linear phase finite impulse response filter.
LLA	List label data.
LRE	List values of signal processing or feature data.
+LSN	Data acquisition program for D/A playback from disk files.
LWP	Linear warping of matrices for time normalization.
+MDF	Modify values of data points.
+MDX	Multiplex and demultiplex multi-channel data.
MRE	Manipulate and extract features.
+MVF	Move, scale or zero sampled data.
NSI	Simulate noisy data by generating and adding noise.
OCT	Use octave band filtering and weighting curves.
OPN	Open signal processing or feature files.
PAN	Analyze using pitch synchronous linear prediction.
PCO	Analyze principal components (rotation and scaling).
PLR	Display feature scatter and perform factor analysis.
PNS	Synthesize from pitch synchronous analysis data.
+PRM	Prompt for arguments of a chosen ILS command.
+PRT	Print values of sampled data.
QUR	Queue sampled data for feature extraction.
RAN	Compute and store residue signal using linear prediction.
+REC	Data acquisition program for A/D digitization to disk files.
+REG	Perform linear regression or curve fitting.
RSO	Solve roots of linear prediction coefficient polynomials.
SDE	Estimate spectral densities and transfer function.
+SDI	Compute and display 3-D spectra.
SGM	Compute and display spectrogram.
SIF	Analyze using linear prediction and SIFT pitch detector.
SME	Compute and store feature statistics.
SNS	Synthesize from analysis data.
SPL	Compute and display 3-D linear prediction spectra.
SRE	Store sampled data and filters as signal processing data.
SSD	Store signal processing data as sampled data.
SSP	Compute and display linear prediction spectrum.
+TBL	Setup directory table for ILS file system.
+TFU	Generate test functions (e.g. exponential and sine waves).
TLA	Transfer label data meeting a given specification.
TRE	Transfer signal processing or feature data between files.
+TRF	Transfer sampled data between files.
+TRM	Set terminal type, erase or copy screen, display message.
+TSI	Generate standard data for examples and training.
TTL	Transfer, concatenate and label sampled data.
UOP	Perform unary arithmetic operations.
VDI	Evaluate feature data distances for probability of error.
VER	Verify sampled data file header.
VTR	Compute and display area functions on facial cross-section.
WRP	Non-linear warping of matrices for template generation.
+WRT	Write ILS file from user ASCII file.
+XPA	Upsample, downsample or interpolate data.
XTR	Find extrema (maxima or minima).

2.3 DSPlay van Burr-Brown

Overzicht van functies:

TABLE OF FUNCTIONS

subGram

Input
- *ADC 1ch
- get File
- *Pulse
- Sin gen
- Cos gen
- Random
- Gaussian
- DC
- Waveform
- *ADC 2ch

Filter
- FIR
- IIR
- iiR cascade
- Adaptive
- Convolve

Window
- Hamming
- Hanning
- Triangle
- Custom
- Rectangular

Spectrum
- FFT
- Inverse FFT
- DFT
- iNv DFT
- Power spectrum
- Correlation

Control
- Delay
- Limiter
- Threshold switch
- Band switch
- Peak

Arithmetic
- Add
- Subtract
- Multiply
- Divide
- Weight
- abs Value
- aCcumulate
- Quantize

Trigonometric
- Sine
- Cosine
- Tangent
- arcsIne
- arccOsine
- Arctangent

Non-linear
- loGarithm
- Logarithm naturalis
- Exponential
- Square
- sQuare root
- Power
- Root
- Integral
- Derivative

Data
- Rect to Polar Conversion
- Polar to Rect Conversion
- Interpolate
- Decimate
- Append
- Cut
- Zero pad
- Complex to Real Conv.
- Real to complex Conv.
- Section
- Conjugate

***UD block**
- UD1.... UD10

***Output**
- Threshold Dig.Output
- Band Digital Output
- Digital to Analog Convert.
- put File

2.4 DFDP van Atlanta Signal Processors

Overzicht van functies:

FEATURES

asy and Fast to Use—

Menus and a complete manual including a five chapter re- esher on Digital Filtering allow a novice filter designer to oduce running assembly language code in less than a day.

asy and Fast to Implement—

Generates fully commented assembly language code for a riety of DSP chips using ASPI's CGEN modules.

Interfaces directly with the ASPI Algorithm Development ckages.

esign Advantages—

Outputs filter coefficients for implementation as a cascade of cond-order sections. Sections are ordered and paired to nimize arithmetic roundoff effects.

Evaluates the filter responses before and after coefficient antization.

Runs on IBM PC, XT, AT and compatibles and will exploit ating-point coprocessors.

Plots magnitude, log magnitude, IIR and FIR designs, and pulse responses for both.

esigns IIR Filters—

tterworth   Chebyshev I and II

iptic

ots IIR Filters—

Sampling Frequency = 10 Kilohertz

Plots the Phase, Group Delay and Pole-zero response for nter or screen display in addition to the magnitude and im- se response.

esigns FIR Filters—

Kaiser window method or optimal design technique due to ks and McClellan

Hilbert transformers and differentiators

ıltiband

Approximate any piecewise linear magnitude specification TIMALLY

Can produce raised-cosine response for reduced intersym- interference in data communication applications

HERE'S HOW IT WORKS

The DFDP2 design modules present menus and querie Through responses to questions, the designer specifies th type of filter, the sampling frequency, the filter cutoff frequer cies, and the attenuation requirements.

The program estimates the required filter order or impuls response length that should be required, but allows the user t make the final decision. The program then calculates the filte coefficients, quantizes them, and evaluates the frequenc response of the resulting filter. A screen message may warı that the specifications could not be met with a filter of th order desired.

Example:

The IIR (Infinite-length Impulse Response, or recursive design module was used to design a bandpass filter. The sam pling frequency, band edges, and nominal error values in th various bands were specified to the values indicated in th table below. The program then determined that these specifi cations could be met with a 16th order Butterworth, a 10th order Chebyshev, or an 8th order elliptic filter. The elliptic filter was chosen and the following table appeared on the monitor of the PC. In addition to summarizing the design spe- cifications, this indicates the actual peak errors in the pass- band and two stopbands.

CHARACTERISTICS OF DESIGNED FILTER

ELLIPTIC BANDPASS FILTER

Filter Order = 8   Sampling Frequency = 10 Kilohertz

	BAND 1	BAND 2	BAND 3
Lower Band Edge	.00000	2.00000	3.50000
Upper Edge Band	1.50000	3.00000	5.00000
Nominal Gain	.00000	1.00000	.00000
Nominal Ripple	.01000	.01000	.01000
Maximum Ripple	.00518	.00568	.00518
Ripple in dB	−45.71958	.04918	−45.71628

DFDP2 will design filters with either quantized (in a form acceptable for the ASPI Code Generators) or unquantized (floating-point) coefficients. For this example the filter coeffi- cients were quantized to 16 bits for an implementation as a cascade of second-order sections. In the following table, which was generated by DFDP2, the B(I,J) coefficients locate the zeros of the filter and the A(I,J) coefficients locate the poles.

2.5 DADISP van DSP Development Corporation

Van dit programmapakket is bij mij geen goede informatie folder aanwezig. Het pakket bevat onder andere de volgende functies:

1. bewerkingen op signalen (berekingen van het gemiddelde; de som enz.);
2. Fouriertransformaties;
3. auto- en kruiscorrelaties;
4. vensterfuncties;
5. (tri)goniometrische functies (sin, cos, sinh, exp enz.);
6. statistische functies (amplitude distributie; standaarddeviatie enz.);
7. piekanalyse (bepaling van minimum, maximum enz.);
8. opwekken van vele soorten signalen;
9. signaal display en manipulatie;
10. data type conversie (van polaire notatie naar cartesische enz.);
11. bewerking van signalen (decimeren, interpoleren enz.);
12. DSP pipeline functies (schrijven naar en lezen van een file enz.);

2.6 MONARCH van The Athena Group

Overzicht van functies:

SYSTEM SPECIFICATIONS

Monarch represents a new approach in digital signal processing software. Monarch offers not only performance and precision, but also powerful DSP features unique in the high-tech marketplace. And at an affordable price. Monarch means state-of-the-art technology for less.

Created to serve both engineers and scientists, Monarch covers the full spectrum of DSP design requirements -- from entry-level to experienced users. Based on five main modules which share common data structure and user interface, Monarch works as an integrated and comprehensive system. The standard package takes a design specification through synthesis, analysis and testing, and generates detailed engineering specifications for hardware integration. Add-on applications packages provide users with specialized DSP tools, including interface to TMS320 and other board-level products. Monarch is the first comprehensive and affordable DSP software tool.

MAIN MENU OPTIONS

Design

- *FIR*
  - ✓ *Chebyshev*
  - ✓ *Maximally flat*
  - ✓ *Direct synthesis*
  - *Other options*
- *IIR*
  - ✓ *Butterworth*
  - ✓ *Chebyshev*
  - ✓ *Elliptic*
  - ✓ *Pole-Zero model*
  - *Other options*
- *Architecture*
  - ✓ *Parallel*
  - ✓ *Cascade*
  - ✓ *Direct II*
  - *Other options*

View

- *FFT*
  - ✓ *Linear*
  - ✓ *Logarithmic*
- *Graph-2D*
  - ✓ *1 window*
  - ✓ *2 windows*
  - ✓ *3 windows*
  - ✓ *4 windows*
  - ✓ *5 windows*
  - ✓ *6 windows*
  - ✓ *Overlays*
- *Graph-3D*
  - ✓ *Top view*
  - ✓ *Side view*
  - ✓ *Oblique view*
- *Text-Edit*
  - ✓ *Text editor*
- *Other Options*

SigLab

- *Functions*
  - ✓ *Real*
  - ✓ *Complex*
  - ✓ *Binary*
- *Operations*
  - ✓ *Signal library*
  - ✓ *User supplied*
- *Linear Algebra*
  - ✓ *Vectors*
  - ✓ *Matricies*
- *Display*
  - ✓ *Graph*
  - ✓ *Text*
- *Control*
  - ✓ *Looping*
  - ✓ *External calls*
  - ✓ *Conditionals*

Applications

- *TMS320*
- *Filters & systems*
- *Adaptive filters*
- *Spectral analysis*
- *Telecommunication*
- *Biomedical*
- *Sound & vibration*
- *Image processing*
- *Multidimensional*
- *Speech*
- *Systolic arrays*
- *DSP-32*
- *Others*

OS Shell

- ✓ Xecute
- ✓ Directory
- ✓ Copy
- ✓ Rename
- ✓ Print
- ✓ Mkdir
- ✓ Type

✓ denotes standard Monarch operation

Monarch answers filter design and performance questions with precise, rapid turnaround. All filter design options use the state variable approach, creating a consistent data structure for analysis. Therefore, custom filters are designed and analyzed as easily as standard ones.

As new methodologies and technologies are introduced, Monarch upgrades and additional applications packages will be made available. Monarch is the new standard in DSP software.' And we intend to continue to set the standard.

2.7 FDAS van Momentum Data Systems

Version 1.4 of the Filter Design and Analysis System is the result of Momentum Data System's continuing research and development to provide a sophisticated digital filter design system which combines both ease of use and affordability.

Version 2.1 is an enhancement of FDAS1 and incorporates advanced design features such as Phase Equalization, Arbitrary Magnitude Filters, Arbitrary Phase Filter, digitization of s-domain transfer functions as well as significantly increased filter lengths.

Unless specifically indicated by asterisks (**), the following features apply to both FDAS1 and FDAS2.

FEATURES:

- **Outstanding graphics** to aid in the design and analysis of digital filters. Output to CRT's, printers or plotters takes into account the characteristics of the devices, resulting in superior quality graphical output.
- **Menu-driven screens** and extensive error checking allow for ease of use.
- **Pop-up Help Screens** on all data entry fields.
- **Use of 64-bit floating point** for all design calculations. This ensures that maximum accuracy is maintained for the design computations.
- **Coefficient quantization** variable from 8 to 32 bits. This allows comparison of the effects of quantization for word lengths of different Digital Signal Processors.
- **Coefficient Scaling** for Cascade Form I and II, Parallel Form I and Floating Point implementation.
- **Recycling of input** for comparative analysis. Input data for a filter is retained until a new specification is required. This allows various IIR designs to be easily compared with FIR window designs as well as FIR equiripple designs.
- **Specification file** which allows the retention and retrieval of previously designed filters.
- **Graphics interchange file** available. Allows graphical output to be incorporated into various document processing systems such as Lotus Manuscript™, Xerox Ventura Publisher™ V1.1.
- **Wide variety** of CRT's, printers and plotters are supported.
- **High Performance** system to maximize user productivity.

DESIGN CAPABILITIES:

Infinite Impulse Response Design

The following Analog Type Filters are available:

Butterworth
Tschebyscheff
Inverse Tschebyscheff
Elliptic
Bessel

Finite Impulse Response Design with Window Functions

Rectangular
Triangular
Hanning
Hamming
Blackman
Kaiser

Equiripple Finite Impulse Response Design (Parks McClellan)

System Analysis

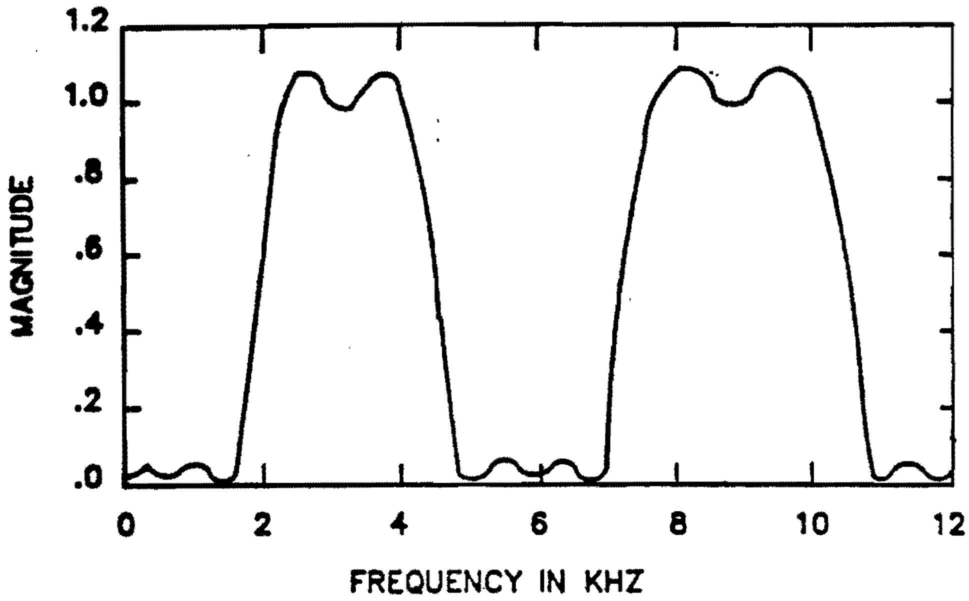
The System Analysis section of the system allows the determination of the characteristics (magnitude, phase, group delay, log magnitude, impulse response and pole/zero locations) of a given transfer function.

Digitization of s-domain transfer functions is offered as a Version 2 feature.

CODE GENERATORS FOR MOST DSP PROCESSORS AVAILABLE

2.8 DFD van Microcraf Corporation

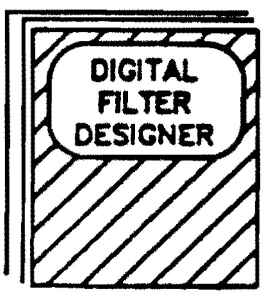
Overzicht van functies:



DIGITAL FILTER DESIGNER (DFD) ALLOWS THE USER TO SPECIFY GENERAL CHARACTERISTICS OF THE DESIRED FILTER. THIS INCLUDES FILTER TYPE, SAMPLING RATE, CUTOFF FREQUENCIES, AND PASS/STOP BAND RIPPLE.

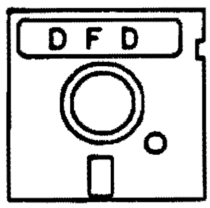
FEATURES THE FOLLOWING CAPABILITIES:

- DESIGNS FIR FILTERS INCLUDING MULTI-BAND PASS/STOP BAND, DIFFERENTIATORS, AND HILBERT TRANSFORMERS.
- DESIGNS IIR FILTERS INCLUDING LOW PASS, HIGH PASS, PASS/ STOP BAND IN THE BUTTERWORTH, CHEBYSHEV I, CHEBYSHEV II, AND ELLIPTIC TYPES.
- COMPATIBLE WITH CGA, EGA, AND HGA GRAPHICS ADAPTERS FOR CRT FREQUENCY RESPONSE PLOTS. GRAPHIC PRINTER PROGRAMS ALSO INCLUDED FOR HARD COPY.



DIGITAL FILTER DESIGNER IS AVAILABLE ON PC-DOS 5 1/4 DISC WITH MANUAL.

MANUAL INCLUDES DESIGN EXAMPLES AND SAMPLE TMS32010 IMPLEMENTATIONS



A POWERFUL PROGRAM AT A REASONABLE PRICE !

MICROCRAFT CORPORATION * PO BOX 513 * THIENSVILLE * WI 53092

2.9 CAT (Computer Aided Teaching) van EPFL in Lausanne

Dit pakket is uitgegeven door Professor M. Kunt van de Ecole Polytechnique Federale in Lausanne. Van dit pakket is bij mij geen inhoudsopgave beschikbaar, maar het bevat onder meer de volgende functies:

1. het opwekken van signalen (sinus, sinuspulsen, enz.);
2. het opwekken van vensterfuncties (rechthoekig, Hamming, enz.);
3. histogram berekening;
4. elementaire filtering;
5. het manipuleren met signalen (gewogen som, product, lengtebeperking, enz.);
6. orthogonale transformaties (Fourier, Hadamard enz.);
7. het berekenen van de impulsresponsie;
8. het berekenen van de convolutie;
9. het berekenen van de autocorrèlatie;
10. het uitvoeren van spectraalanalyse;
11. het ontwerp van digitale filters.

2.10 Digital filter design software for the IBM PC (boek)

Gegevens van dit boek:

Digital filter design software for the IBM PC,
F.J. Taylor en T. Stouraitis,
Marcel Dekker inc,
New York, Basel.

Bij dit boek wordt standaard een floppy disk geleverd met daarop een aantal programmas; o.a. voor:

1. FFT;
2. versterfuncties;
3. filtering;
4. correlatie;
5. interpolatie;
6. bilineaire transformatie;
7. FIR ontwerp;
8. IIR ontwerp.

2.11 Signal processing algorithms (boek)

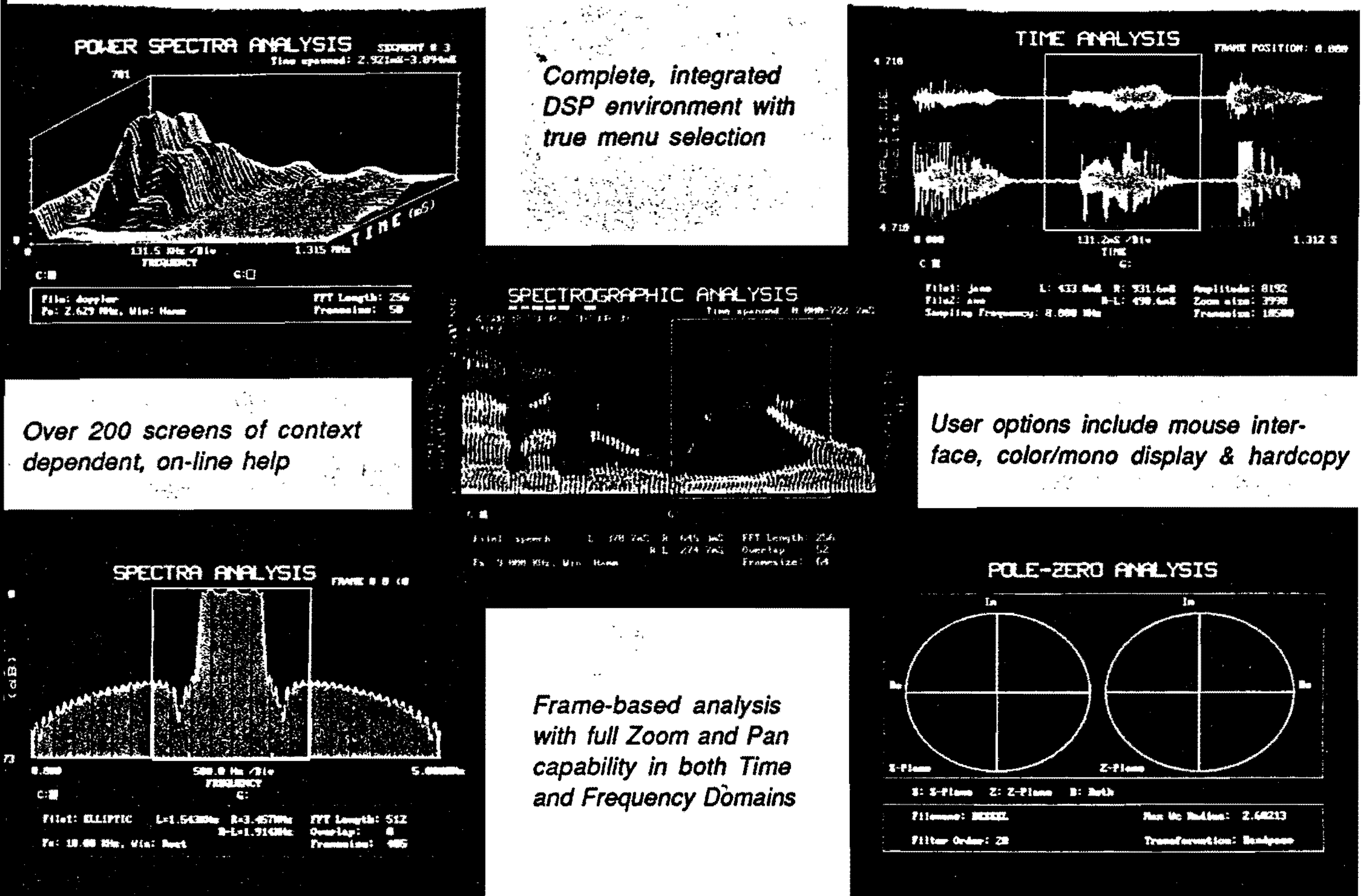
Gegevens van dit boek:

Signal processing algorithms,
S. D. Stearns en R. A. David,
Prentice Hall,
Englewood cliffs,
New Jersey.

Ook bij dit boek wordt standaard een floppy disk geleverd met daarop een aantal programmas. Op de volgende pagina vindt U een uitgebreid overzicht van alle beschikbare programma's.

Routine	Chapter	Purpose or result
SPBFCT	7,11	Ratio of factorials
SPBILN	7	Bilinear analog to digital transformation
SPBSSL	7	Bessel low-pass analog filter coefficients
SPBWCF	7	Butterworth low-pass analog filter coefficients
SPCBII	7	Chebyshev Type 2 low-pass analog filter coefficients
SPCFLT	6	Cascade filtering of real data in place
SPCHBI	7	Chebyshev Type 1 low-pass analog filter coefficients
SPCHRP	14	Chirp z-transform of a complex sequence
SPCOMP	3	Single component of the DFT of a real sequence
SPCONV	9	Fast convolution of two real data sequences
SPCORR	9	Fast correlation of two real data sequences
SPCROS	4	Cross power spectrum of two real sequences
SPDECI	10	Linear decimation of a real data sequence
SPDFTC	3	DFT of a complex sequence of any length
SPDFTR	3	DFT of a real sequence of any length
SPDURB	11	Durbin's solution of special Toeplitz equations
SPEXPN	13	Least-squares coefficients of $a(b**x)$
SPFBLT	7	Conversion from analog to digital filter
SPFFTC	3	FFT of a complex sequence of length = power of 2
SPFFTR	3	FFT of a real sequence of length = power of 2
SPFILT	6	Direct-form filtering of real data in place
SPFIRD	8	FIR filter coefficients via windowed Fourier series
SPFIRL	8	FIR low-pass filter coefficients
SPFLTR	6	Direct-form filtering of real data from X to Y
SPGAIN	5	Complex gain of a linear system at one frequency
SPGAUS	11	Solution of normal equations via Gauss elmination
SPHILB	14	Coefficients of an FIR Hilbert transformer
SPIDTR	3	Inverse DFT; reverse of SPDFTR
SPIFTR	3	Inverse FFT; reverse of SPFFTR
SPLEVS	11	Levinson's solution of Toeplitz normal equations
SPLFIT	13	Least-squares straight-line coefficients
SPLFLT	6	Lattice-form filtering of real data in place
SPLINT	10	Linear interpolation in a real data sequence
SPLMTS	13	Upper and lower limits in a real data sequence
SPLSMT	11	Coefficients required for orthogonal polynomials
SPLTCF	6	Conversion from direct to lattice coefficients
SPMASK	14	Apply a specified data window to a real data vector
SPMEAN	13	Mean value of any real data vector
SPNLMS	12	Normalized LMS algorithm
SPNORM	11	Normal equation coefficients for least-squares polynomial
SPORTH	11	Least-squares coefficients for orthogonal functions
SPPFLT	6	Parallel-form filtering of real data from X to Y
SPPOLY	11	Least-squares coefficients for polynomial functions
SPPOWR	4	Power spectrum of a real sequence
SPPWRC	13	Least-squares coefficients of $a(x**b)$
SPRAND	4	Uniform random number from 0.0 to 1.0
SPRESP	5	Time response of a linear system to any input
SPRFTM	13	Rise and fall time parameters
SPSTRL	11	Stirling numbers of the first kind
SPUNWR	14	Simple phase unwrapping
SPVARI	13	Variance of any real data vector
SPWLSH	14	Single Walsh-ordered Walsh coefficient
SPWNDO	14	Single sample of a data window
SPXEXP	13	Least-squares coefficients of $bx(e**ax)$
SPZINT	10	Frequency-preserving interpolation via zero-padding

2.12 HYPERCEPTION van Hyperception



..and tame the beast in your DSP problems with Hypersignal-Workstation DSP software.

persignal-Workstation has emerged in the last year and a half as the new standard in *professional* P software. Before you pay an arm and a leg for multiple packages that don't play together, con- t of umpteen separate "modules", or worse yet, weren't designed for real DSP problems, consider at customers worldwide already have:

EAL-TIME SUPPORT

ntegrates TMS320C25 , AT&T DSP32, DSP56000, and Data Translation boards
ontinuous acquisition/playback
ample rates up to 150 KHz
ccelerated FFT processing
SP source code interface
ual-channel digital oscilloscope
eal-time spectrum analyzer

ANALYSIS

- o Time display and waveform editing
- o FFT processing, including window, framesize, and overlap control
- o 3-D, 2-D Spectrographic analysis
- o Power spectra estimation
- o Convolution, recursive filtering
- o LPC autocorrelation
- o Pole-zero, unwrapped phase, group-delay, linear/log magnitude, etc.

DESIGN & SIMULATION

- o Classical, arbitrary FIR/IIR design
- o Comprehensive difference equations
- o Math functions and signal arithmetic
- o Arbitrary function generation
- o Code generation and quantization simulation for TMS320xx (including TMS320C30), AT&T DSPxx, DSP56000, ADSP-2100. Selectable IIR structure and quantization.

about our 20 MHz 386/387 DSP workstation

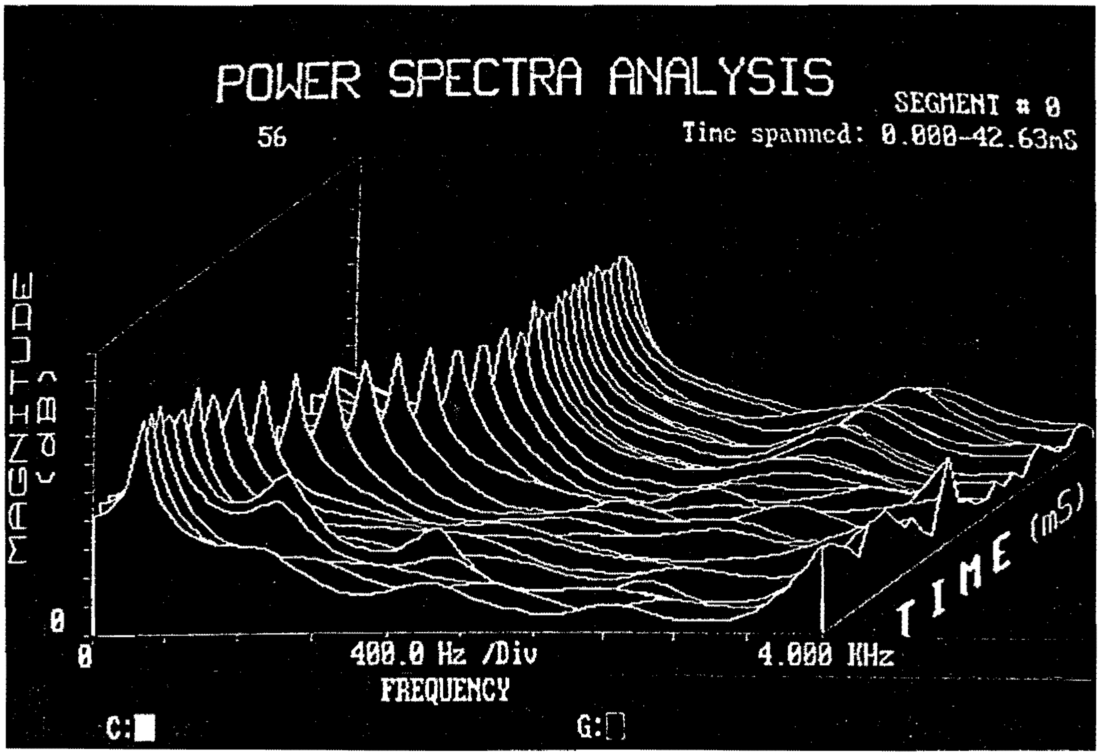
npare before you buy; ask for our free demo disk ***PHONE 214-343-8525 or FAX 214-343-2457***

2.13 HYPERSIGNAL van Loughborourg Sound Images (LSI)

Hypersignal-Workstation

Ein integriertes Softwarepaket für die digitale Signalverarbeitung

In nahezu allen Anwendungen von digitalen Signalprozessoren sind Funktionen wie der Entwurf digitaler Filter (FIR,IIR), Signalanalyse im Zeit- und Frequenzbereich (z.B. FFT) und Echtzeitunterstützung wichtig. Das integrierte Softwarepaket Hypersignal-Workstation unterstützt alle o.g. Funktionen.



Signalanalyse im Zeitbereich
Signale können vergrößert, verkleinert, ausgemessen und editiert werden. Unter den mathematischen Verfahren finden sich Autokorrelation, Faltung und Filterung von Signalen. Etwas ganz Besonderes stellt die Differenzengleichungsfunktion dar. Sie erlaubt die Eingabe von Gleichungen in einer Standard Nomenklatur.

Signalanalyse im Frequenzbereich
Das Programm bietet umfangreiche Möglichkeiten zur Darstellung von Betrag/Phase und Leistungsspektrum. Verschiedene Maßstäbe (logarithmisch, linear, quadriert) sind ebenso möglich wie unterschiedliche zwei- und dreidimensionale Darstellungsformen (z.B Wasserfall oder Spektrogragh).

Echtzeitverarbeitung
Hypersignal-Workstation arbeitet direkt mit verschiedenen Signalprozessor- und Datenerfassungsplatinen zusammen. Dabei dienen die DSP-Karten sowohl als Meßdatenerfassungskarte als auch als Rechenbeschleuniger für die verschiedenen FFT Operationen. Durch diese Kombination sind eine

Fortsetzung Seite 2

2.14 MATLAB van The MATH Works Inc.

Overzicht:

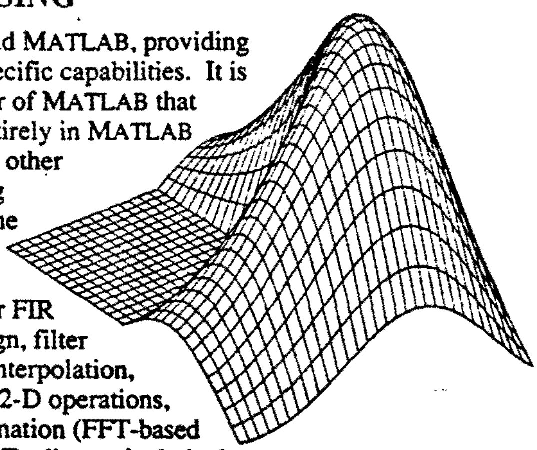
The popular PRO-MATLAB and application specific TOOLBOXES are now available on a wider variety of computers! PRO-MATLAB, called PC-MATLAB on personal computers, is the premier interactive program for numerical linear algebra and matrix computation. With its unique matrix interpreter, complex arithmetic, signal processing algorithms, easy extendibility, and mathematical orientation, MATLAB has rapidly become the software system of choice for high-powered scientific and engineering research.

MATRIX COMPUTATION

MATLAB provides easy access to matrix software from LINPACK and EISPACK including linear algebra functions like eigenvalues, linear-equation solution, least-squares, inverse, pseudoinverse, matrix exponential, singular value decomposition, and almost anything else you can think of to do with matrices. MATLAB is also chock full of other analytical capabilities including complex and polynomial arithmetic, curve fitting, cubic splines, nonlinear optimization, quadrature, ordinary differential equations, and multivariate statistics. Altogether, there are over 200 functions available.

SIGNAL PROCESSING

Optional *Toolboxes* extend MATLAB, providing additional application-specific capabilities. It is a testimonial to the power of MATLAB that Toolboxes are written entirely in MATLAB itself - with no Fortran or other "low-level" programming required. For example, the SIGNAL PROCESSING TOOLBOX is a collection of MATLAB functions for FIR and IIR digital filter design, filter simulation, decimation, interpolation, convolution, correlation, 2-D operations, and power spectrum estimation (FFT-based spectral analysis). Other Toolboxes include the SYSTEM IDENTIFICATION TOOLBOX, for parametric modeling, and the CONTROL TOOLBOX, for control system engineering and state-space modeling.



2-D AND 3-D COLOR GRAPHICS

Graphics tools let you make publication quality 2-D, 3-D, linear, log, semi-log, polar, and contour plots on your plotters, dot-matrix-, and laser-printers.

FAST, ACCURATE AND RELIABLE

MATLAB not only solves mathematical and engineering problems - it does it *fast*. The carefully optimized code fully utilizes any available floating point hardware for maximum performance. For example, on a PC it takes less than 1 second to multiply 20 x 20 matrices and 2.3 seconds to invert them. A 1024 point FFT finishes in 2.4 seconds! On larger machines, the efficient C and assembly language code is even more remarkable. You won't have to question the results either - the numerical algorithms have been programmed by leading experts in mathematical software.

OPEN SYSTEM

Many of MATLAB's features are implemented in programmable *M-files*, made possible because of MATLAB's open-system philosophy. Since MATLAB is the teaching and research system chosen by Engineering, Computer Science and Mathematics departments at most leading universities, you can look forward to an exciting future of new algorithmic developments from leading experts in mathematical and signal processing software.

Please provide additional information:

Name ____________

Company ____________

Dept. ____________

Addr. ____________

City ____________

State,Zip ____________

Tel. ____________

Computer ____________

The MATH WORKS Inc.

Suite 250
20 North Main St.
Sherborn, MA 01770
(617) 653-1415
Telex 910-240-5521

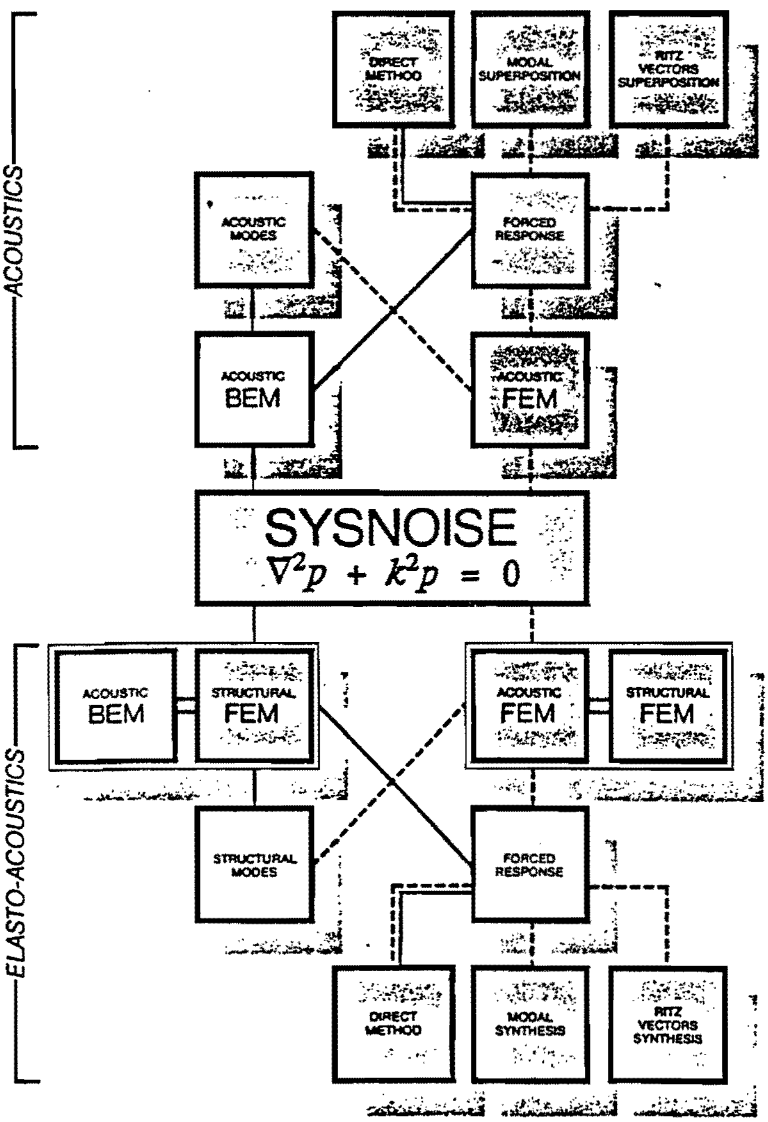
SP-13-88

2.15 SYSNOISE van Dynamic Engineering

Acoustic wave equation

$$\nabla^2\psi = \frac{1}{c^2}\,\frac{\partial^2\psi}{\partial t^2}$$

Analysis features



SYSNOISE is a complete, flexible and easy-to-use numerical noise analysis program. It accurately models both complex acoustic and elasto-acoustic behaviour. The program benefits from a dual boundary element / finite element formulation which makes it possible to describe both interior and exterior noise radiation phenomena. With SYSNOISE it is possible to calculate both acoustic and structural modes as well as the direct response of a system due to applied structural and acoustic forces.

Accurate modelling of acoustic damping behaviour can also be determined using volumetric damping elements or surface impedance treatments. One-, two- and three-dimensional problems may be analysed. It is possible to consider symmetries in the system as well as half-space boundary conditions, baffled radiation and the input from acoustic sources. Experimental data may also be used as input to the problem to specify boundary conditions.

SYSNOISE has powerful 3-D graphic interactive mesh generators and various result plotting facilities (developed in collaboration with FRAMASOFT). It is also coupled to most CAD-systems and is available on a wide range of computers. The SYSNOISE program is developed and supported by DYNAMIC ENGINEERING, an international company specialising in consultancy and sales of noise and vibration software and hardware.

2.16 SPW van Comdisco Systems

Signal Processing WorkSystem™ (SPW™)

Key Features

The Signal Processing WorkSystem (SPW) combines in one integrated package all the tools needed to graphically and interactively design, simulate, and test a broad range of digital signal processing (DSP) systems. Applications include communications, radar/sonar, speech processing, image processing, instrumentation, and feedback control.

Dramatic productivity improvements in algorithm development and simulation are achieved using SPW because it:

- Provides a common, graphical user interface for all aspects of DSP system design and simulation.
- Includes a rich library of DSP models that are ready to insert in a system simulation block diagram.
- Accepts real-world data and sets parameters on test instruments via GPIB port.
- Uses design hierarchy to build upon existing designs and combine existing blocks.
- Supports unlimited feedback path nesting (Patent Pending) for multiple levels of hierarchy and sophisticated designs.
- Allows import of new or existing simulation model code to customize the model library.
- Offers a broad range of signal display and editing tools, as well as on-screen analysis functions.
- Generates high-quality documentation automatically, so users can concentrate on system design.

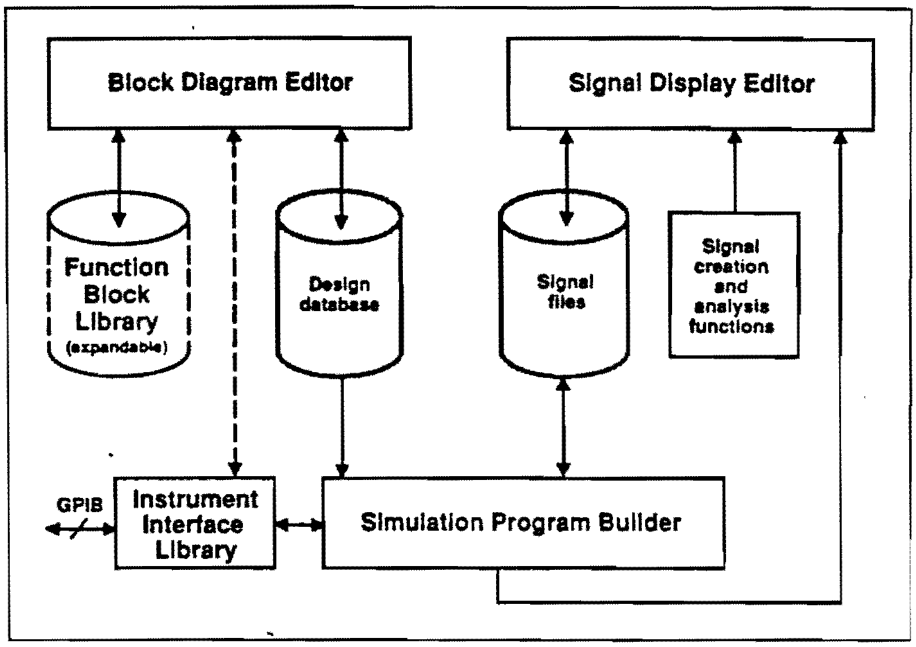


Figure 1 SPW Architecture

The SPW is available on three industry standard platforms: Sun-3™ series/UNIX®, Apollo® 3000 and 4000 series/UNIX and DEC® VAXstation™/VMS™.

Operation

Figure 1 shows the key components of the SPW architecture. The designer uses the Block Diagram Editor to graphically capture a DSP system design as a signal flow block diagram, using blocks from the Function Block Library and Instrument Interface Library or creating new blocks, if desired. Then the designer synthesizes, edits, analyzes, and prepares input signals to test the DSP design, using the graphical Signal Display Editor. Next, the Simulation Program Builder automatically compiles the block diagram and, with the input signals, executes a simulation program. Finally, the designer uses the Signal Display Editor to examine and analyze the results of the simulation and verify or revise the design, all in a fraction of the time required with traditional methods.

Table 1 summarizes some of the capabilities and benefits of SPW.

Block Diagram Editor

- Powerful color graphic signal flow Block Diagram Editor with distributed database
- Multilevel (hierarchical) blocks
- Design revision management functions
- Multiple windows
- Pop-up and customized icon menus

2.17 NAG van de Numerical Algorithms group

De NAG library is een erg algemeen en groot pakket van subroutines; zo zijn er subroutines die de volgende functies uitvoeren:

- Complex Arithmetic
- Zeros of Polynomials
- Roots of One or More Transcendental Equations
- Summation of Series
- Quadrature
- Ordinary Differential Equations
  D02N — Integrators for stiff ordinary differential equations
- Partial Differential Equations
- Numerical Differentiation
- Integral Equations
- Interpolation
- Curve and Surface Fitting
- Minimizing or Maximizing a Function
- Matrix Operations, Including Inversion
- Eigenvalues and Eigenvectors
- Determinants
- Simultaneous Linear Equations
- Orthogonalisation
- Linear Algebra Support Routines
- Simple Calculations on Statistical Data
- Correlation and Regression Analysis
- Analysis of Variance
- Random Number Generators
- Nonparametric Statistics
- Contingency Table Analysis
- Time Series Analysis
- Operations Research
- Sorting
- Error Trapping
- Approximations of Special Functions
- Mathematical Constants
- Machine Constants
- Innerproducts
- Input/Output Utilities

2.18 DSP Processing Experiments (boek)

Gegevens van dit boek:

Digital Signal Processing Experiments,
A. Kamas en E. A. Lee,
Prentice Hall,
Englewood cliffs,
New Jersey.

Ook bij dit boek wordt standaard een floppy disk geleverd met daarop 11 z.g. laboratorium experimenten; ieder experiment bestaat uit een aantal problemen:

- Lab 1. signal generation and convolution,
- Lab 2. the z-transform and system functions,
- Lab 3. pole/zero plots and frequency response,
- Lab 4. linear phase filtering,
- Lab 5. amplitude modulation,
- Lab 6. spectral shifter,
- Lab 7. the DFT, FFT and circular convolution,
- Lab 8. IIR filter design,
- Lab 9. FIR filter design via windowing,
- Lab 10. frequency sampling and equiripple FIR,
- Lab 11. quantization effects,

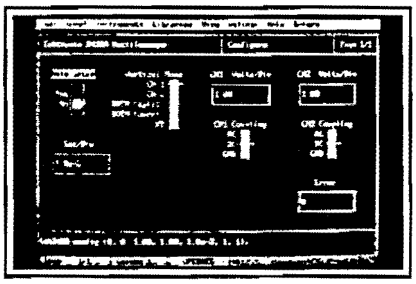
2.19 LABWINDOWS/LABWORKS van National Instruments

We've Invented the Future of Instrumentation Software . . . Twic

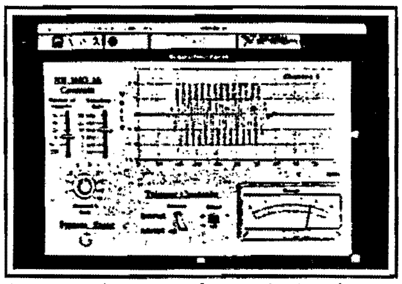
With Words   With Pictures

Acquisition

Integrated libraries for GPIB, RS-232, A/D-D/A-DIO plug-in cards, and modular instruments.



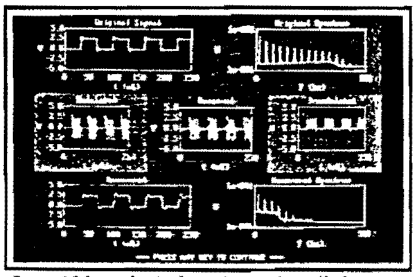
Intuitive character-based function panels that automatically generate source code.



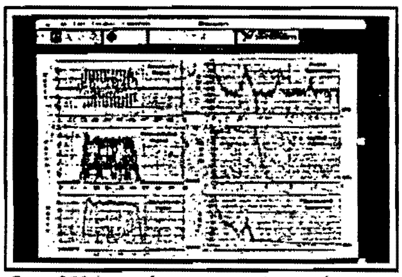
Front panel user interface with virtual instrument block diagram programming.

Analysis

Extensive libraries for data reduction, digital signal processing, and statistical analysis.



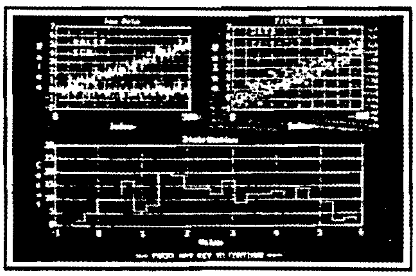
Over 100 analysis functions plus all the built-in functions of your language.



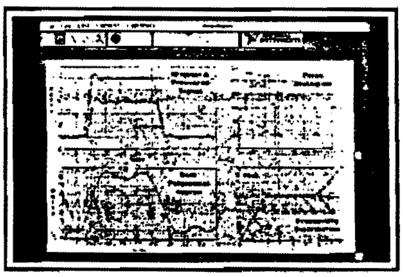
Over 250 icons for computation and analysis.

Presentation

Flexible high-performance graphics and report generation.

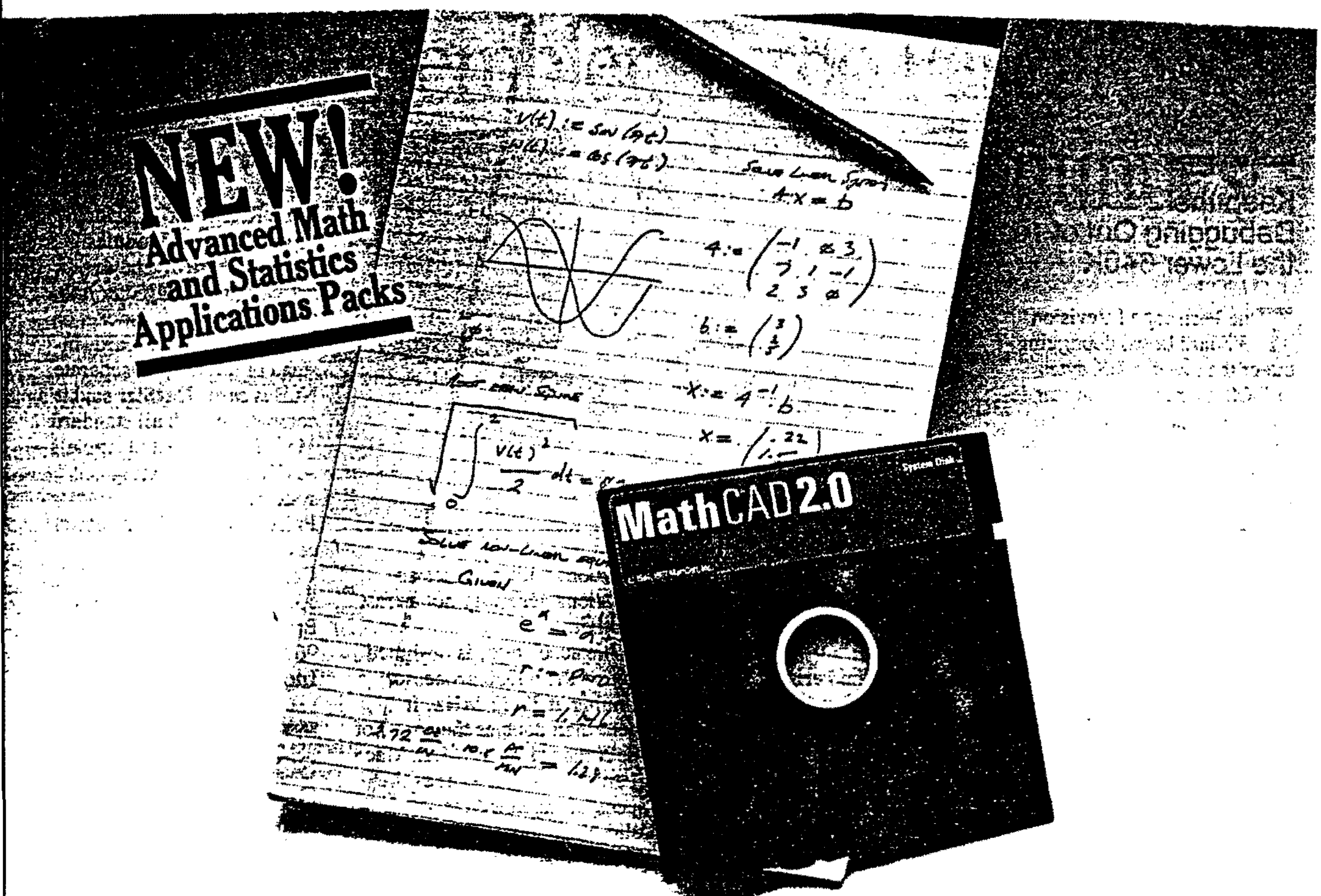


Extensive graphics support for CGA, EGA, MCGA, VGA, and Hercules.

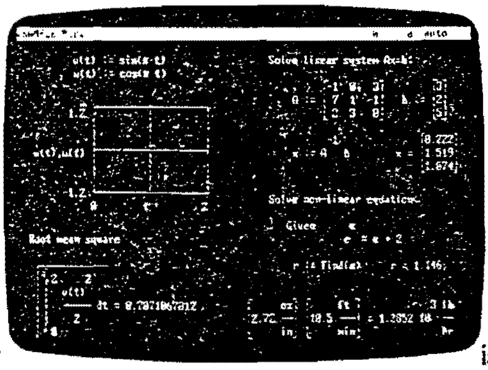


Macintosh Desktop Publishing compatibility.

2.20 MATH CAD van MathSoft Inc.



Your pad or ours?



If you perform calculations, the answer is obvious.

MathCAD 2.0.

It's everything you appreciate about working on a scratchpad – simple, free-form math – and more. More speed. More accuracy. More flexibility.

Just define your variables and enter your formulas anywhere on the screen. MathCAD formats your equations as they're typed. Instantly calculates the results. And displays them exactly as you're used to seeing them – in real math notation, as numbers, tables or graphs.

MathCAD is more than an equation solver. Like a scratchpad, it allows you to add text anywhere to support your work, and see and record every step. You can try an unlimited number of what-ifs. And print your entire calculation as an integrated document that anyone can understand.

Plus, MathCAD is loaded with powerful built-in features. In addition to the usual trigonometric and exponential functions, it includes built-in statistical functions, cubic splines, Fourier transforms, and more. It also handles complex numbers and unit conversions in a completely transparent way.

Yet, MathCAD is so easy to learn, you'll be using its full power an hour after you begin.

What more could you ask for? How about two new applications packs to increase your productivity?

The **Advanced Math Applications Pack** includes 16 applications like eigenvalues and eigenvectors of a symmetric matrix, solutions of differential equations, and polynomial least-squares fit.

The **Statistics Applications Pack** lets you perform 20 standard statistical routines such as multiple linear regression, combinations and permutations, finding the median, simulating a queue, frequency distributions, and much more.

MathCAD lets you perform calculations in a way that's faster, more natural, and less error-prone than the way you're doing them now – whether you use a calculator, a spreadsheet, or programs you write yourself. So come on over to MathCAD and join 45,000 enthusiastic users.

For more information, contact your dealer or call **1-800-MATHCAD** (In MA: 617-577-1017).

2.21 MICRO-CAP III van Spectrum Software

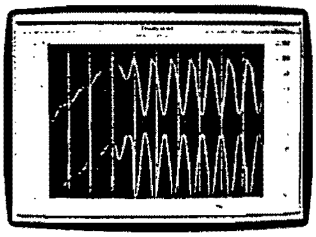
Overzicht:

MICRO-CAP III.™ THIRD-GENERATION INTERACTIVE CIRCUIT ANALYSIS. MORE POWER. MORE SPEED. LESS WORK.

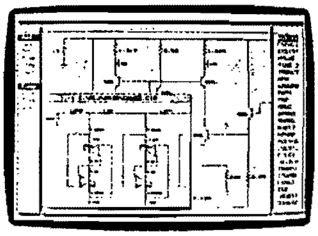
MICRO-CAP III,™ the third generation of the top selling IBM® PC-based interactive CAE tool, adds even more accuracy, speed, and simplicity to circuit design and simulation.

The program's window-based operation and schematic editor make circuit creation a breeze. And super-fast SPICE-like routines mean quick AC, DC, Fourier and transient analysis — right from schematics. You can combine simulations of digital and analog circuits via integrated switch models and macros. And, using stepped component values, rapidly generate multiple plots to fine-tune your circuits.

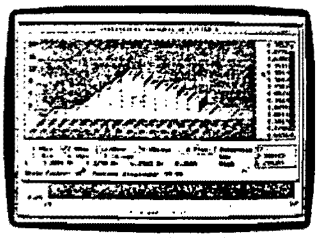
We've added routines for noise, impedance and conductance — even Monte Carlo routines for statistical analysis of production yield. Plus algebraic formula parsers for plotting almost any desired function.



Transient analysis



Schematic editor



Monte Carlo analysis

Modeling power leaps upward as well, to Gummel-Poon BJT and Level 3 MOS — supported, of course, by a built-in Parameter Estimation Program and extended standard parts library.

There's support for Hercules,® CGA, MCGA, EGA and VGA displays. Output for laser plotters and printers. And a lot more.

The cost? Just $1495. Evaluation versions are only $150.

Naturally, you'll want to call or write for a free brochure and demo disk.

spectrum

1021 S. Wolfe Road,
Sunnyvale, CA 94086
(408) 738-4387

MICRO-CAP III is a registered trademark of Spectrum Software.
Hercules is a registered trademark of Hercules Computer Technology.
IBM is a registered trademark of International Business Machines, Inc.

3 Demonstraties

In deze paragraaf zullen een aantal resultaten van computersimulaties beschreven worden. Deze resultaten zijn berekend met een intern Philips programmapakket genaamd DIPRO. Dezelfde resultaten kunnen echter ook zonder veel moeite met de andere programmapakketten, die in dit dictaat beschreven zijn, verkregen worden.

We zullen resultaten geven voor:

1. het ontwerp van een laagdoorlatend filter;
2. het ontwerp van een differentiërend filter;
3. de invloed van eindige woordlengte op de coëfficiënten van een filter;
4. het gedrag van een IIR filter;
5. "leakage" en "windowing" bij de FFT.

3.1 Ontwerp van een laagdoorlatend filter

In deze paragraaf zullen we een laagdoorlatend filter ontwerpen en enkele eigenschappen er van laten zien. We zullen voor dit voorbeeld bij drie programma's precies de vragen laten zien die DIPRO stelt en (waar nodig) zullen we deze van commentaar voorzien.

We willen een laagdoorlatend tijddiscreet filter met een constante rimpel ontwerpen waarbij we het programma van Parks en McClellan gebruiken; zie paragraaf 8.2.1 van het cursusboek "Digitale signaalbewerking". Het ontwerp moet aan de volgende eisen voldoen:

1. de bemonsteringsfrequentie is 8 kHz;
2. de filterlengte is 25;
3. de impulsresponsie $h[n]$ is symmetrisch;
4. de doorlaatband gaat van 0 Hz tot 1840 Hz;
5. de sperband gaat van 2160 Hz tot 1840 Hz;
6. de versterking in de doorlaatband is 100 (40 dB);
7. de rimpel in de doorlaatband = de rimpel in de sperband.

We gaan nu achtereenvolgens de volgende zaken berekenen:

1. het ontwerpen van het filter met het programma DESFIR;

2. het plotten van de impulsresponsie met het programma GRAPH;
3. het plotten van de frequentieresponsie met het programma FIRPLOT;
4. het berekenen van de invloed van een full-T D/A conversie op de frequentieresponsie met programma FIRPLOT;
5. het corrigeren van de invloed van een full-T D/A conversie.

Om het filterontwerp te starten moeten we de naam van het gewenste DIPRO programma intypen: DESFIR

Het programma stelt dan de volgende vragen (het door de gebruiker in te typen antwoord staat achter de dubbele punt.

```
Enter the sampling frequency (Hz)                  : 8000
Enter the filter length (max=3072)                 : 25
Enter the number of bands
        (min=1, max=10)                            : 2
Enter the grid density                             : 16
Enter type of filter:
        1 = multiple pass/stopband
        2 = differentiator                         : 1
Enter type of filter:
        0 = symmetrical h[n]
        1 = antisymmetrical h[n]                   : 0
Enter the lower edge (Hz) of band 1                : 0
Enter the upper edge (Hz) of band 1                : 1840
Is band 1 a passband?                              : y
Enter the desired gain (dB) of band 1              : 40
Enter the ripple (dB) of band 1                    : .0087
Enter the lower edge (Hz) of band 2                : 2160
Enter the upper edge (Hz) of band 2                : 4000
Is band 2 a passband?                              : n
Enter the max. stopband gain (dB) of band 2        : -20
*** INPUT SUMMARY ****
Is this input correct? (y/n)                       : y
*** CALCULATIONS ****
*** RESULTS ****
Should the coefficients be stored?                 : y
Output file name                                   : FIL.COF
Do you want another filter? (y/n)                  : n
*** STOP PROGRAM ****
```

Opmerkingen:

1. naarmate de "grid density" (zie de vierde vraag) toeneemt neemt de CPU-tijd toe maar wordt de constante rimpel in de frequentieresponsie beter benaderd; zie bijvoorbeeld de IEEE publicatie: "Programs for digital signal processing" die eerder genoemd is;
2. de rimpel in de doorlaatband en de maximale versterking in de sperband moet worden opgegeven in dB;
3. de versterking in de doorlaatband mag $100 \pm 0,1$ zijn (100,1 komt overeen met 40,0087dB; de bijbehorende rimpel is dus 0,0087 dB);
4. de maximale versterking in de sperband mag 0,1 zijn (−20 dB);
5. bij **** INPUT SUMMARY **** geeft het programma een uitgebreid overzicht van alle ingetikte data;
6. bij **** CALCULATIONS **** berekent het programma de waarden van de monsters van de impulsresponsie;
7. bij **** RESULTS **** geeft het programma de bereikte rimpel in de doorlaatband en de maximale versterking in de sperband;
8. bij **** STOP PROGRAM **** wordt het programma beëindigd;
9. de naam van de file waarin de coëfficiënten van de impulsresponsie worden opgeslagen is FIL.COF.

Om op het scherm van de terminal een plot van $h[n]$ te krijgen wordt het DIPRO programma GRAPH gebruikt (type dus in: GRAPH). We moeten dan de volgende vragen beantwoorden:

```
Options? (y/n)                              : n
Input file name                             : FIL.COF
*** FILTER INFORMATION ****
Another graph using the same axes?          : n
Do you need all the input samples?          : y
The minimum value for the X-axis            : 1
The maximum value for the X-axis            : 25
The minimum value for the Y-axis            : −20
The maximum value for the Y-axis            : 52
*** PLOT OF IMPULSE RESPONSE ****
```

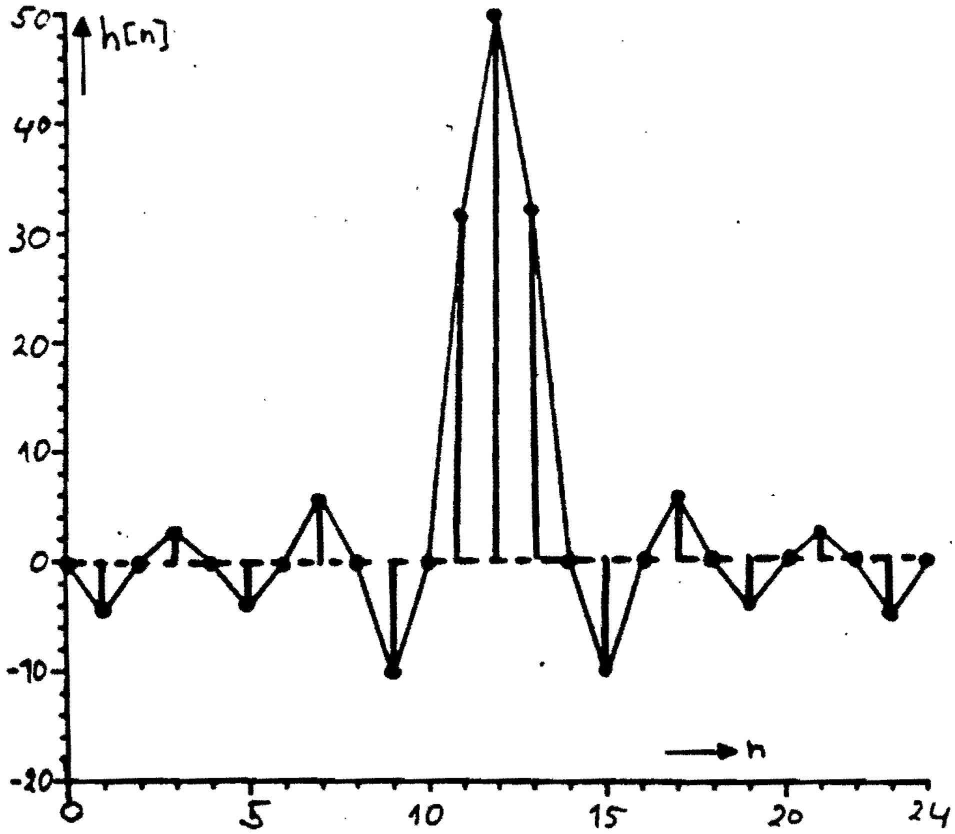
Opmerkingen:

1. bij options kan extra plotinformatie meegegeven worden (bijvoorbeeld over lineaire/logarithmische assen; schalen; teksten langs de assen; enz.); alleen indien deze informatie gewijzigd moet worden moet op de vraag "Options?" geantwoord worden met y (yes).

2. bij **** FILTER INFORMATION **** geeft het programma informatie over het filter (filterlengte, bemonsteringsfrequentie, enz.);

3. bij **** PLOT OF IMPULSE RESPONSE **** begint het daadwerkelijke plotten van de grafiek op het scherm; nu wordt figuur 3.1 op het scherm getekend;

4. in deze laatste figuur zijn niet alleen de monsters van $h[n]$ getekend, maar is ook de omhullende van de impulsresponsie gegeven. Als de tijdfunctie discreet is (zoals bij een impulsresponsie) of als frequentiefunctie discreet is (zoals bij een frequentiespectrum dat met een DFT of een FFT berekend is) zal in het vervolg meestal alleen de omhullende van deze functie gegeven worden.



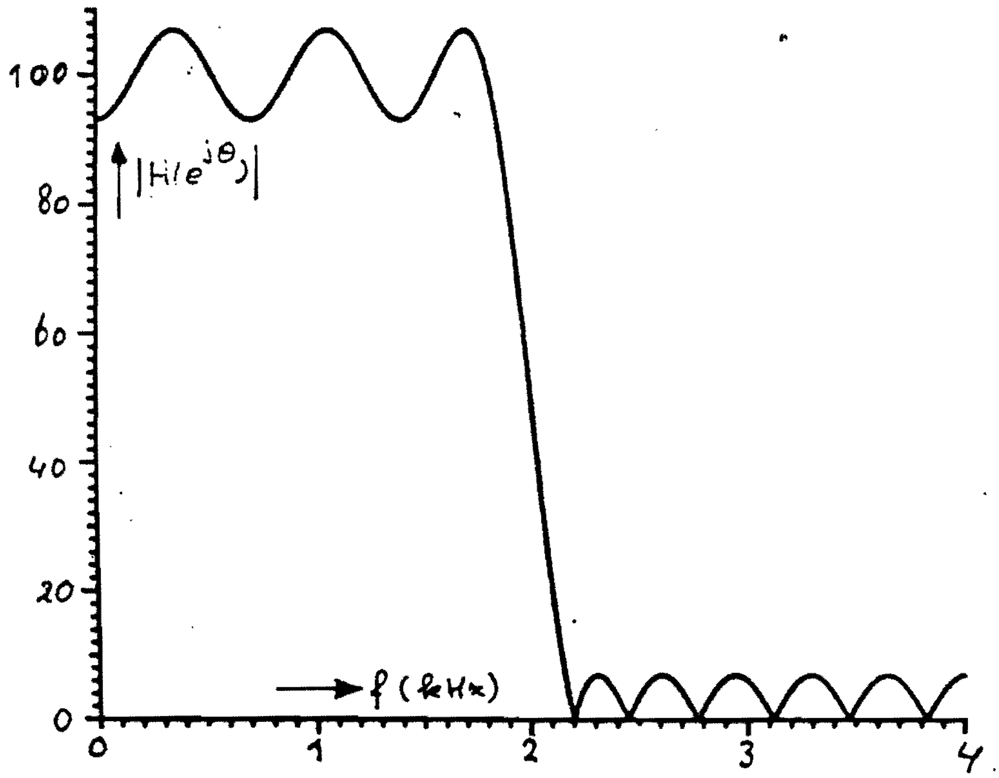
Figuur 3.1 De impulsresponsie $h[n]$.

Om de frequentieresponsie te plotten moet het programma FIRPLOT gebruikt worden (type in: FIRPLOT) en beantwoord de volgende vragen (zie ook de opmerkingen die gemaakt zijn bij het programma GRAPH):

Options?	: n
Input file name	: FIL.COF
*** FILTER INFORMATION ****	
Another filter using the same axes?	: n
The lowest frequency to be plotted	: 0
The highest frequency to be plotted	: 4000
The minimum value for the Y-axis	: 0
The maximum value for the Y-axis	: 120
*** PLOT OF THE RESPONSE ****	

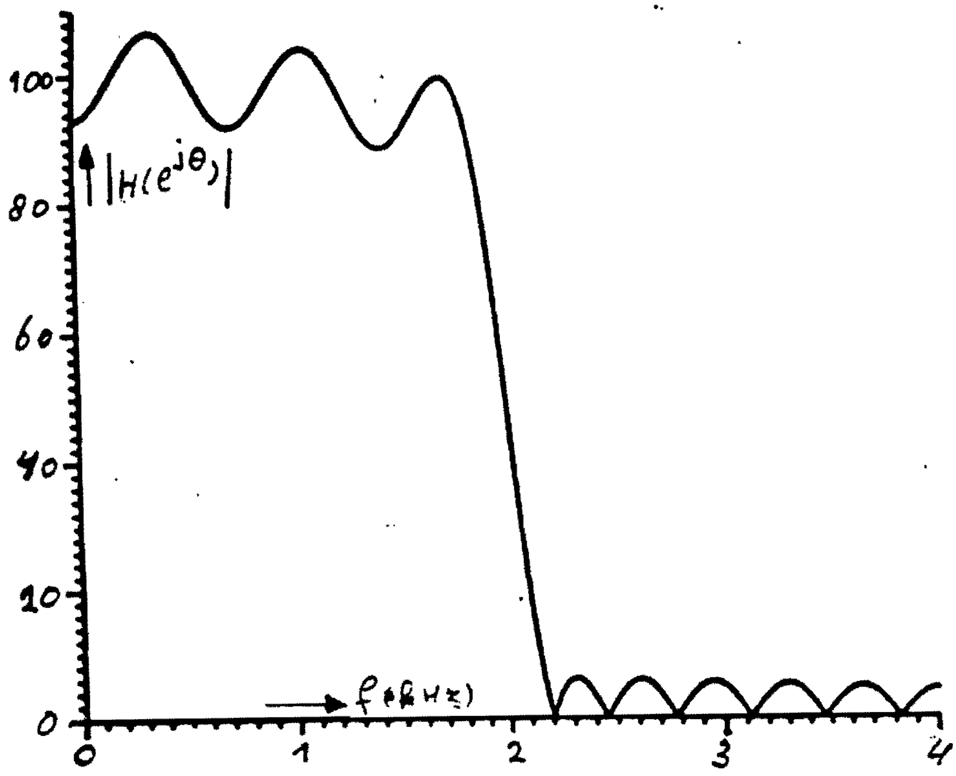
Het programma plot nu de grafiek van figuur 3.2.

In het options blok komt de vraag voor of de invloed van een full-T D/A



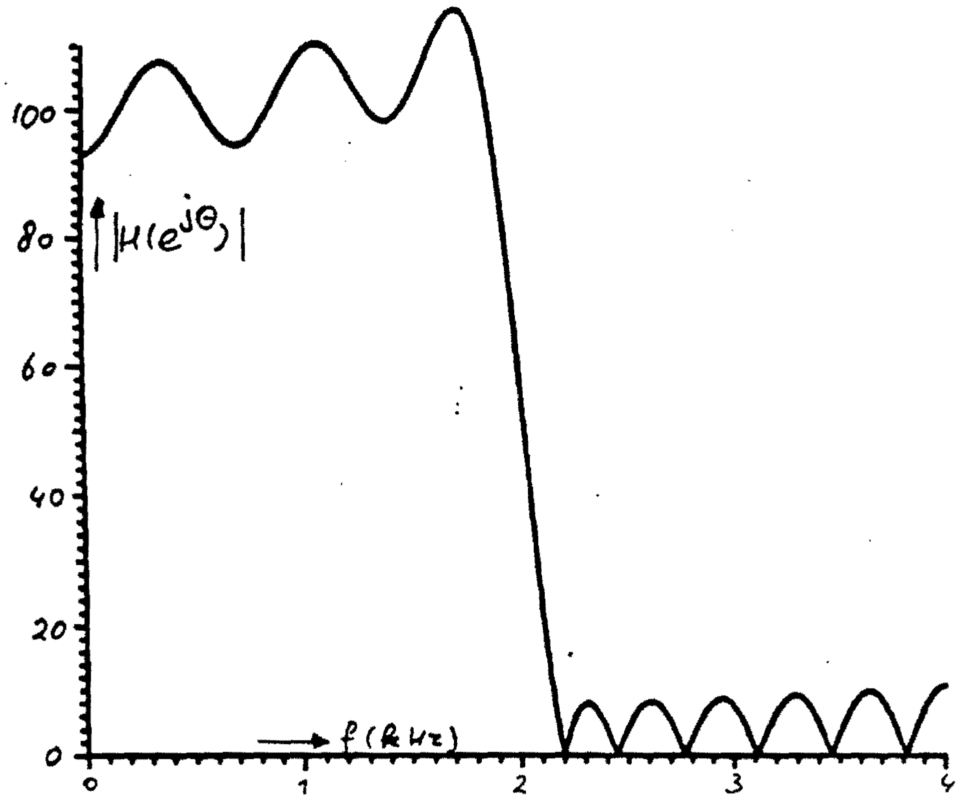
Figuur 3.2 De frequentieresponsie.

conversie meegenomen moet worden. Als dat antwoord bevestigend beantwoord wordt, dan berekent FIRPLOT de frequentieresponsie van figuur 3.3.
Het programma DESFIR kan zodanig gebruikt worden dat de versterking

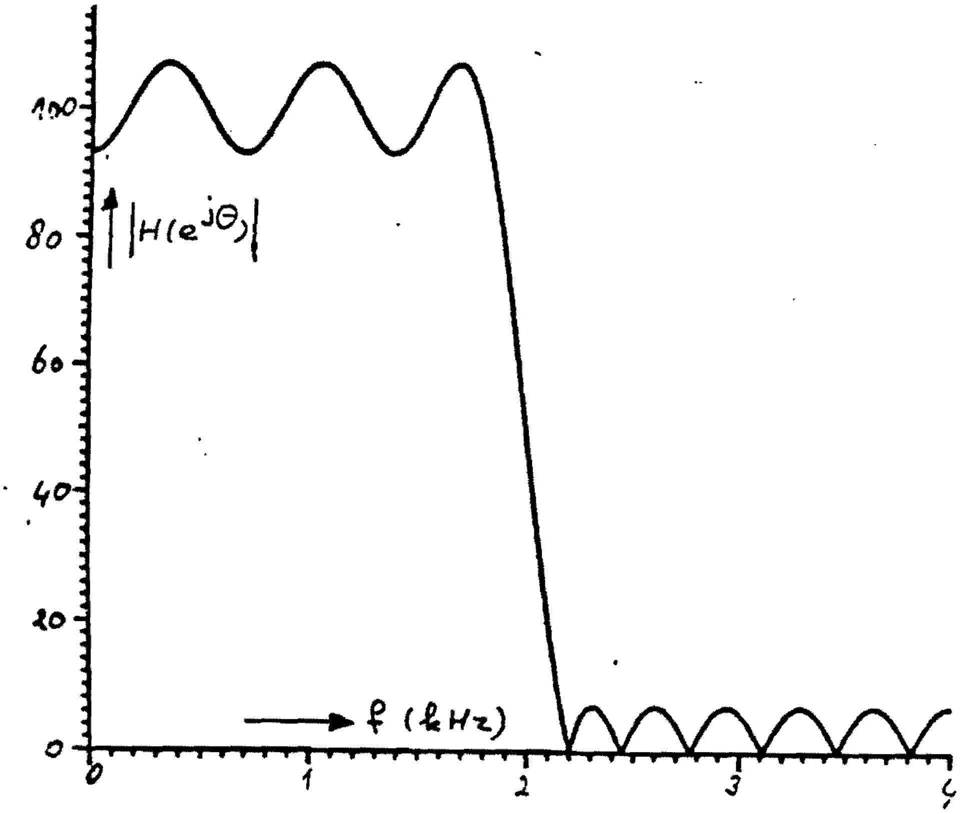


Figuur 3.3 Invloed van een full-T D/A conversie.

en de rimpel frequentieafhankelijk worden. We kunnen deze afhankelijkheid zo kiezen dat de frequentieresponsie van het filter tesamen met de invloed van de D/A convertor een constante rimpel heeft. De frequentieresponsie die DESFIR dan geeft is gegeven in figuur 3.4. Tesamen met de frequentieresponsie van de D/A convertor geeft dit de karakteristiek van figuur 3.5.



Figuur 3.4 De frequentieresponsie met voorcorrectie.



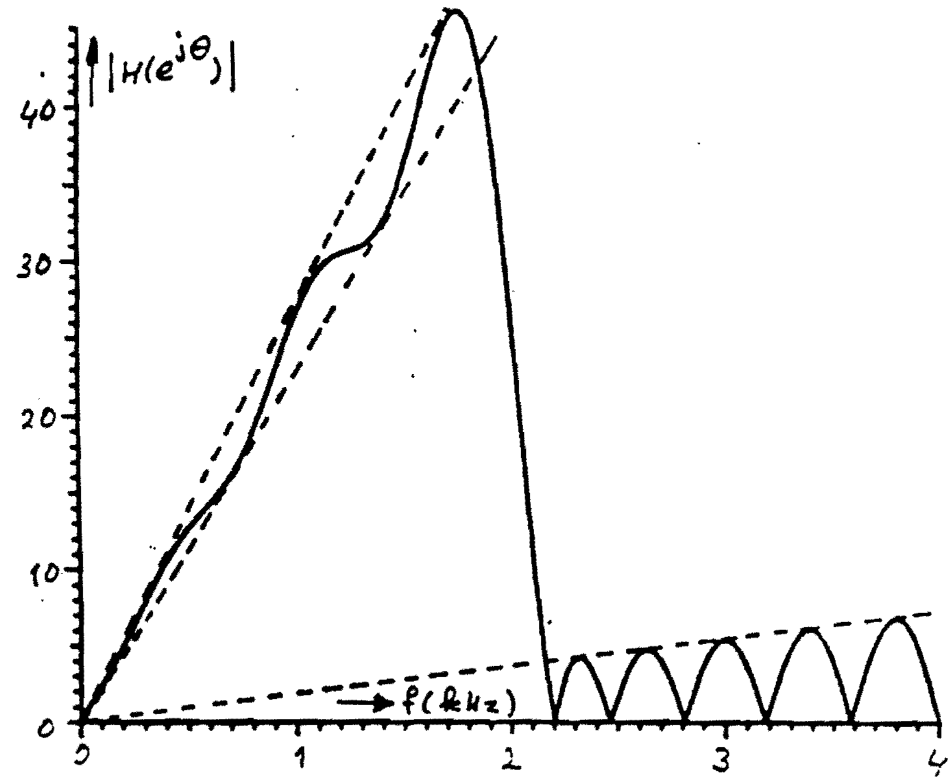
Figuur 3.5 De totale frequentieresponsie.

3.2 Ontwerp van een differentiërend filter

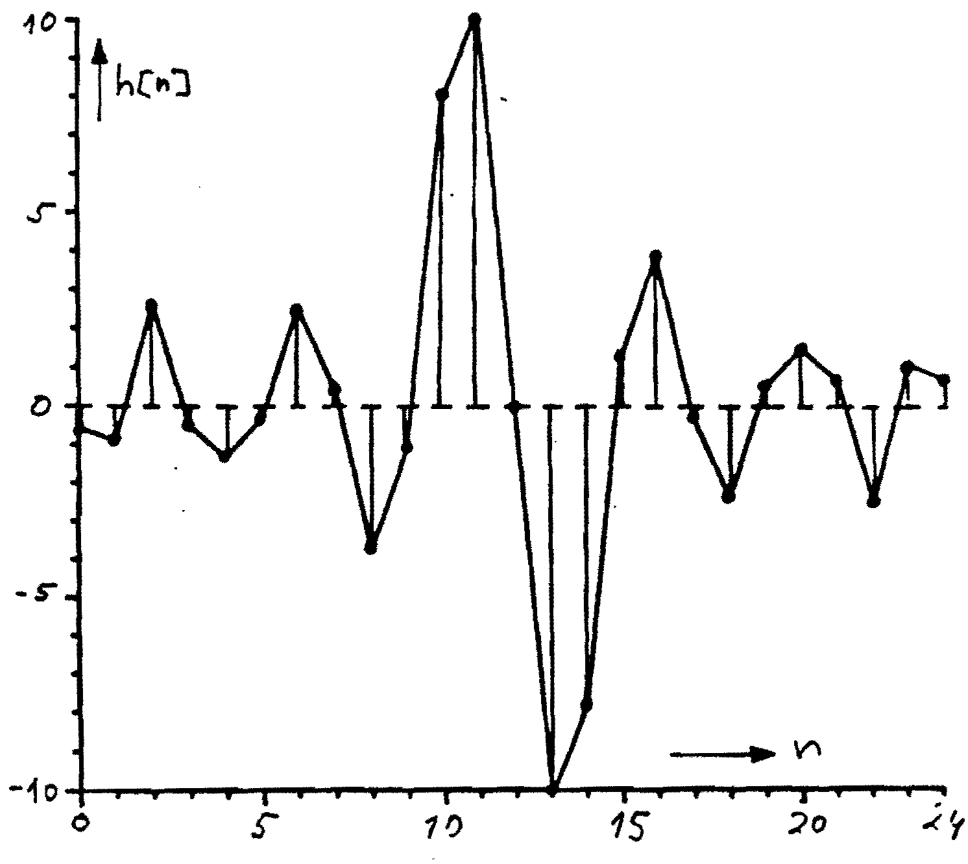
Als tweede voorbeeld gaan we een differentiërend filter ontwerpen dat aan de volgende eisen voldoet:

1. de bemonsteringsfrequentie is 8 kHz;
2. de filterlengte is 25;
3. de impulsresponsie $h[n]$ is antisymmetrisch;
4. de doorlaatband gaat van 0 Hz tot 1840 Hz;
5. de sperband gaat van 2160 Hz tot 1840 Hz;
6. de versterking in de doorlaatband moet lineair toenemen als functie van de frequentie θ, zodanig dat bij de halve bemonsteringfrequentie de waarde 100 (= 40 dB) bereikt zou zijn;
7. de relatieve rimpel in de doorlaatband = de relatieve rimpel in de sperband;
8. de relatieve rimpel = de absolute rimpel gedeeld door de frequentie (immers de versterking van een differentiator moet lineair met de frequentie toenemen).

Figuur 3.6 geeft de frequentieresponsie die verkregen wordt en figuur 3.7 geeft de bijbehorende impulsresponsie $h[n]$.



Figuur 3.6 De frequentieresponsie.



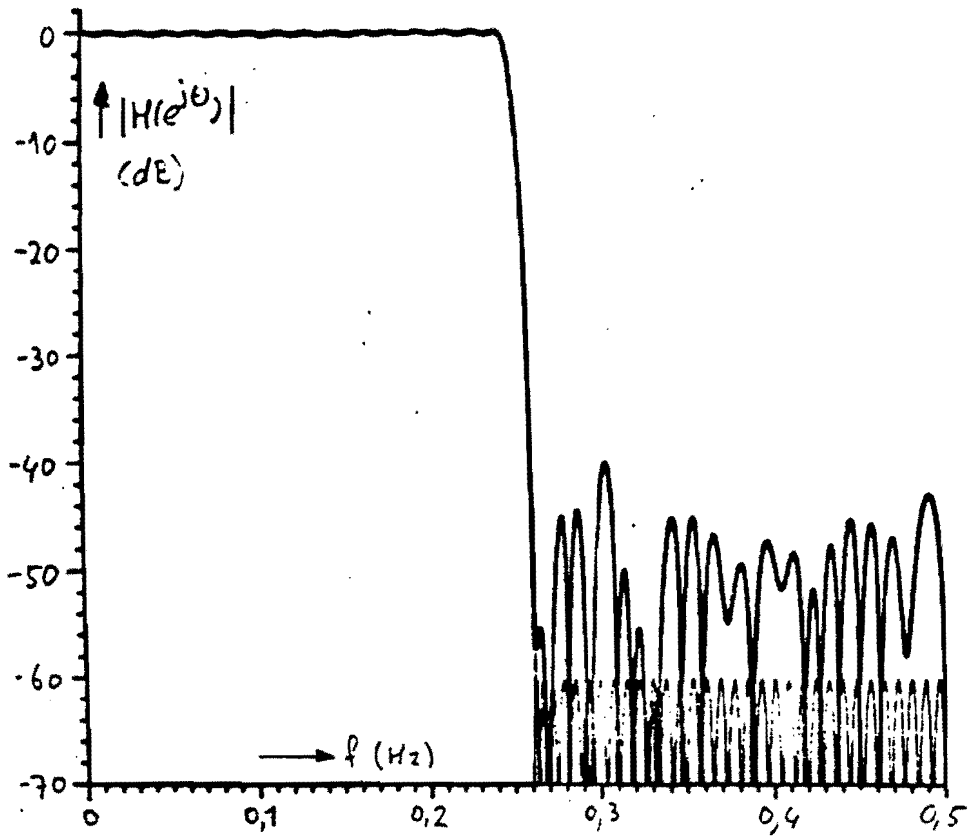
Figuur 3.7 De impulsresponsie h[n].

3.3 Invloed van eindige woordlengte

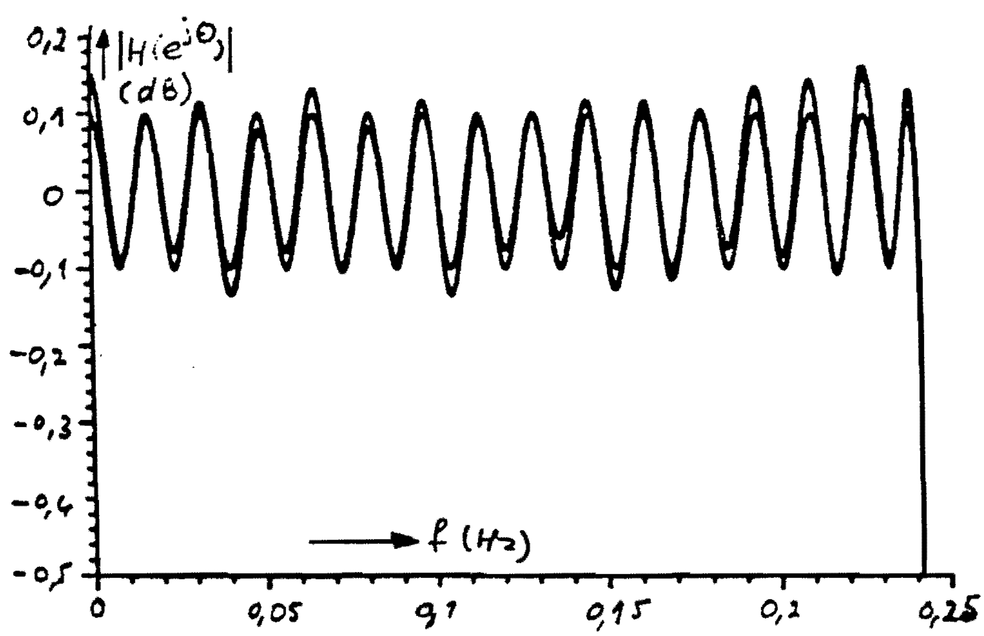
Bij dit derde voorbeeld laten we zien hoe de frequentieresponsie van een transversaal filter verandert als we coëfficiënten van dit filter met een eindige woordlengte representeren. We nemen daarvoor het volgende filter met een constante rimpel:

1. de bemonsteringsfrequentie is 1 Hz;
2. de filterlengte is 128;
3. de impulsresponsie $h[n]$ is symmetrisch;
4. de doorlaatband gaat van 0 Hz tot 0,24 Hz;
5. de sperband gaat van 0,26 Hz tot 0,5Hz;
6. de versterking in de doorlaatband is 1 (0 dB);
7. de rimpel in de doorlaatband is 0,1 dB;
8. de maximale versterking in de sperband is −60 dB;
9. het aantal bits voor de coëfficiënten is 10.

Figuur 3.8 geeft de frequentieresponsie als de coëfficiënten met een oneindige precisie worden gerepresenteerd (dit geeft een constante rimpel in de sperband) en de frequentieresponsie bij afronden van de coëfficiënten naar 10 bit. Figuur 3.9 geeft de invloed in de doorlaatband (oneindig precisie geeft een constante rimpel in de doorlaatband en eindige precisie geeft een kleine afwijking). We zien dat de invloed in de sperband veel dramatischer is dan in de doorlaatband.



Figuur 3.8 De frequentieresponsie.



Figuur 3.9 De invloed in de doorlaatband.

3.4 Gedrag van een IIR filter

In dit vierde voorbeeld laten we de frequentieresponsies en de impulsresponsies zien van twee IIR filters. Als eerste voorbeeld nemen we een vierde orde filter met de volgende polen en nulpunten:

Polen:	$0.236 \pm 0.758j$;
	$0.318 \pm 0.285j$;
nulpunten:	$-0.538 \pm 0.843j$;
	$-0.898 \pm 0.441j$.

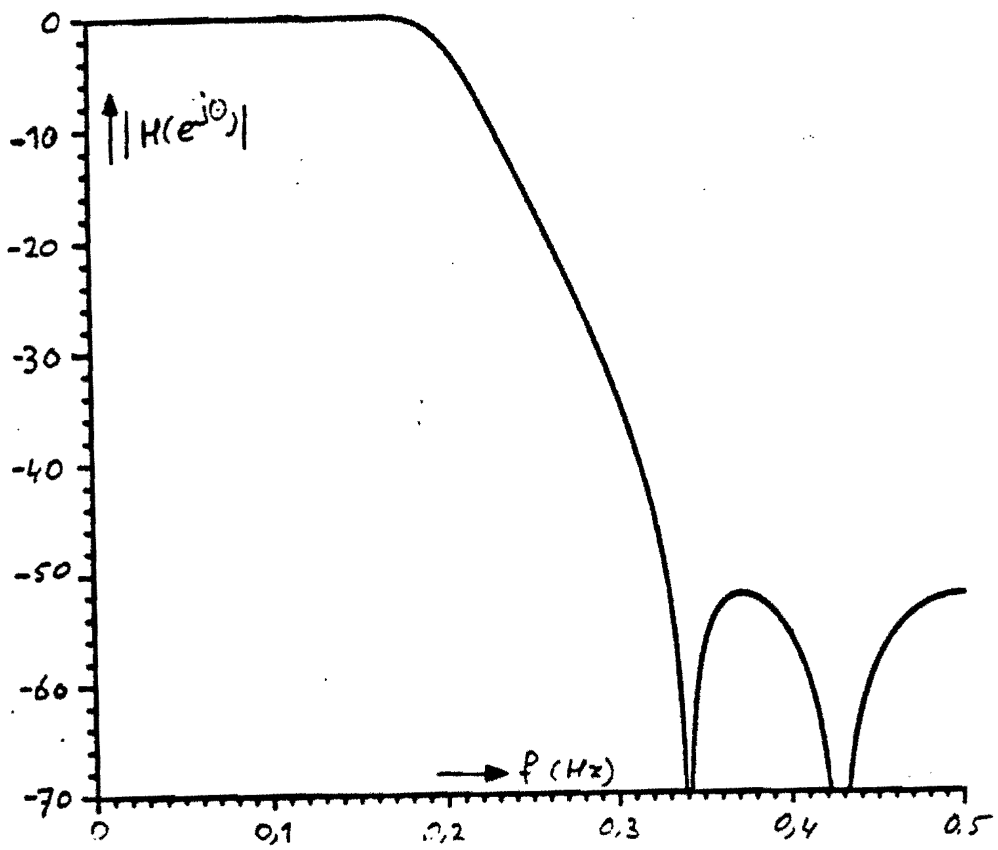
Figuur 3.10 geeft de frequentieresponsie $H(e^{j\theta})$ en figuur 3.11 geeft de bijbehorende impulsreponsie $h[n]$ van dit vierde orde recursieve filter.

Het twaalfde orde recursieve filter wordt beschreven met de parameters $a[0]$ tot en met $a[12]$ en $b[0]$ tot en met $b[12]$ zoals die hieronder gegeven zijn.

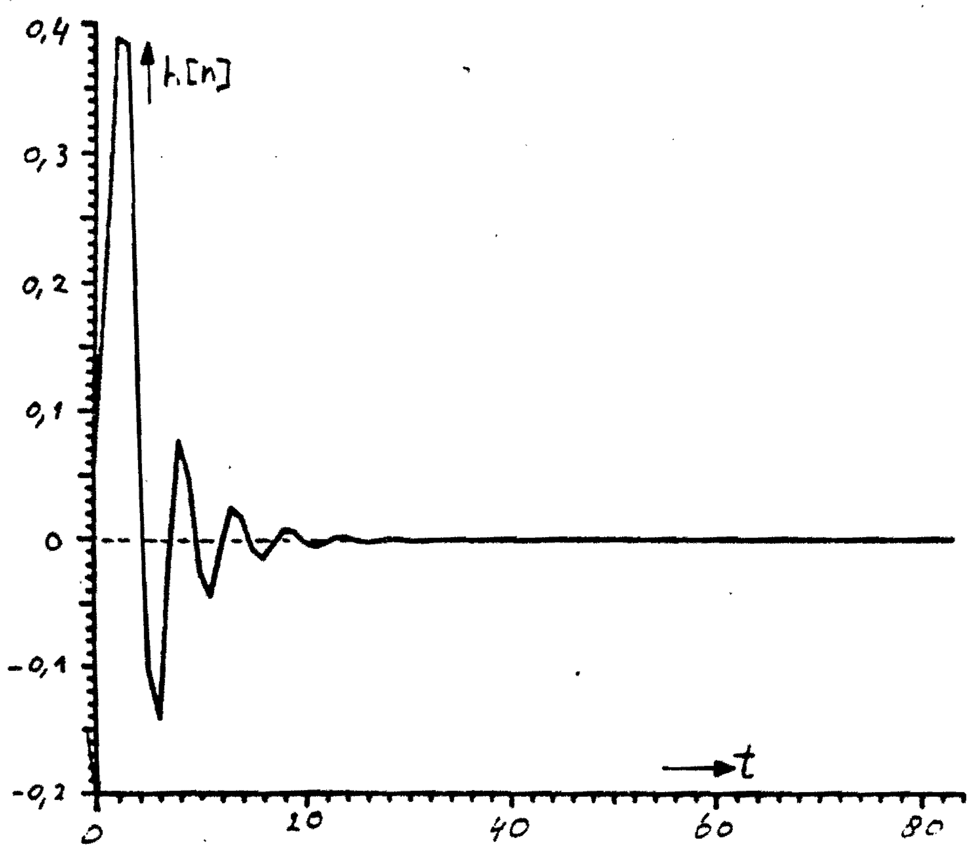
$a[0] = 4,510^{-9}$	$a[1] = 5410^{-9}$	$a[2] = 29710^{-9}$
$a[3] = 99010^{-9}$	$a[4] = 2227,510^{-9}$	$a[5] = 356410^{-9}$
$a[6] = 415810^{-9}$	$a[7] = 356410^{-9}$	$a[8] = 2227,510^{-9}$
$a[9] = 99010^{-9}$	$a[10] = 29710^{-9}$	$a[11] = 5410^{-9}$
$a[12] = 4,510^{-9}$		

$b[0] = 1$	$b[1] = -9,751711918$	$b[2] = 44,84241631$
$b[3] = -128,3755689$	$b[4] = 254,5339309$	$b[5] = -367,9092629$
$b[6] = 397,274152$	$b[7] = -322,7728367$	$b[8] = 195,7871297$
$b[9] = -86,46516334$	$b[10] = 26,39299384$	$b[11] = -5,001185764$
$b[12] = 0,4451259556$		

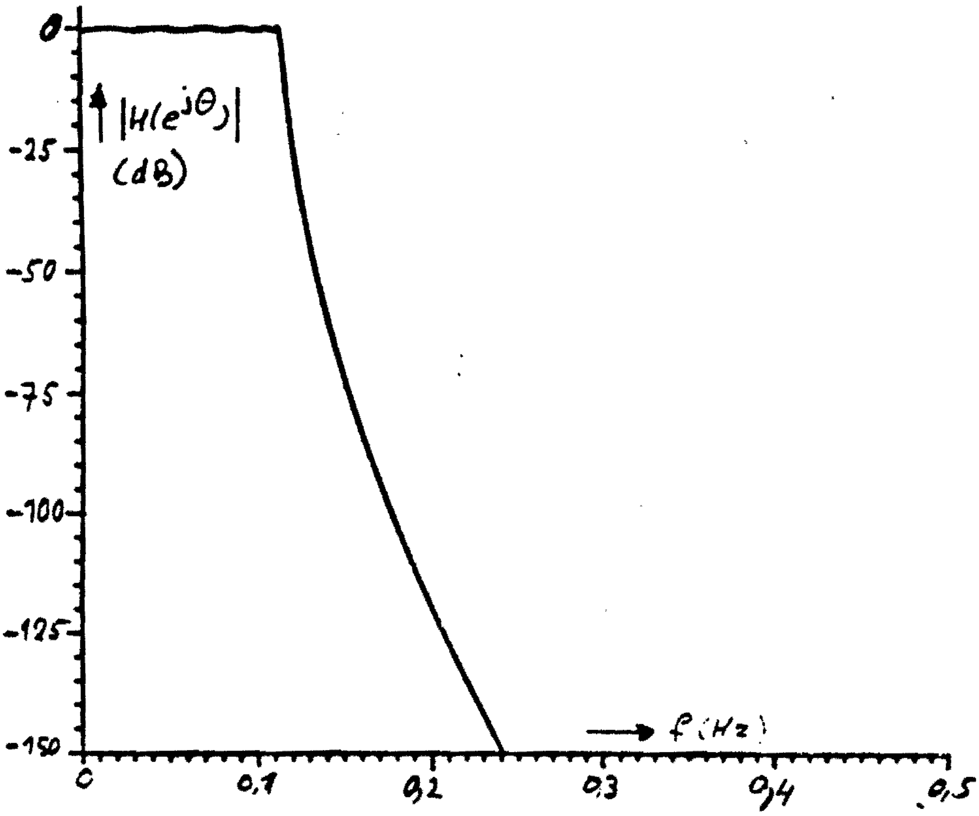
Figuur 3.12 geeft de bijbehorende frequentieresponsie en figuur 3.13 geeft de impulsresponsie $h[n]$ die hoort bij dit twaalfde orde tijddiscrete recursieve filter.



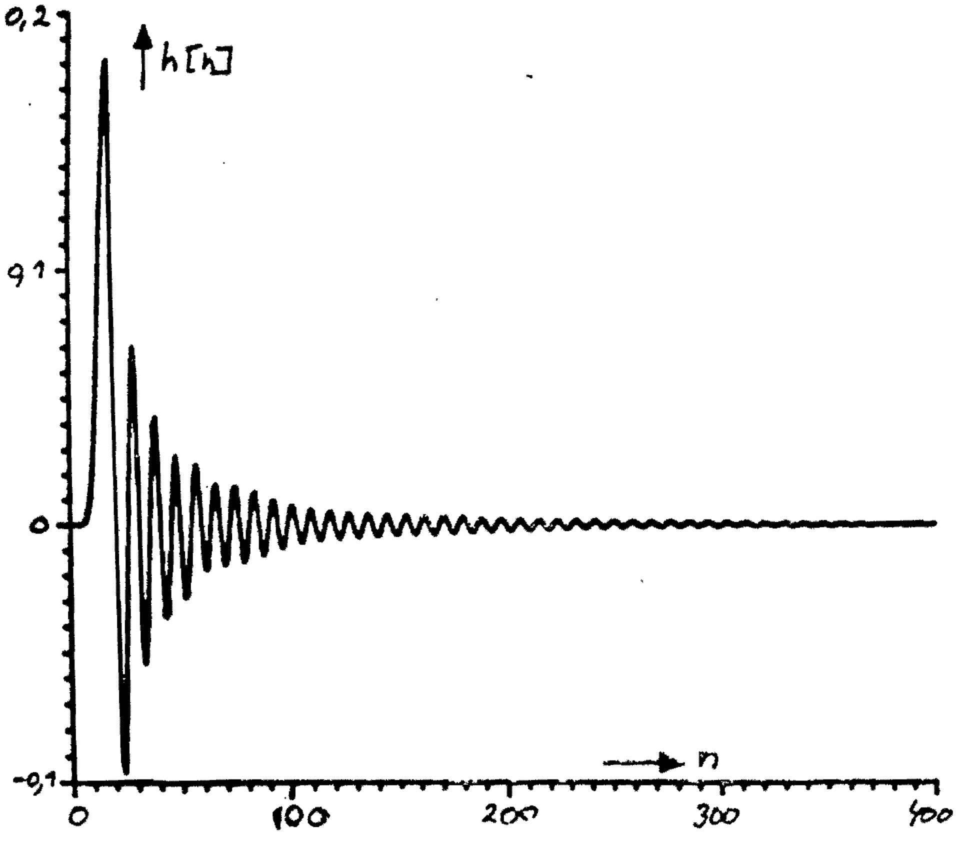
Figuur 3.10 De frequentieresponsie van het vierde orde filter.



Figuur 3.11 De impulsresposie van het vierde orde filter.



Figuur 3.12 De frequentieresponsie van het twaalfde orde filter.



Figuur 3.13 De impulsresposie van het twaalfde orde filter.

3.5 Leakage en windowing bij de FFT

In dit vijfde en laatste voorbeeld zullen we het "leakage effect" dat bij een DFT of een FFT optreedt laten zien. We gaan met behulp van het DIPRO programma SOURCE een sinus opwekken met:

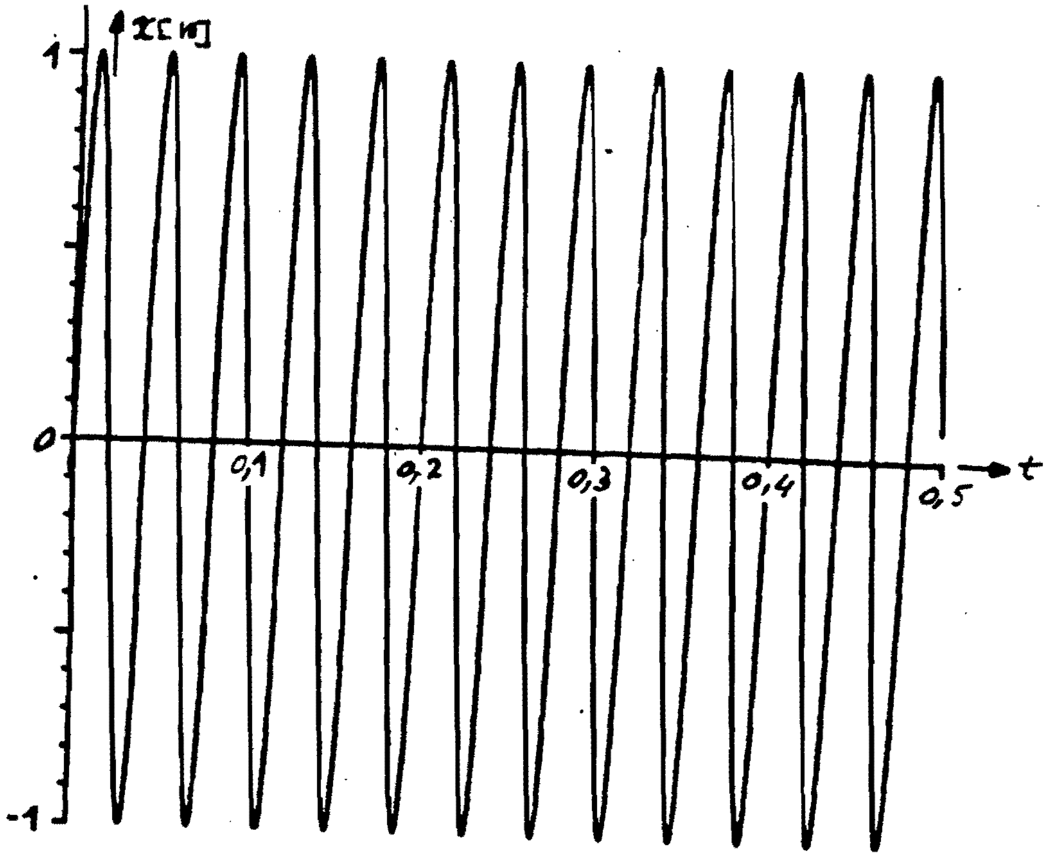
1. een signaalfrequentie van 25 Hz;
2. een bemonsteringsfrequentie van 2048 Hz;
3. een aantal monsters van 1024.

Als we dan het aantal perioden van de oorspronkelijke tijdcontinue sinus berekenen die opgewekt wordt, dan komen we op 12,5 perioden; zie figuur 3.14. Als we van dit tijdsignaal $x[n]$ de DFT berekenen krijgen we te maken met het "leakage effect", zie het DFT spectrum van figuur 3.15.

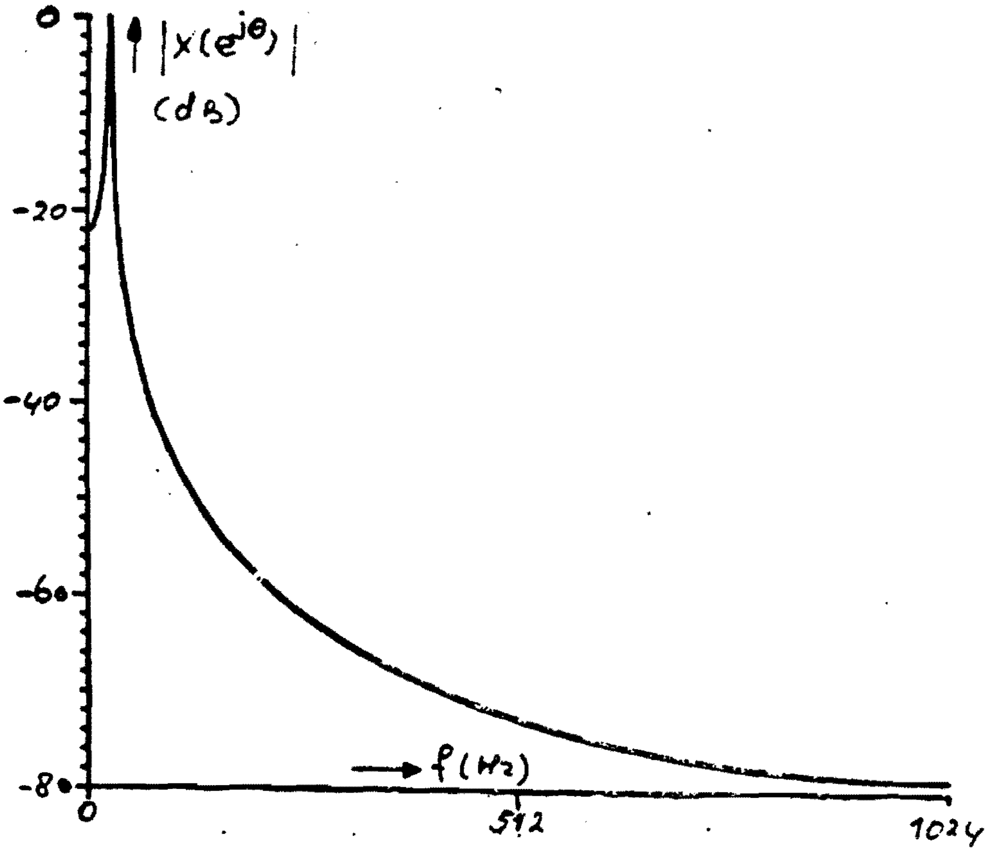
We kunnen het signaal $x[n]$ vermenigvuldigen met een zogenaamd Hanning-venster; dit geeft het signaal $y[n]$. Figuur 3.16 geeft zowel het sinusvormige signaal $x[n]$ als het signaal $y[n]$.

Figuur 3.17 geeft de DFT van $x[n]$ en van $y[n]$; de DFT van $y[n]$ heeft een veel smaller spectrum dan de DFT van $x[n]$. We zien dat de invloed van "leakage" drastisch verminderd is. Figuur 3.18 laat de invloed van de "leakage" in de buurt van de gewenste frequentie zien. We zien dat we de "leakage" hebben verminderd ten koste van een paar dB verlies bij de gewenste signaalfrequentie.

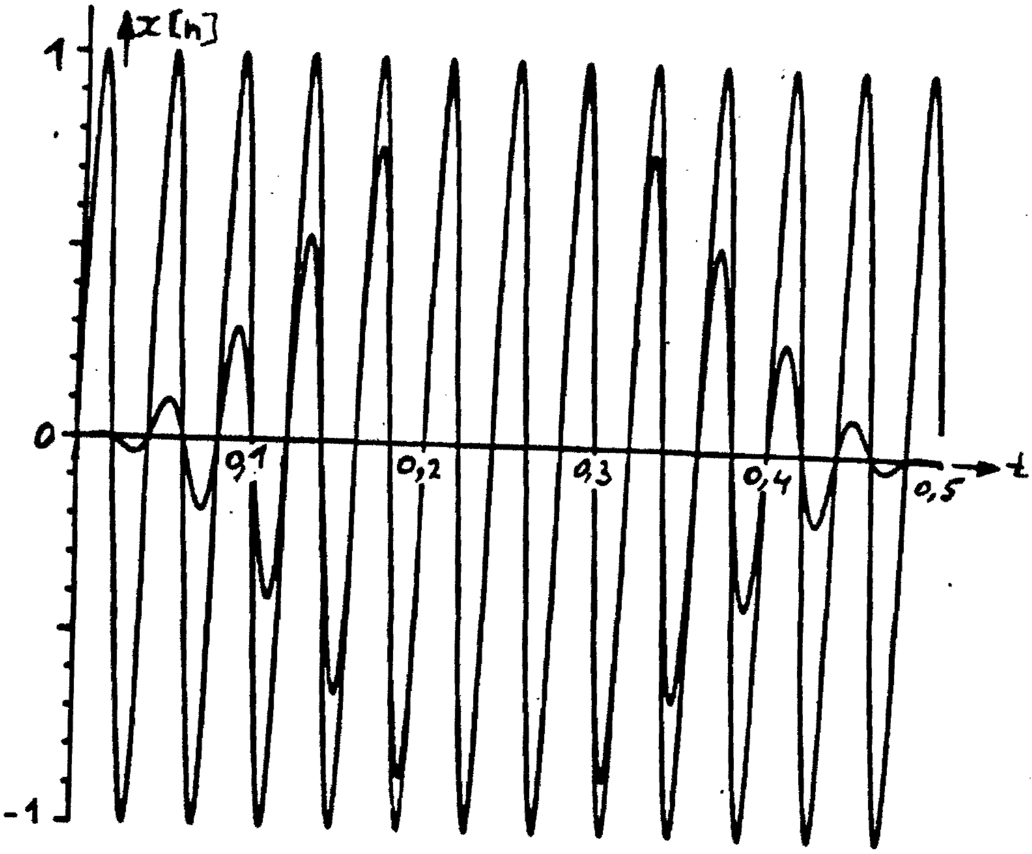
Tenslotte gaan we een sinusvormig signaal $x_1[n]$ opwekken met een signaalfrequentie van 24 Hz in plaats van 25 Hz; we hebben nu precies 12 perioden van het signaal. De resultaten van de DFT berekeningen zijn nu samengevat in figuur 3.19. We zien hier de DFT van $x[n]$ (de bovenste curve) van $y[n]$ (de middelste curve) en van $x_1[n]$ (de onderste curve). Voor de DFT van $x_1[n]$ vinden we inderdaad een enkele signaalfrequentie plus een ruisachtig spectrum afkomstig van afrondingsruis in de digitale rekenmachine.



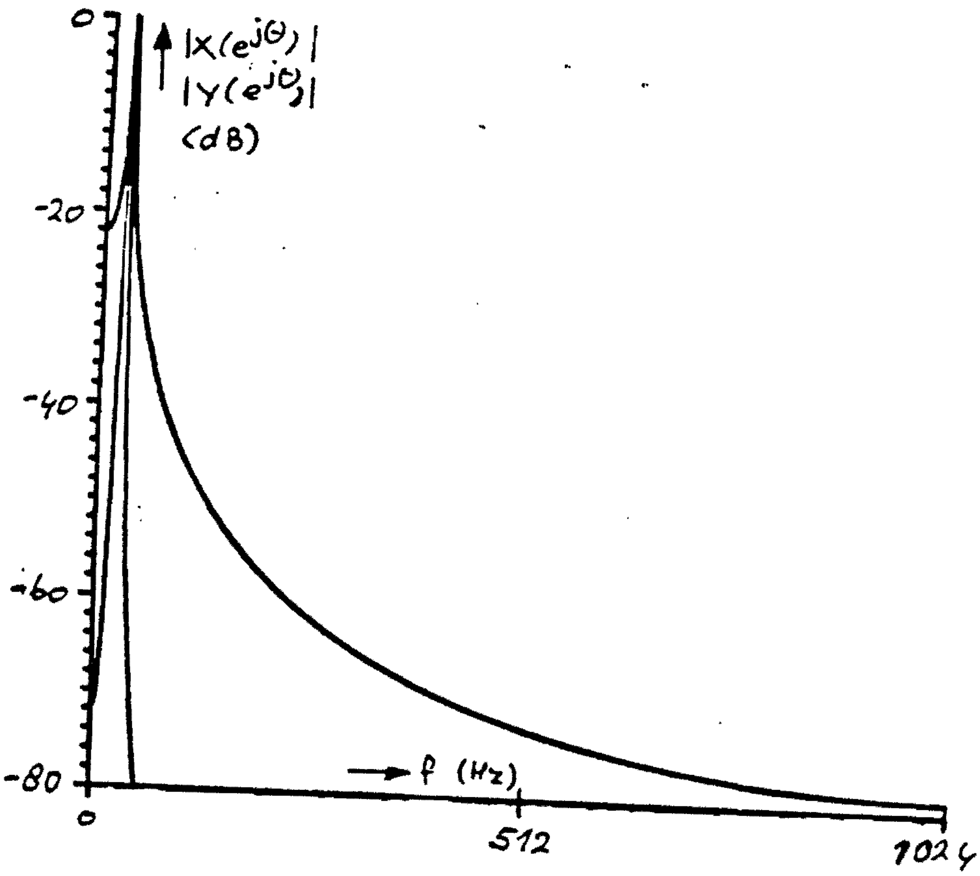
Figuur 3.14 Het sinusvormige signaal x[n].



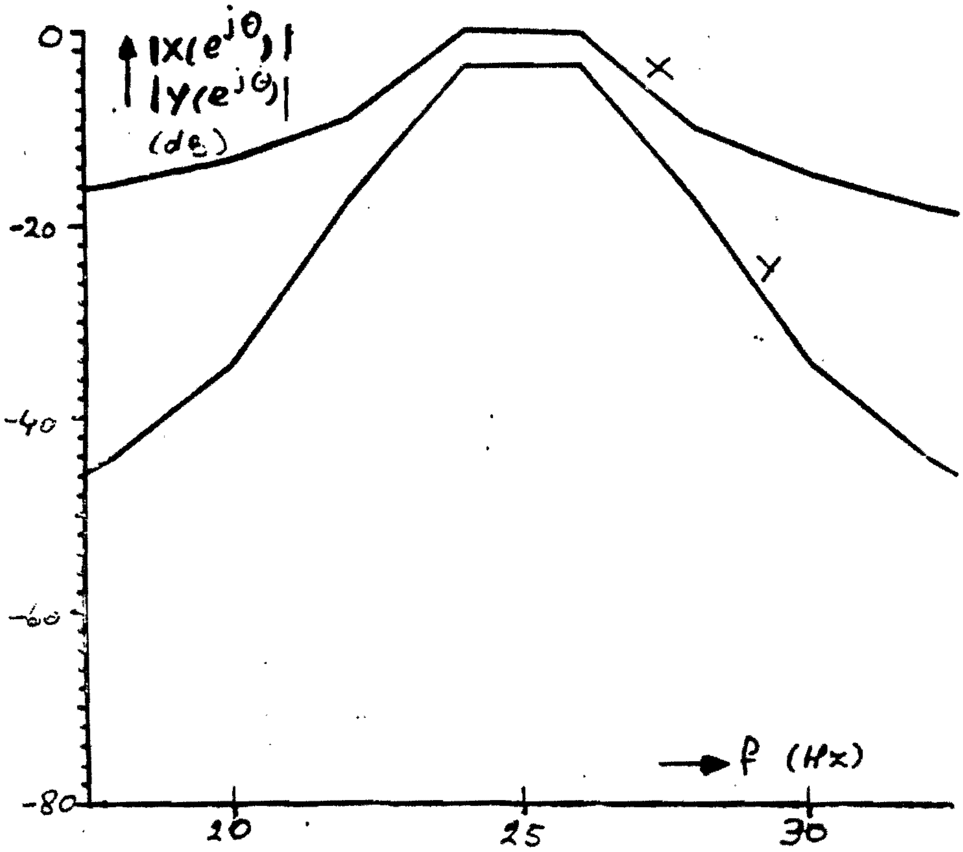
Figuur 3.15 De DFT van x[n].



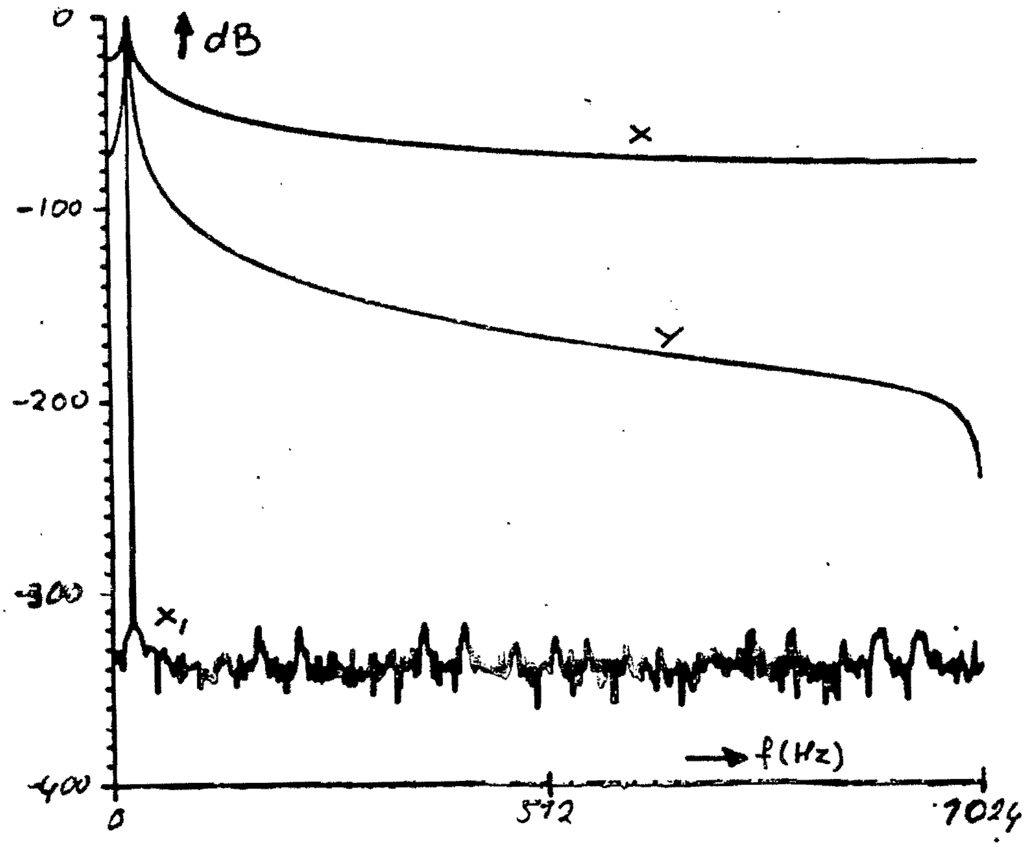
Figuur 3.16 De signalen x[n] en y[n].



Figuur 3.17 De DFT van x[n] en y[n].



Figuur 3.18 De DFT van x[n] en y[n] rondom 25 Hz.



Figuur 3.19 De DFT van x[n], y[n] en $x_1[n]$.

Appendix. Adressen van softwarefabrikanten

Signal Technology, Inc
Tel (800) 235-5787,
5951 Encina road,
Goleta CA 93117
USA.

Burr Brown Corporation,
Tel (602) 746-1111,
P.O. Box 11400,
Tucson, AZ 85734,
USA.

Atlanta Signal Processors, Inc.
Tel (404) 892-7265,
770 Sprong Street, N
Atlanta GA 30308
USA.

DSP Development Corporation,
Tel (617) 577-1133,
One Kendall square,
Cambridge MA 02139
USA.

The Athena Group, Inc
Tel (904) 371-2567,
3424 N.W. 31st Street,
Gainesville FL 32605
USA.

Momentum Data Systems, Inc
Tel (714) 548-3257,
1666 Newport blvd,
Costa Mesa CA 92627
USA.

Microcraf Corporation,
Tel (414) 241-8144,
P.O. Box 513,
Thiensville, Wisconsin 53092
USA.

Professor M. Kunt,
Departement d'Electricite,
Ecole Polytechnique Federale de Lausanne,
Ecublens,
Tel (4121) 472626,
CH-1015 Lausanne, Switzerland.

Hyperception,
9550 Skillman LB125 Ste. 316,
Dallas TX 75243
USA.

Loughborough Sound Images LSI,
Epinal way,
Loughborough Leics LE11 0QE,
England.

The math works Inc
Suite 250,
20 North Main St,
Sherborn, MA 01770,
USA

Dynamic Engineering,
Ambachtenlaan 21,
B-3030 Heverlee,
Belgie.

Comdisco Systems Inc,
919 E. Hillsdale Blvd,
Foster City CA 94404,
USA.

NAG central office,
Mayfield House,
256 Banbury road,
Oxford OX2 7DE
England.

National Instruments,
12109 Technology Boulevard,
Austin Texas 78727-6204,
USA.

MathSoft Inc,
One Kendall Sq,
Cambridge MA 02139,
USA.

Spectrum Software,
1021 S. Wolfe road,
Sunnyvale CA 94086,
USA.

DIGITALE SIGNAALBEWERKING

deel II.6

Digitale Beeldreconstructie in het Fourierdomein

door

J. Biemond

Technische Universiteit Delft

Twee-dimensionale signaalbewerking:

Digitale beeldreconstructie in het Fourierdomein.

door Dr.ir. J. Biemond

September 1988
Vakgroep Informatietheorie
Faculteit der Elektrotechniek
Technische Universiteit Delft.

INHOUDSOPGAVE

1. INLEIDING

Afbeeldingen of beelden van een scène zijn in meer of mindere mate behept met vervorming en ruis. Dit komt omdat geen enkel beeldvormend systeem (elektronenmicroscoop, camera, telescoop) een perfect, onvervormd beeld levert. Ruis kan geïntroduceerd worden tijdens de opname, in de transmissieweg of gedurende het registratieproces, terwijl bij vervorming gedacht kan worden aan aberraties van lenzen, bewegingsonscherpte, onjuiste focussering, niet-lineaire eigenschappen van fotografische materialen, geometrische vervorming of, bij telescoopfoto's, vervorming ten gevolge van atmosferische turbulenties.

Is een afbeelding van groot belang om beslissingen te nemen of erg zeldzaam (satellietbeelden, elektronenmicroscoopbeelden, historische foto's), dan is er behoefte aan technieken die de vervorming en de ruis minimaliseren en daardoor een beter beeld opleveren. Wordt gebruik gemaakt van een computer of van special purpose hardware dan spreekt men in dit verband van digitale beeldrekonstruktie. Ook de foto-industrie toont tegenwoordig belangstelling voor de digitale beeldrekonstruktie, met name voor het restaureren (opscherpen) van bewogen of gedefocusseerde kleurenfoto's ten behoeve van de consumentenmarkt.

In dit rapport zal aandacht worden besteed aan de digitale rekonstruktie van kleurenfoto's m.b.v. in het frekwentiedomein werkende rekonstruktiefilters [1, 3, 8, 20]. Voor de implementatie van deze filters stond een "general purpose" computer systeem ter beschikking, waaraan een zogenaamde arrayprocessor is toegevoegd. Deze arrayprocessor maakt het mogelijk om grote hoeveelheden data, denk aan beelden, zeer snel te bewerken. Vanwege de speciale architectuur, met een z.g. "pipeline" en een gescheiden data- en programmageheugen, is de arrayprocessor tevens in staat om een twee-dimensionale fast Fourier transform (FFT) daadwerkelijk zeer snel uit te voeren, dit in tegenstelling tot conventioneel gestruktureerde computers. Met name deze laatste eigenschap maakt dat filters, die in het frekwentiedomein werken, bij uitstek geschikt zijn voor implementatie op een arrayprocessor. De rekentijd voor het rekonstrueren van één enkele kleurenfoto kan hierdoor teruggebracht worden van vele minuten (of zelfs uren) tot enkele sekonden.

Alvorens enige rekonstruktiefilters te formuleren, zal in hoofdstuk 2 eerst worden ingegaan op de modelvorming van het beeldvormend systeem en van het registratieproces. Tevens zullen twee veel voorkomende typen vervorming, bewegingsonscherpte en defocussering, worden besproken, en zal worden ingegaan op het digitaliseren van beelden.

In hoofdstuk 3 zal vervolgens een aantal in het frekwentiedomein werkende rekonstruktiefilters worden besproken, die gebaseerd zijn op het in hoofdstuk 2 opgestelde model van het beeld(ver)vormend systeem en die gebruik maken van verschillende optimaliseringscriteria.

In hoofdstuk 4 zal worden ingegaan op de gevolgen van de eindige afmetingen van een beeld voor de rekonstruktie. Het gebruik van de 2-D FFT impliceert dat het beeld periodiek wordt verondersteld in horizontale en vertikale richting. Bestaan er aanzienlijke grijswaarden-verschillen tussen boven- en onderrand en/of linker- en rechterrand van het beeld dan uit zich dat in strepenpatronen ("ringing") in het rekonstruktieresultaat, hetgeen in sommige gevallen zeer storend kan zijn. Om deze randeffekten zoveel mogelijk te onderdrukken worden twee interpolatiemethoden besproken, lineaire interpolatie en interpolatie m.b.v. een derde-orde polynoom, om tegenoverliggende randen zoveel mogelijk op elkaar te doen aansluiten.

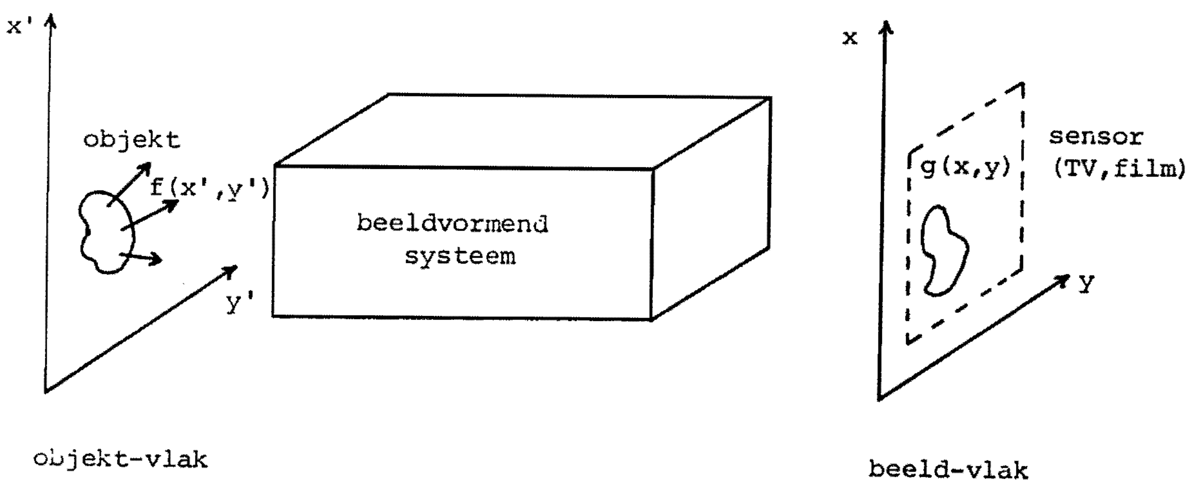
In hoofdstuk 5 wordt beknopt de werking van de AP500 arrayprocessor van Analogic besproken.

2. MODELVORMING

2.1. Inleiding

Alvorens we tot beeldrekonstruktie kunnen overgaan, dient de vervorming die een objekt (oorspronkelijke scène) ondergaat t.g.v. beeldvorming en -registratie, gemodelleerd te worden. In dit hoofdstuk wordt aandacht besteed aan het opstellen en uitwerken van zo'n model. Dit betekent uiteraard dat een vereenvoudigde doch handzame voorstelling wordt gegeven van hoe in de praktijk een beeld wordt gevormd c.q. vervormd.

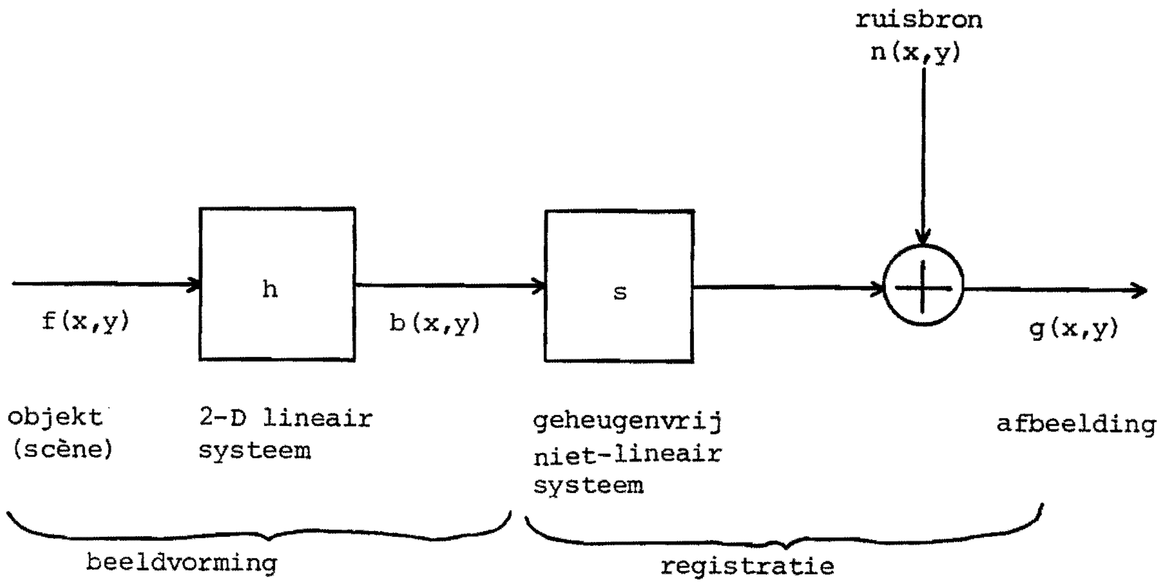
Uitgegaan wordt van de volgende schematische voorstelling van een beeldvormend systeem, zoals weergegeven in figuur (2.1).



Figuur (2.1). Beeldvorming en -registratie.

Uitgestraalde danwel gereflecteerde energie (b.v. lichtgolven, röntgenstraling) van een objekt in het objektvlak doorloopt een twee-dimensionaal beeldvormend systeem (b.v. een optisch lenzen-stelsel), en creëert een beeld in het beeldvlak. Dit beeld wordt afgetast met een sensor en geregistreerd op film (fotochemisch) of TV (foto-elektrisch). Vanwege fysische onvolkomenheden zullen er altijd enige degradaties optreden tijdens de beeldvorming en -registratie.

Figuur (2.2) beschrijft in blokschematische vorm de beeldvorming en -registratie van figuur (2.1).



Figuur (2.2). Blokdiagram van het beeldvormings- en registratieproces.

De oorspronkelijke scène of objekt in het objektvlak wordt beschreven door de reëele funktie f van de spatiële variabelen x en y. Voor kleurenbeelden is f een vectoriële funktie met als elementen de primaire kleurcomponenten rood, groen en blauw. Het beeldvormend systeem wordt in onze benadering gemodelleerd als een twee-dimensionaal (2-D) lineair systeem met een 2-D impulsresponsie of puntspreidingsfunktie (PSF) h. Aan de uitgang van dit systeem ontstaat het vervormde beeld b(x,y). De lineariteitsaanname van het systeem is voor een aantal veel voorkomende typen vervorming zoals bewegingsonscherpte en defocussering een goede benadering. Het vervormde beeld wordt vervolgens afgetast en geregistreerd. Het registratieproces betreft een niet-lineaire geheugenvrije transformatie (puntoperatie), die we symboliseren met de funktie s in figuur (2.2).

Tenslotte wordt, omdat de beeldsensor niet ideaal is, additieve ruis toegevoegd aan het geregistreerde beeld. Op de aanname van additiviteit zal in paragraaf (2.3) nader worden ingegaan. We nemen eenvoudigheidshalve aan, dat het aftasten en registreren van het beeld de enige bron van ruis is.

De beeldvorming kan nu beschreven worden m.b.v. de volgende superpositie-integraal met PSF h:

$$b(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x,y,x',y')\ f(x',y')\ dx'dy'. \tag{2.1}$$

Het geregistreerde beeld wordt dan gegeven door

$$g(x,y) = s\{b(x,y)\} + n(x,y). \tag{2.2}$$

Het beeldrekonstruktieprobleem komt nu neer op het schatten van f, gegeven de PSF h, het geregistreerde beeld g, de niet-lineaire funktie s en enige a priori kennis (vaak van statistische aard) omtrent de ruis n en het objekt (scène) f.

2.2. Beeldvorming

De puntspreidingsfunktie h in vergl. (2.1) is afhankelijk van alle vier de coördinaatvariabelen (x,y,x',y'). De PSF is voor verschillende objektpunten verschillend en wordt aangeduid als "verschuivingsvariante" PSF ("space-variant point-spread function", SVPSF). Als de PSF voor elk objektpunt gelijk is, betekent dit dat de PSF alleen afhankelijk is van x-x' en y-y'. Het beeldvormend systeem dat aldus ontstaat heeft een verschuivingsinvariante PSF ("space-invariant point-spread function", SIPSF).
De superpositie-integraal in vergl. (2.1) gaat dan over in een convolutie-integraal

$$b(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x-x',y-y')\ f(x',y')\ dx'dy', \tag{2.3}$$

welke ook wel wordt genoteerd als

$$b(x,y) = h(x,y) \circledast f(x,y). \tag{2.4}$$

Indien we op een convolutie-integraal de twee-dimensionale Fourier-transformatie toepassen, gaat deze over in een vermenigvuldiging in het frekwentiedomein, zodat

$$B(\omega_1,\omega_2) = H(\omega_1,\omega_2)\ .\ F(\omega_1,\omega_2), \tag{2.5}$$

waarin B, H en F resp. de Fourier-getransformeerden zijn van b, h en f, en ω_1 en ω_2 de spatiële frekwenties voorstellen.

Een verdere vereenvoudiging is mogelijk door het beeldvormend proces scheidbaar te veronderstellen. Een puntspreidingsfunktie is scheidbaar, indien deze kan worden geschreven als

$$h(x,y,x',y') = h_1(x,x')\ h_2(y,y'), \tag{2.6}$$

of, in het geval van een SIPSF als

$$h(x-x',y-y') = h_1(x-x')\ h_2(y-y'). \tag{2.7}$$

Is de puntspreidingsfunktie scheidbaar, dan kunnen de integraties van vergl. (2.1) achter elkaar worden uitgevoerd en wordt

$$b(x,y) = \int_{-\infty}^{\infty} h_1(x,x') \left[\int_{-\infty}^{\infty} h_2(y,y')f(x',y')dy'\right] dx' \tag{2.8a}$$

$$= \int_{-\infty}^{\infty} h_2(y,y') \left[\int_{-\infty}^{\infty} h_1(x,x')f(x',y')dx'\right] dy', \tag{2.8b}$$

in het geval van een SVPSF, en

$$b(x,y) = \int_{-\infty}^{\infty} h_1(x-x') \left[\int_{-\infty}^{\infty} h_2(y-y')f(x',y')dy'\right] dx' \tag{2.9a}$$

$$= \int_{-\infty}^{\infty} h_2(y-y') \left[\int_{-\infty}^{\infty} h_1(x-x')f(x',y')dx'\right] dy', \tag{2.9b}$$

voor een SIPSF.

De scheidbaarheid van de puntspreidingsfunktie is ook door te voeren in het Fourier-domein. Twee-dimensionale Fourier-transformatie van vergl. (2.9) geeft

$$B(\omega_1,\omega_2) = H_1(\omega_1)\ H_2(\omega_2)\ F(\omega_1,\omega_2). \tag{2.10}$$

Vergl. (2.9) en vergl. (2.10) tonen het beeldvormend proces als onafhankelijke horizontale en vertikale processen, in resp. het plaats- en het frekwentiedomein.

De voornoemde vergelijkingen zijn de fundamentele mathematische beschrijvingen van het beeldvormingsproces, waarin de puntspreidingsfunktie een centrale rol speelt. Als voorbeeld zullen twee veel voorkomende gevallen van vervorming worden besproken, te weten bewegingsonscherpte en defocussering, beiden in termen van de puntspreidingsfunktie.

2.2.1. Horizontale bewegingsonscherpte (translatie)

Bewegingsonscherpte doet zich vooral voor in de fotografie, waar een bepaald eindig belichtingsinterval $[0,T]$ geldt. Stel, dat er een relatieve beweging tussen objekt en beeldvormend systeem is (of meer algemeen, tussen objekt en te vormen beeld), dan zal een objektpunt zich uitbreiden tot een lijn in het beeld, volgens de richting van de beweging. Indien deze beweging gedurende het interval $[0,T]$ een constante snelheid v heeft, en bovendien slechts in horizontale richting is, dan zal het beeld vervormd zijn met horizontale lineaire bewegingsonscherpte. De PSF $h(k,\ell)$ heeft dan de volgende vorm [18]:

$$h(x,y) = \frac{1}{vT} \qquad 0 \leq y \leq vT;\ x=0 \qquad (2.11)$$

$$= 0 \qquad \text{elders.}$$

De diskrete versie hiervan, die we bij de rekonstruktie zullen gebruiken (zie ook par. 2.4), is:

$$h(k,\ell) = \frac{1}{L+1} \qquad \ell=0,1,\ldots,L;\ k=0 \qquad (2.12)$$

$$= 0 \qquad \text{elders.}$$

Hierin is L het (geheel) aantal punten waarover de beweging zich uitstrekt. Duidelijk is, dat indien L=0, er geen vervorming is.

2.2.2. Defocussering

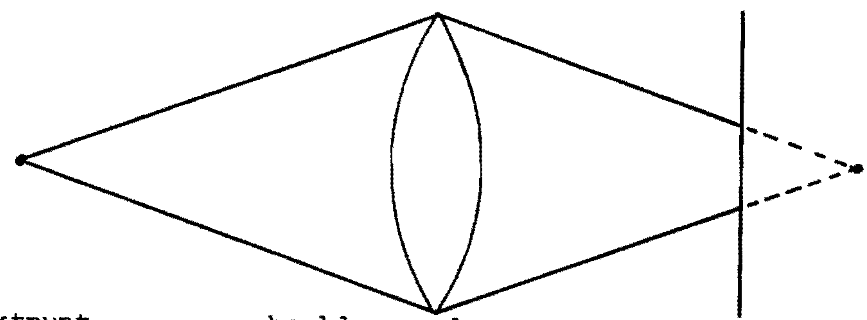
In figuur (2.3) is schematisch een mogelijke oorzaak van defocussering toegelicht. In dit geval valt de afbeelding van één objektpunt via het beeldvormend systeem niet exact op het vlak van het te vormen beeld, maar erachter.

Het gevolg is, dat indien het beeldvormend systeem een cirkelvormige apertuur heeft, dit ene objektpunt zich in het beeld zal uitsmeren tot een cirkelvormige vlek. De PSF heeft dan de volgende vorm [18]:

$$h(x,y) = a(x,y) \qquad x^2+y^2 \leq R^2$$

$$= 0 \qquad \text{elders.} \qquad (2.13)$$

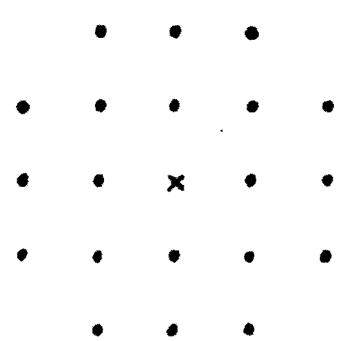
Hierin is $a(x,y)$ een nader vast te stellen funktie van x en y. Diskretisering van de continue PSF op een vierkant raster geeft



Figuur (2.3). Defocusseringsonscherpte.

$$h(k,\ell) = a(k,\ell) \qquad k^2+\ell^2 \leqq R^2$$
$$= 0 \qquad \text{elders.} \tag{2.14}$$

Omdat k en ℓ hier slechts gehele waarden mogen aannemen, zal de voorheen continue vlek zich nu voordoen als een rooster van punten, die tezamen een benadering van een cirkel vormen. In figuur (2.4) is een voorbeeld gegeven voor $R^2=7$.



Figuur (2.4). Roosterpatroon van de cirkel $k^2+\ell^2 \leqq 7$.

Goede benaderingen van een cirkel worden gegeven bij o.a. $R^2=3,7,12$ en 19 [18]

Nu rest nog de vraag hoe we de funktie $a(k,\ell)$ zullen invullen. Hiervoor zijn twee modellen gebruikt, het zogenaamde "pillen-doos" ("pill-box") model, en het "cosinus-kwadraat" ("cosine-squared") model [7].

(1) Pillen-doos model.

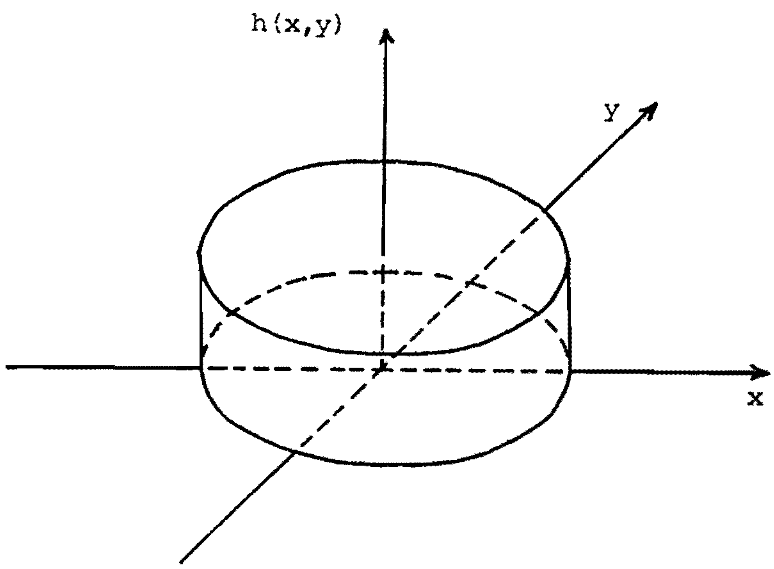
Dit is het eenvoudigste model voor defocussering. Er wordt verondersteld dat de waarde van alle punten even groot is zodat de diskrete PSF wordt:

$$h(k,\ell) = \frac{1}{c} \qquad k^2+\ell^2 \leqq R^2$$
$$= 0 \qquad \text{elders.} \tag{2.15}$$

Hierin moet c voldoen aan

$$c = \sum_{k=-\infty}^{\infty} \sum_{\ell=-\infty}^{\infty} h(k,\ell), \tag{2.16}$$

om te zorgen dat de totale som gelijk aan 1 is. Dit wil niets anders zeggen, dan dat er bij de beeldvorming geen energie verloren gaat of bijkomt. Tenslotte is de continue funktie nog geschetst in figuur (2.5).



Figuur (2.5). Pillen-doos model voor defocussering.

(2) Cosinus-kwadraat model.

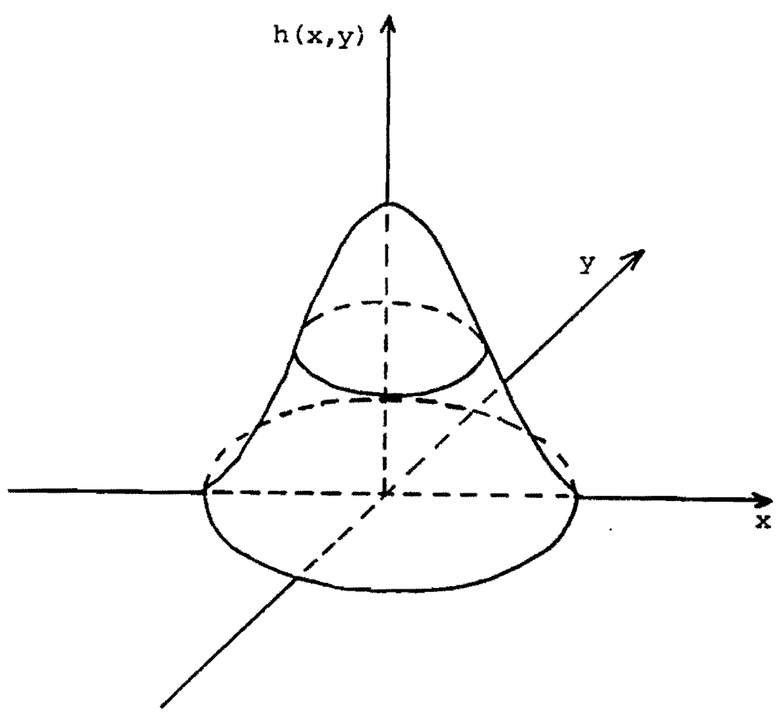
Een meer met de werkelijkheid overeenkomende PSF voor defocussering wordt verkregen met een cosinus-kwadraat funktie [7]. De continue mathematische beschrijving hiervan is:

$$h(x,y) = \frac{1}{c}\cdot\cos^2\left(\frac{x^2+y^2}{R^2}\frac{\pi}{2}\right) \qquad x^2+y^2 \leqq R^2$$
$$= 0 \qquad \text{elders,} \tag{2.17}$$

met

$$c = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x,y)\,dxdy. \tag{2.18}$$

De continue puntspreidingsfunktie is geschetst in figuur (2.6).



Figuur (2.6). Cosinus-kwadraat model voor defocussering.

De definitie van de cosinus-kwadraat in het diskrete geval is zo ingericht, dat de waarden waar de cosinus nul wordt, dat is bij $-\frac{\pi}{2}$ en $+\frac{\pi}{2}$, op de beide assen juist één punt buiten de (diskrete) cirkel vallen, dus

$$h(k,\ell) = \frac{1}{c}\cos^2\left(\frac{k^2+\ell^2}{R^2+1}\frac{\pi}{2}\right) \qquad k^2+\ell^2 \leqq R^2$$
$$= 0 \qquad \text{elders,} \tag{2.19}$$

waarin c weer aan vergl. (2.16) voldoet.

2.3. Beeldregistratie

Beeldregistratie vindt in hoofdzaak volgens twee manieren plaats: foto-elektrisch of fotochemisch [14]. Voorbeelden hiervan zijn resp. registratie met een televisiecamera of op film. Als voorbeeld zal hieronder de registratie op film nader worden besproken.

Fotografische film is gebaseerd op de eigenschappen van zilver-zouten (b.v. AgCl, AgBr) om beelden te registreren. Zilver-zouten veranderen bij blootstelling aan licht, zodat de inwerking van bepaalde chemicalieën, ontwikkelaars, resulteren in de neerslag van vrij zilver. Hurter en Driffield [16] hebben op experimentele wijze gevonden, dat de massa van het neergeslagen zilver lineair verband houdt met de logaritme van de totale belichting E, welke gedefinieerd is als de integraal van de lichtintensiteit over het tijdsinterval van de belichting. Naast dit verband is er een gebied van verzadiging, waar alle zilver is neergeslagen, en een gebied waar wat zilver is neergeslagen, zelfs in de afwezigheid van licht (een ekwivalent in de TV-techniek wordt wel "donkerstroom" genoemd). Deze relatie is grafisch weergegeven als in figuur (2.7), waar D de optische dichtheid voorstelt. In het gebied waar de film-responsie lineair is, geldt een zuiver logaritmisch verband, volgens

$$D = \gamma \log E - D_o. \tag{2.20}$$

In vergl. (2.20) is γ de helling van het lineaire gebied. Dit is de z.g. gamma van de film. Een film met een negatieve γ staat bekend als een positief materiaal, en omgekeerd. D_o is de offset, omdat het lineaire gedeelte niet door de oorsprong loopt. We nemen aan dat deze offset verwaarloosbaar is, zodat we het model voor beeldregistratie van vergl. (2.2) m.b.v. vergl. (2.20) kunnen schrijven als

$$g(x,y) \simeq \gamma \log \{b(x,y)\} + n(x,y). \tag{2.21}$$

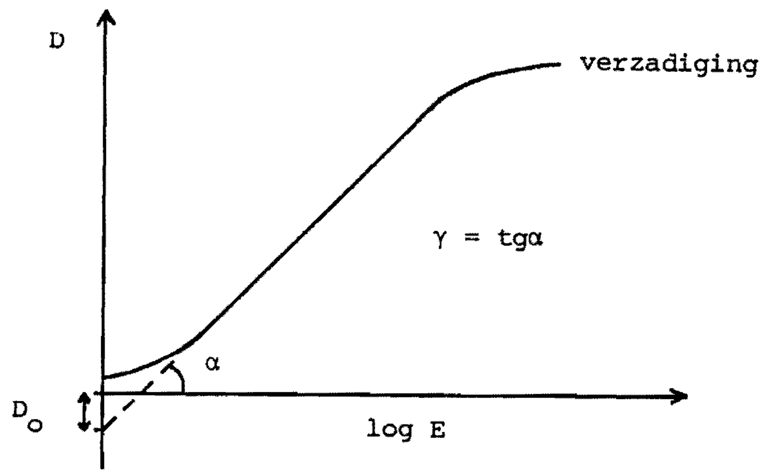


Fig. (2.7). Film responsie-kromme.

Als we tenslotte aannemen, dat het contrast in het beeld b niet te hoog is, dan kan de logaritme benaderd worden met een lineaire funktie, zodat vergl. (2.21) overgaat in

$$g(x,y) \simeq \gamma\, b(x,y) + n(x,y). \tag{2.22}$$

Hoewel dit niet noodzakelijk is, wordt voor het gemak meestal $\gamma=1$ genomen.

De additiviteit van de ruis n in vergl. (2.2) is gerechtvaardigd op basis van de aard van ruis in beeldsensoren [13]. Voor film-registratie systemen is de ruis hoofdzakelijk afkomstig van de korrel-struktuur van de film. Ten eerste zijn de grootte en de vorm van de korrels willekeurig. Ten tweede zullen twee schijnbaar identieke korrels niet gelijk reageren op dezelfde hoeveelheid licht. Deze willekeurigheid van het filmmateriaal resulteert in de zogenaamde korrelruis.

Korrelruis is een zeer complexe signaal-afhankelijke vorm van ruis. Een vereenvoudigde aanname, die de werkelijkheid niet al te veel geweld aandoet, is de ruis additief te veronderstellen met een constante variantie, met betrekking tot de optische dichtheid van de film. Bovendien hebben pogingen tot beeldrekonstruktie op basis van een model met signaal-afhankelijke ruis aangetoond, dat hier slechts zeer weinig mee te winnen is [22], zodat in het vervolg van dit verslag deze additiviteitsaanname wordt gehanteerd.

2.4. Digitalisering

Omdat het geregistreerde beeld digitaal bewerkt wordt, zal het eerst bemonsterd en gekwantiseerd moeten worden. De bemonsteringsfrekwenties moeten voldoen aan het Nyquist-theorema, dat ook in twee dimensies voor beelden bewezen kan worden [14]. Een beeld is in het algemeen niet bandbreedte-begrensd, daarom kunnen vouweffekten optreden tijdens het bemonsteren van het beeld. In de één-dimensionale signaalverwerking wordt het vouweffekt voorkomen door laagdoorlaatfilters te gebruiken, die afvallen tot nul bij de Nyquist-frekwentie van de bemonsteraar. In de digitale beeldverwerking is deze analoge filtering van de data niet zo eenvoudig. Het is soms mogelijk om een copie van het origineel te maken dat iets gedefocusseerd is, echter, een copie introduceert extra breedbandige korrelruis.

Vouweffekten in beelden zijn het meest hinderlijk wanneer de visuele patronen periodiek zijn. Bemonsteren van zo'n periodieke struktuur met een te lage frekwentie geeft Moiré-patronen in de weergave; de effekten hiervan zijn vaak wel storend, maar niet desastreus, en kunnen eenvoudig gedetecteerd worden, waarna een hogere bemonsteringsfrekwentie kan worden gekozen. Het totale effekt van zelfs een bescheiden vouweffekt kan vaak niet worden gedetecteerd in het bewerkte of onbewerkte beeld. Indien een groot gedeelte, circa 95 procent van de spatiële frekwentie-energie van het beeld beneden de Nyquist frekwentie ligt, dan veroorzaakt het vouweffekt geen zichtbare vervorming [15].

Als het beeld eenmaal is bemonsterd, dan moeten de bemonsteringen gekwantiseerd worden, zodat het beeld digitaal opgeslagen kan worden. Dit betekent dat het continue bereik van de mogelijke waarden van elke bemonstering opgedeeld moet worden in diskrete gebieden. Elk gebied kan dan door een bepaald getal in de computer gerepresenteerd worden. Kwantiseringsfouten zijn ekwivalent aan optelling van een stochastisch proces (ruis) bij de originele data. De keuze van de beslissingsgebieden wordt vaak bepaald door de wens om deze z.g. kwantiseringsruis te minimaliseren [3].

Combineren we nu de beeldvormings-vergelijking van vergl. (2.3) en de registratie-vergelijking van vergl. (2.22) met γ=1, dan komen we tot het in dit verslag gebruikte vervormingsmodel

$$g(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x-x',y-y')\ f(x',y')\ dx'dy' + n(x,y), \tag{2.23}$$

welke ook geschreven kan worden als

$$g(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x',y')\ f(x-x',y-y')\ dx'dy' + n(x,y). \tag{2.24}$$

Schrijven we vergl. (2.24) in diskrete vorm, dan geeft dit

$$g(k,\ell) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} h(m,n)\ f(k-m,\ell-n) + n(k,\ell), \tag{2.25}$$

waarin k,ℓ,m en n diskrete variabelen zijn. Deze vergelijking is de algemene uitdrukking voor een 2-D diskrete convolutie, waarin de PSF $h(m,n)$ een oneindig uitgestrekte impulsresponsie ("infinite-duration impulse response", IIR) bezit. In de werkelijkheid wordt de waarneming g in het punt (k,ℓ) slechts bepaald door beeldpunten van het objekt f die in een kleine omgeving rond het punt (k,ℓ) liggen, terwijl bijdragen van punten die ver van het punt (k,ℓ) verwijderd zijn onbelangrijk worden geacht. Met andere woorden, het beeldvormend systeem kan goed worden benaderd door een systeem met een eindig uitgestrekte impulsresponsie ("finite-duration impulse response", FIR), zodat de sommatie van vergl. (2.25) slechts behoeft te worden uitgevoerd over een eindig aantal punten. In dat geval kan vergl. (2.25) worden geschreven als

$$g(k,\ell) = \sum_{m=-m_1}^{m_2} \sum_{n=-n_1}^{n_2} h(m,n)\ f(k-m,\ell-n) + n(k,\ell), \tag{2.26}$$

waarin het gebied $(-m_1 \le m \le m_2,\ -n_1 \le n \le n_2)$ de afmetingen ("support") van de PSF $h(m,n)$ bepaalt.

Fourier-transformatie van vergl. (2.26) levert ons vervolgens het vervormingsmodel in het frekwentiedomein. De 2-D diskrete Fourier-transformatie is gedefinieerd als

$$F(u,v) = \sum_{k=0}^{M-1} \sum_{\ell=0}^{N-1} f(k,\ell)\ \exp\left[-\ 2\pi i\ \left(\frac{ku}{M} + \frac{\ell v}{N}\right)\right] \tag{2.27a}$$

$$f(k,\ell) = \frac{1}{M.N} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v)\ \exp\left[2\pi i\ \left(\frac{ku}{M} + \frac{\ell v}{N}\right)\right], \tag{2.27b}$$

waarin de waarden M en N de (eindige) afmetingen zijn van de bemonsterde funktie $f(k,\ell)$, en waarmee vergl. (2.26) overgaat in

$$G(u,v) = H(u,v)\ .\ F(u,v) + N(u,v), \tag{2.28}$$

waarin G, H, F en N resp. de Fourier-getransformeerden zijn van g, h, f en n, en u en v de diskrete spatiële frekwenties voorstellen.

Aan de hand van het model volgens vergl. (2.28) zullen we in het volgende hoofdstuk een aantal rekonstruktie-methodes bespreken.

3. REKONSTRUKTIE

3.1. Inleiding

In dit hoofdstuk zal een aantal frekwentiedomein-beeldrekonstruktie methoden worden besproken, die uitgaan van het in het voorgaande hoofdstuk afgeleide model van een vervormd beeld (vergl. 2.28):

$$G(u,v) = H(u,v)\ F(u,v) + N(u,v).$$

Getracht zal worden een "zo goed mogelijke" schatter te vinden voor $f(x,y)$ op basis van een te definiëren kwaliteits- of foutcriterium.

De eerste stap die daartoe gezet moet worden om tot een zinvolle rekonstruktie te kunnen overgaan, is na te gaan wat voor vervorming het beeld heeft ondergaan, de z.g. vervormingsidentifikatie. In sommige gevallen zijn er voldoende redenen aanwezig om aan te nemen dat het originele onvervormde beeld $f(x,y)$ geïsoleerde objektpunten bevat, welke direkt de PSF in het vervormde beeld representeren. Ook is het mogelijk de PSF a priori te bepalen op grond van kennis omtrent de eigenschappen van het beeldvormend systeem; bij fotografie kan b.v. de PSF van de lens en van de gebruikte film gemeten worden. Opgemerkt dient te worden dat sommige vervormingseffekten van de lens en van de film plaats-variant en niet-lineair zijn, en dan niet passen in het bovenstaande eenvoudige model.
In veel andere gevallen is het systeem dat de vervorming veroorzaakt niet voorhanden, zoals bij atmosferische turbulentie, bewegingsvervorming en defocussering. De PSF kan dan worden gemodelleerd met behulp van a priori kennis omtrent de beschrijving van een dergelijk fysisch proces. Bijvoorbeeld is het gebruikelijk (zie par. 2.2.1) om lineaire eenparige bewegingsonscherpte te modelleren m.b.v. een PSF met een constante amplitude over een zeker interval. In dit geval reduceert het vervormingsidentifikatieprobleem zich tot het meten van dit interval in het plaatsdomein, of tot het bepalen van de afstand tussen de nuldoorgangen van de overdrachtsfunktie (hier een Sinc-funktie) in het frekwentiedomein. Spectrale en cepstrale technieken om deze nuldoorgangen te bepalen worden besproken in [3,7,9]. Deze frekwentiedomeintechnieken hebben echter een aantal tekortkomingen [4]. Als voorbeeld hiervan het feit dat een gaussische PSF, die geen nuldoorgangen bezit in het frekwentiedomein, met deze technieken niet kan worden geïdentificeerd. Vandaar dat naast frekwentie-domeintechnieken ook identifikatietechnieken in het plaatsdomein zijn ontwikkeld. Hierbij wordt er vanuit gegaan dat een onvervormd beeld voldoende nauwkeurig kan worden gemodelleerd m.b.v. een autoregressief (AR) proces van lage orde [6,23]. Het verschuivings-invariante

vervormde beeld kan dan worden beschreven m.b.v. een autoregressief moving-average (ARMA) model. Het bepalen van de vervorming is dan op te vatten als een ARMA-identifikatieprocedure, waarbij de AR-coëfficiënten de modelcoëfficiënten vormen van het onderliggende AR-proces en de MA-coëfficiënten de vervormingsparameters [5]. In dit verslag wordt waar nodig gebruik gemaakt van de in [5] en [7] beschreven identifikatie-methoden.

De ruis wordt gemodelleerd als een realisatie van een stochastisch proces. Sommige rekonstruktiefilters maken gebruik van het vermogensdichtheidsspectrum van de ruis, terwijl andere rekonstruktiemethoden slechts de variantie van de ruis nodig hebben. In het geval van experimenteel verkregen beeld-data is het mogelijk om de variantie van de ruis te verkrijgen uit vlakke gebieden van het beeld, waar de afwijkingen van een gemiddelde waarde veroorzaakt worden door de ruis. Een schatting van het vermogensdichtheidsspectrum kan verkregen worden indien het vlakke gebied van redelijke afmetingen is (b.v. 64 x 64 pixels).

Het is tenslotte belangrijk om op te merken, dat zelfs met een maximum aan a priori kennis, vanwege het stochastische karakter van de ruis geen één-éénduidige oplossing van vergl. (2.28) mogelijk is. De verschillende rekonstruktiemethoden zoeken eerder een "zo goed mogelijke" schatting van $f(x,y)$, waarbij het criterium "zo goed mogelijk" slechts de a priori aannames weerspiegelt, die in de betreffende rekonstruktiemethoden zijn ingebouwd.

3.2. Het inverse filter

De meest eenvoudige rekonstruktiemethode wordt verkregen door de ruisterm in vergl. (2.28) te verwaarlozen, zodat de volgende multiplicatieve relatie wordt gevonden tussen het vervormde beeld $G(u,v)$ en het onvervormde beeld $F(u,v)$:

$$G(u,v) = H(u,v) \, . \, F(u,v) \, . \qquad (3.1)$$

Herschrijven van vergl. (3.1) geeft vervolgens

$$F(u,v) = \frac{G(u,v)}{H(u,v)} \, . \qquad (3.2)$$

Dit impliceert, dat indien $H(u,v)$ bekend is, we $f(x,y)$ kunnen rekonstrueren door in het Fourier-domein $G(u,v)$ te vermenigvuldigen met $1/H(u,v)$ en dan de inverse Fourier-transformatie toe te passen. Met andere woorden, de overdrachtsfunktie $L(u,v)$ van het inverse filter is:

$$L(u,v) = \frac{1}{H(u,v)} \, . \qquad (3.3)$$

Een aantal problemen rijst echter indien men praktisch gebruik wil maken van vergl. (3.2). Zo kunnen er punten of gebieden in het (u,v)-vlak zijn, waar $H(u,v) = 0$. Indien er geen ruis aanwezig is, dan bezit $G(u,v)$ ook nulpunten op deze frekwenties, hetgeen leidt tot een onbepaald systeem. Dus, zelfs in de afwezigheid van ruis is het in het algemeen onmogelijk $f(x,y)$ exact te reconstrueren indien $H(u,v)$ nulpunten in het (u,v)-vlak bezit. Als echter $H(u,v)$ een aftelbaar oneindig aantal nulpunten bezit, dan is $f(x,y)$ in principe exact rekonstrueerbaar [20].

In aanwezigheid van ruis zullen de nulpunten van $G(u,v)$ en $H(u,v)$ niet samenvallen. Daarom zal in de buurt van een nulpunt van $H(u,v)$ de deling in vergl. (3.2) tot zeer grote waarden aanleiding geven. In feite, als er additieve ruis aanwezig is, hebben we de volgende relatie (vergl. (2.28)):

$$G(u,v) = H(u,v) \; . \; F(u,v) + N(u,v),$$

zodat toepassen van het inverse filter resulteert in

$$\frac{G(u,v)}{H(u,v)} = F(u,v) + \frac{N(u,v)}{H(u,v)} \; . \tag{3.4}$$

In de buurt van nulpunten van $H(u,v)$ kan de term $N(u,v)/H(u,v)$ een veel grotere amplitude bezitten dan $F(u,v)$. Vooral omdat $H(u,v)$ in veel gevallen snel afvalt, terwijl $N(u,v)$ vaak een "wit" breedbandig karakter vertoont. De inverse transformatie van $G(u,v)/H(u,v)$ zal dan sterk beïnvloed worden door deze grote waarden, en zal niet langer op $f(x,y)$ lijken. In plaats hiervan zal het snelle ruisachtige variaties vertonen (hoogfrekwent ruis) en zal zeker niet leiden tot een betekenisvolle rekonstruktie van $f(x,y)$.

Het inverse filter staat de ruisterm in vergl. (2.28) dus niet toe en kan daarom slechts bruikbare resultaten geven, indien de signaal-ruisverhouding hoog is (> 60/70 dB). In de praktijk komen zulke hoge signaal-ruisverhoudingen zelden voor, zodat de meer overwogen methoden, die hieronder beschreven worden, betere resultaten zullen opleveren.

3.3. Het Wiener filter

Een manier om de "willekeur" in het inverse filter te vermijden is door de ruis expliciet in beschouwing te nemen, volgens vergl. (2.28), en door een rekonstruktiemethode (foutcriterium) te definiëren die de één of andere maat van het verschil tussen het origineel $f(x,y)$ en de schatting $\hat{f}(x,y)$ minimaliseert. Een maat die zeer intensief wordt gebruikt vanwege zijn mathematische eenvoud is de gemiddelde kwadratische fout:

$$\underset{\hat{f}}{\text{minimaliseer}} \quad E\{(f-\hat{f})^2\}. \tag{3.5}$$

De overeenkomstige definitie van het minimaliserings-probleem van vergl. (3.5) in het Fourier-domein luidt:

$$\underset{\hat{F}}{\text{minimaliseer}} \quad E\{|F-\hat{F}|^2\}, \tag{3.6}$$

waarin $\hat{F}$ en F resp. de Fourier-getransformeerden zijn van $\hat{f}$ en f. De resulterende schatter noemt men kleinste kwadraten schatter ("minimum mean square error"-estimator, MMSE-estimator). Deze schatters zijn in het algemeen niet-lineair. We beperken ons echter tot lineaire kleinste kwadraten schatters (LMMSE-estimators). Opgemerkt dient te worden, dat lineaire kleinste kwadraten schatters bij Gaussische processen leiden tot een absoluut minimum van de gemiddelde kwadratische schattingsfout (LMMSE = MMSE). In dit geval hebben niet-lineaire schatters gebaseerd op dit criterium geen voordeel.

We concentreren ons op een schatting van $\hat{F}$ (en dus van $\hat{f}$), die wordt verkregen door een lineaire operatie op het geregistreerde beeld,

$$\hat{F}(u,v) = L(u,v) \,.\, G(u,v). \tag{3.7}$$

Deze lineaire operatie is de overdrachtsfunktie van het Wiener filter, en wordt gegeven door (voor de afleiding zie bijlage A):

$$L(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \dfrac{S_n(u,v)}{S_f(u,v)}}, \tag{3.8}$$

waarin $S_n(u,v)$ het vermogensdichtheidsspectrum is van de ruisterm $n(k,\ell)$, $S_f(u,v)$ het vermogensdichtheidsspectrum van het onvervormde beeld $f(k,\ell)$ en * complex geconjugeerde aanduidt.

De afleiding van dit filter is gebaseerd op de volgende aannames [8]: (1) De functies $f(k,\ell)$ en $g(k,\ell)$ zijn realisaties van homogene stochastische velden (d.w.z., $f(k,\ell)$ en $g(k,\ell)$ zijn plaatsinvariant en de correlatiefunkties zijn translatie-invariant), (2) de funkties $f(k,\ell)$ en $n(k,\ell)$ zijn ongecorreleerd.

In veel gevallen is het onwaarschijnlijk dat aan al deze voorwaarden is voldaan, maar toch zal blijken dat het Wiener filter een aanzienlijke verbetering is t.o.v. het inverse filter.

De reden hiervoor kan worden gehaald uit de vorm van het Wiener filter in het (u,v)-vlak. Voor gebieden met een hoge signaal-ruisverhouding geldt, dat het Wiener filter het inverse filter benadert, of, als $S_n(u,v) \to 0$, dan

$$\hat{F}(u,v) \to \frac{G(u,v)}{H(u,v)} . \tag{3.9}$$

Gebieden met een lage signaal-ruisverhouding, ofwel waar $S_f(u,v) \to 0$, leiden tot

$$\hat{F}(u,v) \to 0. \tag{3.10}$$

Dit gedrag is niet onredelijk, omdat we niet kunnen verwachten enige informatie te verkrijgen uit spatiële frekwenties, waar de ruis volledig domineert.
Ook het afvallen naar nul van $H(u,v)$ zoals besproken bij het inverse filter zal de rekonstruktie niet kunnen bederven, omdat de noemer van vergl. (3.8) niet beneden de ondergrens kan komen, die gedefinieerd is als $S_n(u,v)/S_f(u,v)$. De funktie $H(u,v)$ kan zelfs nul worden terwijl er nog steeds een rekonstruktie gemaakt kan worden, ook in de aanwezigheid van ruis. In feite maakt het bestaan van ruis de rekonstruktie mogelijk wegens de aanwezigheid van de term $S_n(u,v)$ in de noemer van vergl. (3.8).

De LMMSE- (of Wiener-) schatter heeft aantrekkelijke eigenschappen en is reeds met succes toegepast in een aantal praktische beeldrekonstruktieproblemen. In het geval van lage signaal-ruisverhoudingen echter zijn de rekonstruktieresultaten visueel niet erg aantrekkelijk.
Dit kan de volgende oorzaken hebben:

(1) De LMMSE-schatting is gebaseerd op lineaire aannames, terwijl er nietlineariteiten aanwezig zijn in het registratieproces en in het menselijk visuele systeem dat de beelden evalueert.

(2) De LMMSE is niet het criterium dat het menselijk visuele systeem van nature toepast. LMMSE-rekonstrukties voor lage signaal-ruisverhoudingen blijken zwaar te effenen; het menselijk oog is echter vaak bereid om meer zichtbare ruis te accepteren, in ruil voor extra beeldstruktuur die anders verloren gaat tijdens het rekonstruktieproces.

(3) De LMMSE-schatting veronderstelt homogene stochastische velden, hetgeen waarschijnlijk onvoldoende is om een betekenisvolle struktuur in beelden te beschrijven, en gebruikt alleen de covariantie-informatie (in de vorm van de vermogensdichtheidsspectra $S_n(u,v)$ en $S_f(u,v)$) van het stationaire model bij de schatting.

Een praktisch nadeel van het Wiener filter is tevens dat het naast uitgebreide a priori kennis van de puntspreidingsfunktie h bovendien gedetailleerde informatie vereist omtrent de beeld- en ruis-vermogensdichtheidsspectra.

3.4. Geometrische gemiddelde filter

Een rekonstruktiefilter dat een geheel andere foutmaat hanteert werd voorgesteld door Cannon [9]. Het filter tracht het vermogensdichtheidsspectrum van het gerekonstrueerde beeld $\hat{f}$ gelijk te maken aan dat van het originele beeld f. Dit criterium kan worden geschreven als:

$$S_{\hat{f}}(u,v) = S_f(u,v). \qquad (3.11)$$

Indien we weer, net als bij het Wiener filter, een lineaire oplossing zoeken, dan is

$$\hat{F}(u,v) = L(u,v) \,.\, G(u,v), \qquad (3.12)$$

en geldt voor het vermogensdichtheidsspectrum van $\hat{F}(u,v)$

$$S_{\hat{f}}(u,v) = |L(u,v)|^2 \, S_g(u,v), \qquad (3.13)$$

waarin $S_g(u,v)$ het vermogensdichtheidsspectrum is van het te rekonstrueren vervormde beeld. Vullen we voor $S_g(u,v)$ het vermogensdichtheidsspectrum van ons vervormingsmodel in, dan leidt dit tot

$$S_{\hat{f}}(u,v) = |L(u,v)|^2 \, \{|H(u,v)|^2 \, S_f(u,v) + S_n(u,v)\}. \qquad (3.14)$$

Indien we nu forceren dat het spectrum $S_{\hat{f}}(u,v)$ gelijk is aan $S_f(u,v)$ (vergl. (3.11)), dan leidt dit tot het volgende rekonstruktiefilter:

$$|L(u,v)| = \left[\frac{S_f(u,v)}{|H(u,v)|^2 \, S_f(u,v) + S_n(u,v)}\right]^{\frac{1}{2}}. \qquad (3.15)$$

Dit filter wordt wel het "power-spectrum equalization" filter [1] of het homomorfe filter [3] genoemd. Met name het limietgedrag van dit filter is van belang. Indien $S_n(u,v) \approx 0$, dan geldt dat

$$|L(u,v)| \rightarrow \frac{1}{|H(u,v)|}\,, \tag{3.16}$$

d.w.z. het filter gaat over in het inverse filter (zonder fase-informatie). Het Wiener-filter in de vorige paragraaf vertoont eenzelfde gedrag. Evenzo als het spectrum $S_f(u,v)$ nul is, dan wordt $|L(u,v)| = 0$. Een kenmerkend verschil tussen het Wiener filter en dit filter volgt uit het geval wanneer $H(u,v) = 0$. Voor het Wiener filter (vergl. (3.8)) gold, dat dan de LMMSE-schatting nul was. In het geval van het homomorfe filter is echter

$$|L(u,v)| \rightarrow \left[\frac{S_f(u,v)}{S_n(u,v)}\right]^{\frac{1}{2}}. \tag{3.17}$$

Opgemerkt dient wel te worden dat het homomorfe filter slechts resulteert in een amplitude-filter, terwijl het Wiener-filter in staat is om ook fase-informatie te verwerken. Het is echter mogelijk het filter van vergl. (3.15) uit te breiden met een fase-term [9].

Indien we veronderstellen, dat $H(u,v)$ de nulfase bezit, ofwel dat de PSF $h(k,\ell)$ symmetrisch is, dan kunnen we de overdrachtsfunktie van het homomorfe filter eenvoudig omschrijven naar een andere interessante vorm:

$$L(u,v) = \left[\frac{1}{H(u,v)}\right]^{\frac{1}{2}} \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \dfrac{S_n(u,v)}{S_f(u,v)}}\right]^{\frac{1}{2}}. \tag{3.18}$$

Dit wordt het geometrisch gemiddelde filter genoemd, omdat deze beschouwd kan worden als het geometrisch gemiddelde tussen het inverse en het Wiener filter. Dit filter bezit dus een tussenvorm tussen het opscherpende maar ruisgevoelige karakter van het inverse filter en het teveel ruisuitsmerende karakter van het Wiener filter.

3.5. Gegeneraliseerd geometrisch gemiddelde filter

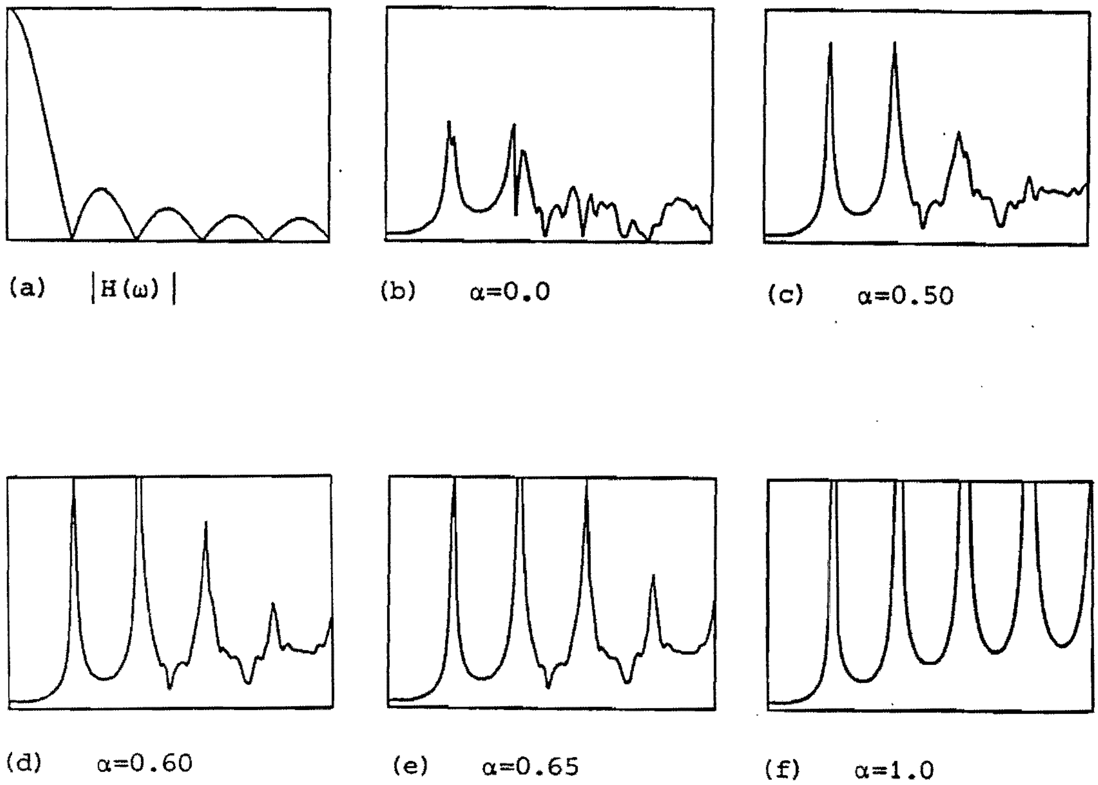
Digitale beeldrekonstruktie kan gezien worden als een afruil tussen winst in resolutie en afnemende gevoeligheid voor ruis. Een goede resolutie kan worden bereikt met het inverse filter, echter de ruisgevoeligheid van dit filter is veel te hoog gebleken. Anderzijds heeft het Wiener filter goede eigenschappen m.b.t. ruisreductie, maar introduceert als laag-doorlaat filter een verlies in resolutie. Stockham [21] stelde daarom voor om een filter te gebruiken, gebaseerd op vergl. (3.18) dat ligt tussen het inverse en het Wiener filter, het z.g. gegeneraliseerde geometrisch gemiddelde filter. Exacter geformuleerd, een filter dat het mogelijk maakt, om de afruil tussen de twee filters in te stellen m.b.v. twee parameters α en γ overeenkomstig het voorhanden zijnde probleem:

$$L(u,v) = \left[\frac{1}{H(u,v)}\right]^{\alpha} \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \gamma \dfrac{S_n(u,v)}{S_f(u,v)}}\right]^{1-\alpha} . \tag{3.19}$$

Opgemerkt kan worden, dat indien $\alpha=½$ en $\gamma=1$, dit leidt tot het geometrisch gemiddelde filter van de vorige paragraaf. Het geometrisch gemiddelde filter is dus één mogelijk compromis in het resolutie/ruis-probleem, net als het inverse filter ($\alpha=1$ óf $\gamma=0$) en het Wiener filter ($\alpha=0$ en $\gamma=1$).

Het zal duidelijk zijn, dat indien $H(u,v)$ nul wordt, het inverse gedeelte van het filter groot zal worden en de gebruiker dit effekt zal willen onderdrukken door α een waarde dicht bij nul te geven. Aan de andere kant, voor beelden met een hoge signaal-ruisverhouding en een $H(u,v)$ die niet-nul is, zal men kiezen voor $\alpha > ½$ en $\gamma = 1$, in welk geval het inverse filter zal domineren. Het is ook mogelijk het inverse filter te konstrueren met $\alpha=1$. Beelden met een lage signaal-ruisverhouding zullen gefilterd moeten worden met een lage α, om de invloed van het inverse filter op het resultaat klein te houden.

Ter illustratie zijn in figuur (3.1) enige filterresponsies geschetst van een één-dimensionaal gegeneraliseerd geometrisch gemiddelde filter voor verschillende waardes van α. Hieruit valt af te lezen, dat indien α groter wordt, dus het filter richting inverse filter gaat, de nulpunten van $H(u,v)$ steeds dominanter worden, en dus eventueel aanwezige ruis op die frekwenties opgeslingerd wordt.



Figuur (3.1). Filterresponsies van het 1-D gegeneraliseerd geometrisch gemiddelde filter.

3.6. Kleinste-kwadraten schatter met randvoorwaarden

Het Wiener filter zoals dat in par. 3.3 is beschreven staat bij rekonstruktie de aanwezigheid van beeldruis toe, maar is echter alleen dan succesvol te noemen indien de signaal-ruisverhouding niet al te klein is en het onvervormde beeld en de ruis als homogene stochastische velden te modelleren zijn met bekende vermogensdichtheidsspectra. Het is echter niet waarschijnlijk dat de aanname van homogeniteit voor het onvervormde beeld zondermeer geldig is en dat men het vermogensdichtheidsspectrum a priori kent. Vandaar dat een methode die niet zoveel a priori kennis omtrent het onvervormde beeld vereist mogelijkerwijs betere rekonstruktieresultaten op zal leveren. Eén zo'n methode is de z.g. kleinste kwadraten schatter met randvoorwaarden ("constrained least squares filter") die het eerst werd voorgesteld door Philips [17] en later is uitgebreid naar twee dimensies door Hunt [12].

Philips onderkende dat het bepalen van de numerieke inverse van integraalvergelijkingen van de vorm van vergl. (2.23) behoort tot de klasse van slecht gestelde problemen ("ill-posed problems"), en zodoende aanleiding zal geven tot wilde ruisopslingering, hoewel deze oscillaties onverenigbaar zijn met onze a priori kennis dat de oplossing een enigermate glad verloop dient te hebben. Philips verbond daarom aan de oplossing een numerieke maat voor gladheid, waarvoor hij op mathematische gronden de tweede afgeleide van de rekonstruktie $\hat{f}$ koos.

Als we hier voorlopig in het midden laten welke maat we kiezen, dan kan het criterium in het algemeen worden gesteld als: minimaliseer een zekere (lineaire) operatie c op de rekonstruktie $\hat{f}$, of

$$\underset{\hat{f}}{\text{minimaliseer}}\ E\{[c(\hat{f})]^2\}, \tag{3.20a}$$

met als randvoorwaarde

$$E\{[g-h \circledast \hat{f}]^2\} = E\{n^2\}. \tag{3.20b}$$

De fysische betekenis van deze vergelijkingen kan als volgt ingezien worden. c is een lineaire operator die, wanneer toegepast op $\hat{f}$, een maat is voor de hoeveelheid uitspreiding (gladheid). Het verschil $g-h \circledast \hat{f}$ is een maat voor de fout tussen de vervormde beeld-data g en de convolutie van de rekonstruktie $\hat{f}$ met de PSF h. De randvoorwaarde in vergl. (3.20b) is geïntroduceerd, omdat we verwachten dat de gemiddelde kwadratische fout gelijk zal zijn aan $E\{n^2\}$.
De kleinste-kwadraten methode met randvoorwaarden kan daarom beschouwd worden als het vinden van de gladste lineaire rekonstruktie die verenigbaar is met het vervormde beeld g, dat is, die voldoet aan de randvoorwaarde van vergl. (3.20b).
Het minimaliseringsprobleem luidt in het diskrete 2-D Fourier-domein:

$$\underset{\hat{F}(u,v)}{\text{minimaliseer}}\ E\{|C(u,v)\ \hat{F}(u,v)|^2\}, \tag{3.21a}$$

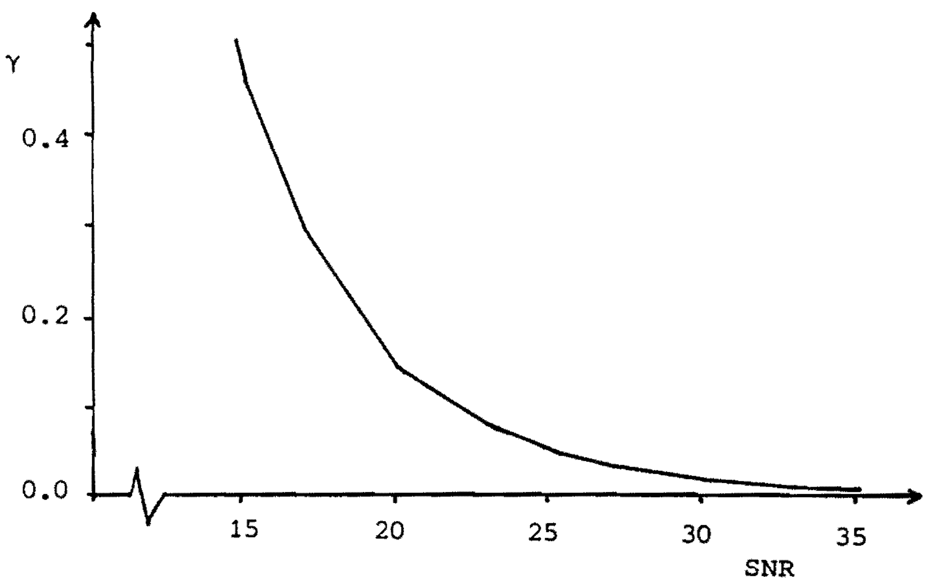
ten opzichte van

$$E\{|G(u,v) - H(u,v)\ \hat{F}(u,v)|^2\} = E\{|N(u,v)|^2\}. \tag{3.21b}$$

Deze vergelijkingen kunnen worden opgelost met behulp van de methode met de onbepaalde Lagrange-vermenigvuldigers (zie bijlage B(1)). De uitdrukking voor de overdrachtsfunktie van het filter die dan gevonden wordt is

$$L(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \gamma\ |C(u,v)|^2}\ , \tag{3.22}$$

waarin $\lambda = 1/\gamma$ een Lagrange-vermenigvuldiger is, en $C(u,v)$ de Fourier-getransformeerde van de lineaire operator c.
De parameter γ moet zodanig worden ingesteld, dat aan de randvoorwaarde van vergl. (3.21b) wordt voldaan, hetgeen vaak op iteratieve wijze wordt gedaan (zie bijlage B(2)) [12]. γ is dus de parameter waarin de informatie omtrent de ruis zit opgesloten, en is dan ook omgekeerd evenredig met de variantie van de ruis; dit is geschetst in figuur (3.2).



Figuur (3.2). γ als funktie van de ruis-variantie, bij iteratieve bepaling van γ, (voor defocussering met R^2=19 en met de tweede afgeleide als gladheidsmaat).

De algemeenheid van de lineaire operator C waarmee we dit filter hebben afgeleid, stelt ons nu in staat om een verscheidenheid aan filters te ontwikkelen, met alle dezelfde struktuur als in vergl. (3.22). Bijvoorbeeld, indien

$$C(u,v) = 1, \tag{3.23}$$

dan wordt een minimum-norm oplossing voor $\hat{f}$ gezocht, hetgeen resulteert in

$$L(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \gamma}\ . \tag{3.24}$$

In de limiet als $\gamma \to 0$, wordt het resulterende filter gedefinieerd als het pseudo-inverse filter [2], genoteerd als

$$[H(u,v)]^+ = \lim_{\gamma \to 0} \left\{ \frac{H^*(u,v)}{|H(u,v)|^2 + \gamma} \right\} , \tag{3.25}$$

waarin de notatie $[\ ]^+$ verwijst naar de inverse van $H(u,v)$, of de pseudo-inverse indien de inverse niet bestaat. Indien we kiezen

$$C(u,v) = \left[\frac{S_n(u,v)}{S_f(u,v)} \right]^{\frac{1}{2}} , \tag{3.26}$$

dan leidt dit tot het z.g. "parametrische Wiener filter",

$$L(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \gamma \dfrac{S_n(u,v)}{S_f(u,v)}} , \tag{3.27}$$

terwijl met $\gamma=1$ het traditionele Wiener filter wordt verkregen. M.b.v. de parameter γ is het mogelijk om enigermate de signaalstatistiek ($\gamma < 1$) danwel de ruisstatistiek ($\gamma > 1$) te benadrukken.

Tenslotte zullen hieronder nog twee belangrijke keuzes voor de operator $C(u,v)$ worden besproken, te weten de Laplace operator (tweede afgeleide) en een visueel criterium, dat tracht de eigenschappen van het menselijk visuele systeem in rekening te brengen.

3.6.1. Het tweede-afgeleide criterium

De tweede afgeleide van het beeld $f(x,y)$, ook wel Laplace operator genoemd, wordt gedefinieerd als

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} . \tag{3.28}$$

In het diskrete geval worden de verschillen tussen helderheidswaarden gehanteerd in plaats van de werkelijke afgeleiden. Voor de eerste afgeleide in de x- en y-richtingen geldt dan

$$\Delta_x f(k,\ell) = f(k,\ell) - f(k-1,\ell),$$
$$\Delta_y f(k,\ell) = f(k,\ell) - f(k,\ell-1). \tag{3.29}$$

Voor de tweede afgeleide wordt genomen:

$$\Delta_x^{\ 2} = \Delta_x f(k+1,\ell) - \Delta_x f(k,\ell)$$
$$= f(k-1,\ell) - 2\ f(k,\ell) + f(k+1,\ell), \tag{3.30}$$
$$\Delta_y^{\ 2} = f(k,\ell-1) -2\ f(k,\ell) + f(k,\ell+1).$$

De diskrete Laplace operator wordt hiermee

$$\nabla^2 f(k,\ell) = f(k-1,\ell) + f(k+1,\ell) + f(k,\ell-1) + f(k,\ell+1) - 4f(k,\ell) \tag{3.31}$$

Dit betekent, dat het toepassen van de diskrete Laplace operator op een beeld neerkomt op de convolutie met de puntspreidingsmatrix [11]

$$c(k,\ell) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \tag{3.32}$$

waarmee de diskrete PSF eenvoudig gekonstrueerd wordt door deze matrix aan te vullen met nullen, waarbij de factor -4 op het nulpunt van het (k,ℓ)-vlak valt. Aangetoont kan worden, dat [8]

$$C(u,v) = 2\{[\cos(\frac{2\pi u}{M})-1] + [\cos(\frac{2\pi v}{N})-1]\}, \tag{3.33}$$

waarin u en v gehele getallen zijn welke de afstand van het nulpunt in het (u,v)-vlak representeren, met $|u| < M/2$ en $|v| < N/2$. Indien een cirkelsymmetrische vorm gewenst is, en indien $M=N$ dan volgt [12]

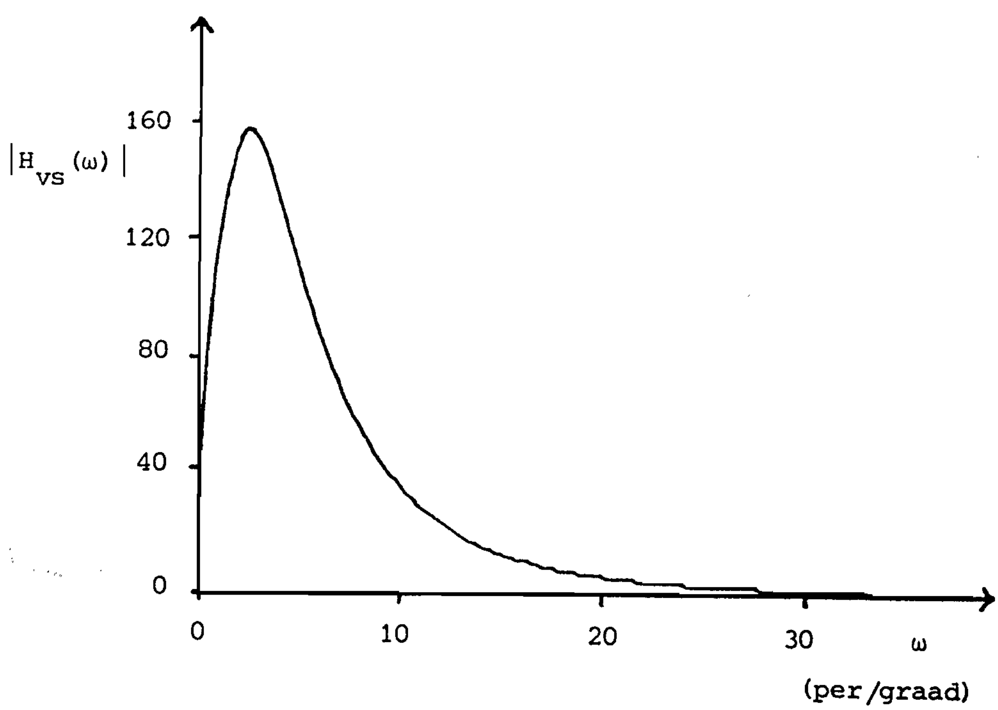
$$C(u,v) = 2\{\cos(\frac{2\pi\sqrt{u^2+v^2}}{N})-1\} \qquad m^2+n^2 \leq \frac{N^2}{4}$$
$$= -4 \qquad m^2+n^2 \geq \frac{N^2}{4}. \tag{3.34}$$

3.6.2. Visueel criterium

Het bepalen van de responsie van het menselijk visuele systeem is bijzonder moeilijk, vanwege de ontoegankelijkheid en de verdeeldheid van het systeem. Het is niet mogelijk om een component van dit systeem te isoleren en z'n eigenschappen te meten. In plaats daarvan moet het gedrag afgeleid worden op basis van indirekte metingen.

Experimenten tonen aan, dat het menselijk visuele systeem de logaritme waarneemt van in het oog invallende licht-intensiteiten. Dit resultaat mag verwonderlijk lijken, maar is overeenkomstig met wat het oog kan waarnemen aan dynamisch bereik van licht-intensiteiten. Visuele waarnemingen vinden plaats op belichtingsniveaus van een maanloze nacht tot volle zon op een sneeuwveld, een intensiteitsbereik van acht ordes van grootte.

Het menselijk visuele systeem werkt tevens als een spatieel frekwentiefilter. Geen mens kan zien met een oneindige resolutie, dus moet er een limiet zijn aan de maximale spatiële frekwentie die het oog kan waarnemen op een bepaalde kijk-afstand. In dit opzicht moet het oog zich gedragen als een laag-doorlaat filter, met geen resolutie boven een zekere frekwentie-bovengrens. Verrassender is de frekwentie-responsie in het lage- en midden-frekwentiegebied. Het visuele systeem weegt de middenfrekwenties zwaarder dan de lage en de hoge frekwenties. Een 1-D afbeelding van de spatiële overdrachtsfunktie van het menselijk visuele systeem als funktie van de frekwentie is gegeven in figuur (3.3).



Figuur (3.3). Spatiële modulatie-overdrachtsfunktie van het menselijk visuele systeem.

Er zijn tot nu toe een tweetal zogenaamde psychofysische metingen voorgesteld om de spatiële modulatie-overdrachtsfunktie te bepalen: de zogenaamde "drempel-meting" [19] en een door Davidson [10] ontworpen methode van "contrast-vergelijking".
Bij drempelmeting bijvoorbeeld krijgt een proefpersoon een sinusvormig raster met een bepaalde spatiële frekwentie als stimulus aangeboden. Het is dan de bedoeling dat de proefpersoon het contrast (modulatie-diepte) zodanig afregelt, dat het raster nog net gezien wordt, of dat het raster nog net van een uniform veld te onderscheiden is. Indien deze meting voor verschillende spatiële frekwenties herhaald wordt, wordt de in figuur (3.3) afgebeelde kromme gevonden [19].

Indien er vanuit gegaan wordt, dat het visuele systeem een lineair en homogeen systeem is, dan hangt de twee-dimensionale impulsresponsie $h_{vs}(x,y)$ slechts af van de afstand tussen twee punten in het (x,y)-vlak. Nemen we als afstandsfunktie de z.g. euclidische afstand tussen twee punten

$$d_E\{(x,y),(p,q)\} = \sqrt{(x-p)^2 + (y-q)^2} = r, \tag{3.35}$$

en voor het punt (p,q) de oorsprong, dan kan de twee-dimensionale impulsresponsie $h_{vs}(x,y)$ die nu cirkelsymmetrisch is, als volgt worden geschreven:

$$h_{vs}(x,y) = h_{vs}(\sqrt{x^2+y^2}) = h_{vs}(r). \tag{3.36}$$

In vergl. (3.36) stelt $h_{vs}(r)$ de één-dimensionale impulsresponsie voor, genomen langs een rechte lijn met een willekeurige oriëntatie. Het kan nu bewezen worden, dat de Fourier-getransformeerde $H_{vs}(\omega_1,\omega_2)$ van $h_{vs}(x,y)$ tevens cirkelsymmetrisch is [19]:

$$H_{vs}(\omega_1,\omega_2) = H_{vs}(\sqrt{\omega_1^2+\omega_2^2}) = H_{vs}(\omega). \tag{3.37}$$

$H_{vs}(\omega)$ is de één-dimensionale overdrachtsfunktie van het menselijk visuele systeem, zoals die in figuur (3.3) is afgebeeld. De twee-dimensionale spatiële modulatie-overdrachtsfunktie wordt dus verkregen door "omcirkelen" van de één-dimensionale overdrachtsfunktie $H_{vs}(\omega)$.

De overdrachtskarakteristiek $H_{vs}(\omega_1,\omega_2)$ heeft bepaalde effekten m.b.t. de waarneming van beelden door het visuele systeem. Neem aan, dat de in het oog binnenvallende intensiteiten afkomstig zijn van een beeld dat gecorreleerd is,

d.w.z. de covariantie-funktie van het beeld is geen Dirac-funktie. Het beeld dat door de hersenen wordt waargenomen zal nu niet dezelfde correlatie bezitten. Het lineaire filter van de oog-responsie zal nieuwe correlatie in het beeld introduceren terwijl het vermogensdichtheidsspectrum van het waargenomen beeld zal zijn vermenigvuldigd met $|H_{VS}(\omega_1,\omega_2)|^2$.
Omdat het moeilijk is om het vermogensdichtheidsspectrum S_f van het originele beeld als a priori kennis te verkrijgen (één van de nadelen van het Wiener filter), is het gebruik ervan minder gewenst en doen we net als of het beeld f een wit proces representeert, m.a.w. de covariantiefunktie is een Dirac-funktie. In dat geval is de enige correlatie die door de hersenen wordt waargenomen de correlatie die door het menselijk visuele systeem wordt geïntroduceerd. Deze aanname staat bekend als het "maximum ignorance principle".

Een filter dat gebaseerd is op dit principe heeft dan de volgende overdrachtsfunktie:

$$L(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \dfrac{\gamma}{|H_{VS}(u,v)|^2}}, \qquad (3.38)$$

waarin H(u,v) de Fourier-transformatie van de PSF h is en $H_{VS}(u,v)$ de 2-D overdrachtsfunktie is van het menselijk visuele systeem.
Dit filter besteed dus de meeste aandacht aan die frekwentiecomponenten waarvoor het menselijk visuele systeem het meest gevoelig is.

4. RANDWAARDEN

4.1. Inleiding

In het vorige hoofdstuk hebben we met behulp van het in hoofdstuk 2 opgestelde vervormingsmodel enkele beeldrekonstruktiemethodes (filters) besproken. Willen we echter daadwerkelijk overgaan tot filteren, dan zullen we rekening moeten houden met de eindige beeldafmetingen.

Beschouwen we het vervormingsmodel van vergl. (2.26),

$$g(k,\ell) = \sum_{m=-m_1}^{m_2} \sum_{n=-n_1}^{n_2} h(m,n)\ f(k-m,\ell-n) + n(k,\ell),$$

dan zien we dat alvorens de ruis wordt toegevoegd, een objektpunt vervormd wordt door het een waarde te geven die een gewogen som is van zichzelf (m=0, n=0) en zijn omgeving (support). De support van de PSF veronderstellen we eindig en in principe zelfs klein t.o.v. de afmetingen van het beeld.

Voor de rekonstruktie van een objektpunt geldt dus, dat hiervoor een aantal omliggende beeldpunten nodig zijn. Dit zal echter problemen opleveren wanneer we punten die dicht bij de beeldrand liggen willen rekonstrueren, aangezien we daar niet meer de beschikking hebben over alle voor rekonstruktie benodigde punten.

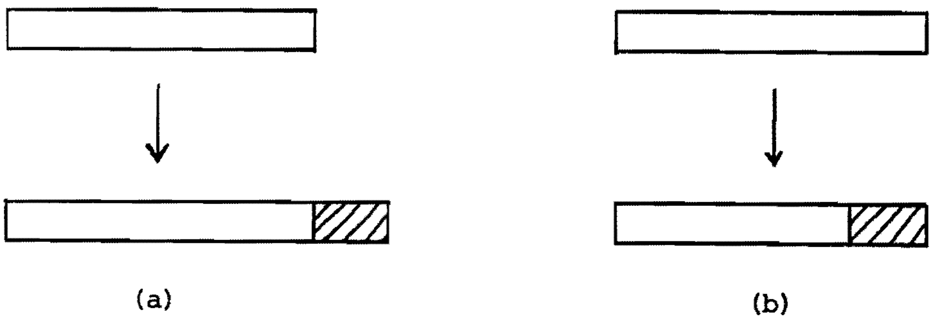
Gebruiken we voor beeldrekonstruktie in het plaatsdomein een deconvolutiefilter met een eindige impulsresponsie (FIR), dan zullen zich randeffekten voordoen tot die afstand van de beeldrand tot waar de support van het FIR-filter zich uitstrekt; daarbinnen zijn alle voor dit filter benodigde beeldpunten beschikbaar en doen zich geen randeffekten voor. Wordt echter als deconvolutiefilter gebruik gemaakt van het Kalman filter, hetgeen een voorbeeld is van een filter met een oneindige impulsresponsie (IIR), dan zullen rekonstruktiefouten t.g.v. de rand van het beeld zich over het gehele beeld manifesteren als z.g. "ringing". Het is dus met name bij IIR-filters van belang dat goede aannames worden gedaan voor de waarden van die punten die niet binnen de grenzen van het beeld vallen, maar wel voor de rekonstruktie bekend moeten zijn.

Een mogelijke realisering van de randwaarden is om deze de waarde van het gemiddelde van het beeld te geven. In het geval van het Kalman filter zal dit echter toch aanleiding geven tot ringing, hetgeen veroorzaakt wordt door de abrupte overgang tussen beeld en randwaarden.

Gemotiveerd door methodes ter oplossing van partiële differentiaalvergelijkingen gebruiken Woods e.a. [24] voor het Kalman filter zogenaamde "nulde-afgeleide"-randwaarden. Dit houdt in dat voor de randwaarden dezelfde waarden worden genomen als de eerste rij of kolom van het beeld, om zodoende de abrupte overgang tussen beeld en randwaarden te vermijden.

In [24] gebruiken Biemond e.a. als rekonstruktiemethode een Kalman filter dat een beeld lijn voor lijn rekonstrueert. Door gebruik te maken van een geschikte decorrelerende rij-transformatie leiden zij een hybride filterschema af, bestaande uit een parallelbank van 1-D Kalman filters die de in de richting van de rijen getransformeerde beelddata in de kolomrichting recursief filteren. Als decorrelerende transformatie is de 1-D diskrete Fourier-transformatie gekozen, die geïmplementeerd is m.b.v. de FFT ("Fast Fourier Transform"). Dit betekent dat impliciet uitgegaan wordt van een zich periodiek herhalende beeldlijn, waarbij het meest linkse beeldlijnpunt aansluit bij het meest rechtse punt van de beeldlijn.
Indien er een aanzienlijk verschil bestaat tussen de waarden van deze punten, zal dit leiden tot een sprong, die in het Fourierdomein aanleiding geeft tot frekwentiecomponenten die in de oorspronkelijke beeldlijn niet aanwezig waren, de zogenaamde lekfrekwenties ("leakage"). Lekfrekwenties die liggen in de buurt van de nulpunten van de overdrachtsfunktie van de vervorming (bewegingsonscherpte, defocussering) zullen versterktworden door de deconvoluerende werking van het Kalman filter, en na terugtransformatie als artefacten duidelijk in het beeld zichbaar zijn. Voor bijvoorbeeld horizontale bewegingsonscherpte ontstaat een vertikaal lijnenpatroon waarbij de lijnen een onderlinge afstand hebben die gelijk is aan de lengte van de vervorming, d.w.z. de herhalingsfrekwenties van deze lijnen komen overeen met de frekwenties behorende bij de nulpunten van de overdrachtsfunktie voor bewegingsonscherpte.
Ter reductie van deze ringing wordt in [24] voorgesteld, om elke beeldlijn te verlengen met een aantal punten, die zodanige waarden krijgen dat linker- en rechterzijde van elke beeldlijn goed op elkaar aansluiten. Geconcludeerd wordt dat lineaire interpolatie en interpolatie m.b.v. een derde-orde polynoom vrijwel alle ringing elimineert.

Een praktisch nadeel om een beeldlijn uit te breiden is echter, dat voor de FFT van een beeldlijn een aantal punten vereist is, dat exact een macht van twee is, (of het produkt van machten van priemgetallen voor meer gecompliceerde FFT-algoritmes). Dit legt altijd een beperking op voor wat betreft het aantal randwaarden.
Om hieraan te ontkomen zullen in dit verslag i.p.v. de randwaarden aan een beeldlijn toe te voegen, deze over de beeldpunten aan de rand heengelegd worden, zodat de lengte van de beeldlijn na voorzien te zijn van randwaarden gelijk zal blijven; dit is geschetst in figuur (4.1).



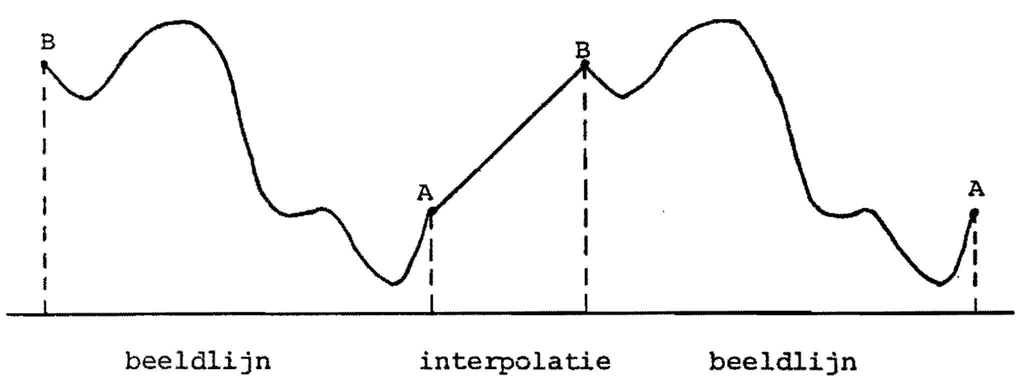
Figuur (4.1). een beeldlijn voorzien van randwaarden door
(a) deze als extra punten aan de beeldlijn toe te voegen, of
(b) deze over bestaande beeldlijnpunten heen te leggen.

De consequentie is wel, dat beelddata weggegooid wordt, maar omdat het aantal randwaarden meestal beperkt zal blijven tot 10 à 15, met een gangbare beeldlijnlengte van 256 beeldpunten of groter, is dit niet van groot belang; bovendien staat verreweg de meeste informatie in het midden van het beeld.

In paragraaf 4.2 resp. 4.3 zal een één-dimensionale vector-matrix formulering worden gegeven van lineaire interpolatie en interpolatie m.b.v. een derde-orde polynoom, waarna tenslotte in paragraaf 4.4 de interpolatie zal worden uitgebreid naar twee dimensies.

4.2. Lineaire interpolatie

Een geleidelijke overgang tussen twee tegenover elkaar liggende beeldpunten kan worden verkregen door lineair te interpoleren tussen de waarden van deze punten. Beschouw daartoe de één-dimensionale situatie, zoals die is afgebeeld in figuur (4.2).



Figuur (4.2). Lineaire interpolatie.

De punten met de waarden A en B zijn de twee tegenover elkaar liggende punten waartussen geïnterpoleerd wordt. Als het aantal punten dat geïnterpoleerd moet worden ℓ is, dan kunnen de interpolaties f geschreven worden als

$$f(k) = a_0 + a_1 k \qquad 1 \leq k \leq \ell, \tag{4.1a}$$

waarin a_0 en a_1 berekend worden uit de randvoorwaarden

$$\begin{aligned} f(0) &= A \\ f(\ell+1) &= B. \end{aligned} \tag{4.1b}$$

De oplossing hiervan wordt gevonden door vergl. (4.1a) en (4.1b) om te schrijven naar vector-matrixvergelijkingen:

$$[f(1) \; . \; . \; . \; . \; . \; f(\ell)] = [a_0 \quad a_1] \begin{bmatrix} 1 & 1 & . & . & . & . & . & 1 \\ 1 & 2 & . & . & . & . & . & \ell \end{bmatrix} \tag{4.2a}$$

$$[A \quad B] = [a_0 \quad a_1] \begin{bmatrix} 1 & 1 \\ 0 & \ell+1 \end{bmatrix}. \tag{4.2b}$$

Deze vergelijkingen kunnen verkort worden genoteerd als

$$\underline{f} = \underline{a}_2 \cdot K_2 \tag{4.3a}$$
$$\underline{r}_2 = \underline{a}_2 \cdot M_2, \tag{4.3b}$$

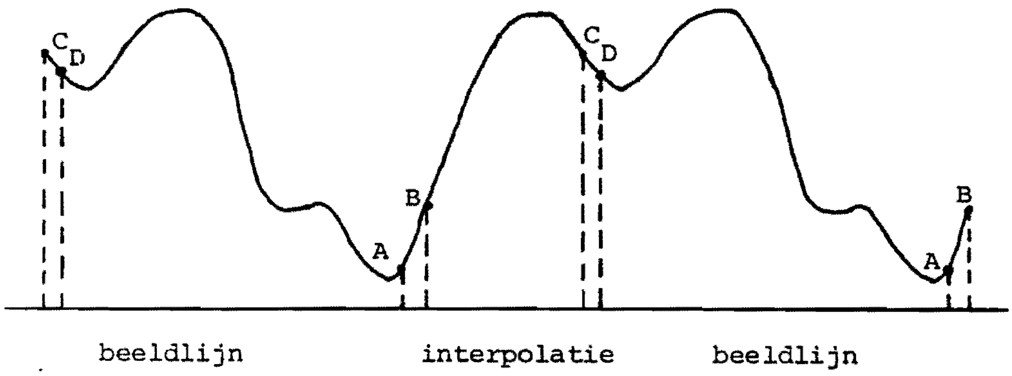
waaruit de vector $\underline{a}_2$ kan worden geëlimineerd:

$$\underline{f} = \underline{r}_2 \cdot M_2^{-1} \cdot K_2. \tag{4.4}$$

M.b.v. deze vergelijking kan dus lineair geïnterpoleerd worden tussen twee randwaarden, gerepresenteerd door de vector $\underline{r}_2$.

4.3. Interpolatie met een derde-orde polynoom

Bij lineaire interpolatie wordt slechts gebruik gemaakt van de buitenste punten van een beeld. We willen nu echter een nog geleidelijkere overgang tussen de tegenover elkaar liggende beeldzijden realiseren, door behalve de waarde aan de rand van een beeld ook de richting mede in beschouwing te nemen. We zullen dit doen d.m.v. interpolatie met een derde-orde polynoom. De één-dimensionale situatie is afgebeeld in figuur (4.3), waaruit valt af te lezen, dat ook de richting aan de randen van de beeldlijn bepalend is voor de interpolatie.



Figuur (4.3). Interpolatie met een derde-orde polynoom.

De punten met waarden A en B zijn de laatste twee punten aan de rechterkant van de beeldlijn, de punten met waarden C en D aan de linkerkant. Als ℓ het aantal punten is dat ligt tussen B en C, zijn de waarden van de interpolaties

$$f(k) = a_0 + a_1k + a_2k^2 + a_3k^3 \qquad 1 \leq k \leq \ell, \tag{4.5a}$$

waarin de parameters a_0, a_1, a_2 en a_3 berekend kunnen worden uit de randvoorwaarden

$$\begin{aligned} f(-1) &= A \\ f(0) &= B \\ f(\ell+1) &= C \\ f(\ell+2) &= D. \end{aligned} \tag{4.5b}$$

Net als bij de lineaire interpolatie wordt de oplossing gevonden door vergl. (4.5a) en (4.5b) om te schrijven tot

$$[f(1)\ \ldots\ldots\ f(\ell)] = [a_0\ \ a_1\ \ a_2\ \ a_3] \begin{bmatrix} 1 & 1 & \ldots\ldots & 1 \\ 1 & 2 & \ldots\ldots & \ell \\ 1 & 4 & \ldots\ldots & \ell^2 \\ 1 & 8 & \ldots\ldots & \ell^3 \end{bmatrix} \qquad (4.6a)$$

$$[A \quad B \quad C \quad D] = [a_0\ \ a_1\ \ a_2\ \ a_3] \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 0 & \ell+1 & \ell+2 \\ 1 & 0 & (\ell+1)^2 & (\ell+2)^2 \\ -1 & 0 & (\ell+1)^3 & (\ell+2)^3 \end{bmatrix} \qquad (4.6b)$$

en kan verkort genoteerd worden als

$$\underline{f} = \underline{a}_4 \cdot K_4 \qquad (4.7a)$$

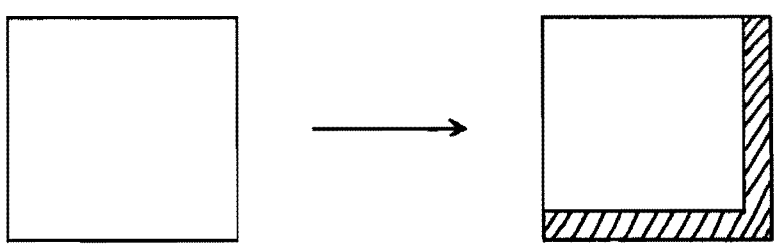
$$\underline{r}_4 = \underline{a}_4 \cdot M_4,$$

zodat door eliminatie van de vector $\underline{a}_4$ de oplossing verkregen wordt:

$$\underline{f} = \underline{r}_4 \cdot M_4^{-1} \cdot K_4. \qquad (4.8)$$

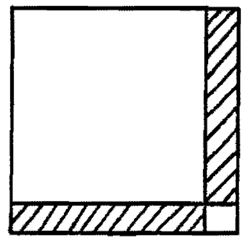
4.4. Twee-dimensionale randwaarden

In figuur (4.4) is afgebeeld hoe een beeld twee-dimensionaal voorzien moet worden van randwaarden, waarbij de afmetingen van het beeld omwille van de FFT gelijk zijn gebleven; het te filteren beeld is dus in feite iets kleiner geworden.



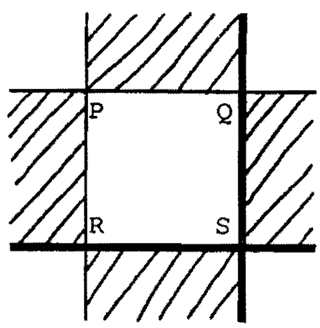
Figuur (4.4). Een beeld voorzien van randwaarden onder gelijkblijvende afmetingen.

Met behulp van vergl. (4.4) of vergl. (4.8) is het mogelijk, om door het beeld in horizontale en vertikale richting herhaald te denken, elke beeldrij en -kolom te voorzien van randwaarden. Daarna rest nog het bepalen van de randwaarden in het niet gearceerde vierkantje in figuur (4.5). Het zal blijken dat het niet uitmaakt of we hier horizontaal of vertikaal interpoleren.



Figuur (4.5). Rand na interpolatie voor alle beeldrijen en -kolommen (gearceerde gebied).

Om dit in te zien is de situatie van figuur (4.5) periodiek en vergroot afgebeeld in figuur (4.6).



Figuur (4.6). Periodieke en vergrootte situatie van figuur (4.5).

De vier hoekpunten van het nog te interpoleren gebied zijn aangegeven met hun waarden P, Q, R en S. De lijnen PQ, RS, PR en QS zijn reeds zelf interpolaties. Volgens vergl. (4.4) of vergl. (4.8) kan een horizontale interpolatie worden gerepresenteerd door

$$\underline{f}_H = [f_H(1) \ldots\ldots f_H(\ell_H)] = \underline{r}_H \cdot M^{-1} \cdot K_H, \tag{4.9}$$

en een vertikale interpolatie door de getransponeerde ervan:

$$\underline{f}_V^t = \begin{bmatrix} f_V(1) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ f_V(\ell_V) \end{bmatrix} = K_V^t \cdot (M^{-1})^t \cdot \underline{r}_V^t. \qquad (4.10)$$

We zullen nu alle waarden van het resterende gebied PQRS uitrekenen voor het geval dat we resp. horizontaal of vertikaal interpoleren, waarna zal blijken dat deze gelijk zijn. We zullen dit slechts doen voor lineaire interpolatie; voor interpolatie met een derde-orde polynoom gaat dit analoog.

Voor horizontale interpolatie hebben we de randvoorwaarden nodig waartussen we moeten interpoleren: deze staan op de lijnen PR en QS, welke geschreven kunnen worden als de vertikale interpolaties tussen resp. P en R, en Q en S:

$$\text{PR: } \underline{f}_{PR}^t = K_V^t \cdot (M^{-1})^t \cdot \begin{bmatrix} P \\ R \end{bmatrix}, \qquad (4.11a)$$

$$\text{QS: } \underline{f}_{QS}^t = K_V^t \cdot (M^{-1})^t \cdot \begin{bmatrix} Q \\ S \end{bmatrix}. \qquad (4.11b)$$

Indien we deze als randvoorwaarden van een horizontale interpolatie gebruiken, vinden we de uitdrukking voor alle waarden van het gebied PQRS, gerepresenteerd door de matrix F_H:

$$F_H = \begin{bmatrix} \underline{f}_{PR}^t & \underline{f}_{QS}^t \end{bmatrix} \cdot M^{-1} \cdot K_H \Rightarrow$$

$$F_H = K_V^t \cdot (M^{-1})^t \cdot \begin{bmatrix} P & Q \\ R & S \end{bmatrix} \cdot M^{-1} \cdot K_H. \qquad (4.12)$$

De berekening van de waarden van het gebied PQRS indien we vertikaal interpoleren gaat analoog. De randvoorwaarden voor vertikale interpolatie staan op de horizontale lijnen PQ en RS, welke zelf geschreven kunnen worden als horizontale interpolatie tussen resp. P en Q, en R en S:

$$\text{PQ: } \underline{f}_{PQ} = [P \quad Q] \cdot M^{-1} \cdot K_H, \qquad (4.13a)$$

$$\text{RS: } \underline{f}_{RS} = [R \quad S] \cdot M^{-1} \cdot K_H. \qquad (4.13b)$$

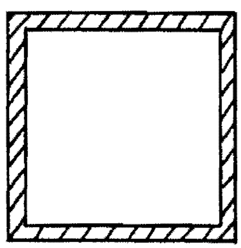
Deze vullen we als randvoorwaarden voor een vertikale interpolatie in, zodat we de matrix F_V krijgen:

$$F_V = K_V^t \cdot (M^{-1})^t \cdot \begin{bmatrix} \underline{f}_{PQ} \\ \underline{f}_{RS} \end{bmatrix} \Rightarrow$$

$$F_V = K_V^t \cdot (M^{-1})^t \cdot \begin{bmatrix} P & Q \\ R & S \end{bmatrix} \cdot M^{-1} \cdot K_H. \qquad (4.14)$$

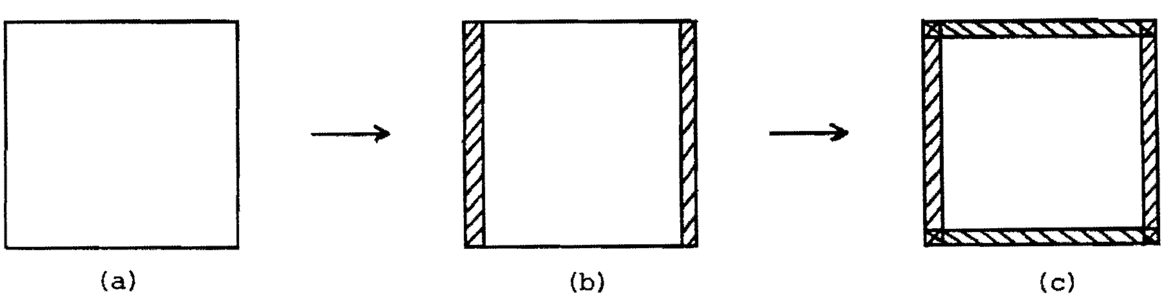
Vergelijking van vergl. (4.12) en (4.14) leert, dat deze gelijk zijn, zodat het niet uitmaakt of we horizontaal of vertikaal interpoleren.

Omdat de meeste beeldinformatie zoals gezegd in het midden van het beeld zal staan, is het logischer om de randwaarden gelijk over resp. de linker- en rechterkant, en de boven- en onderkant van het beeld te verdelen, hetgeen toegestaan is vanwege de door de FFT aangenomen periodiciteit; dit is afgebeeld in figuur (4.7).



Figuur (4.7). Gelijkmatige verdeling randwaarden rondom het beeld.

Een algoritme tenslotte om een beeld van randwaarden te voorzien zou er dan uitzien als afgebeeld in figuur (4.8): interpoleer eerst horizontaal voor alle beeldrijen ((a) → (b)) en interpoleer vervolgens vertikaal voor alle beeldkolommen inclusief de reeds gevonden randwaarden ((b) → (c)).

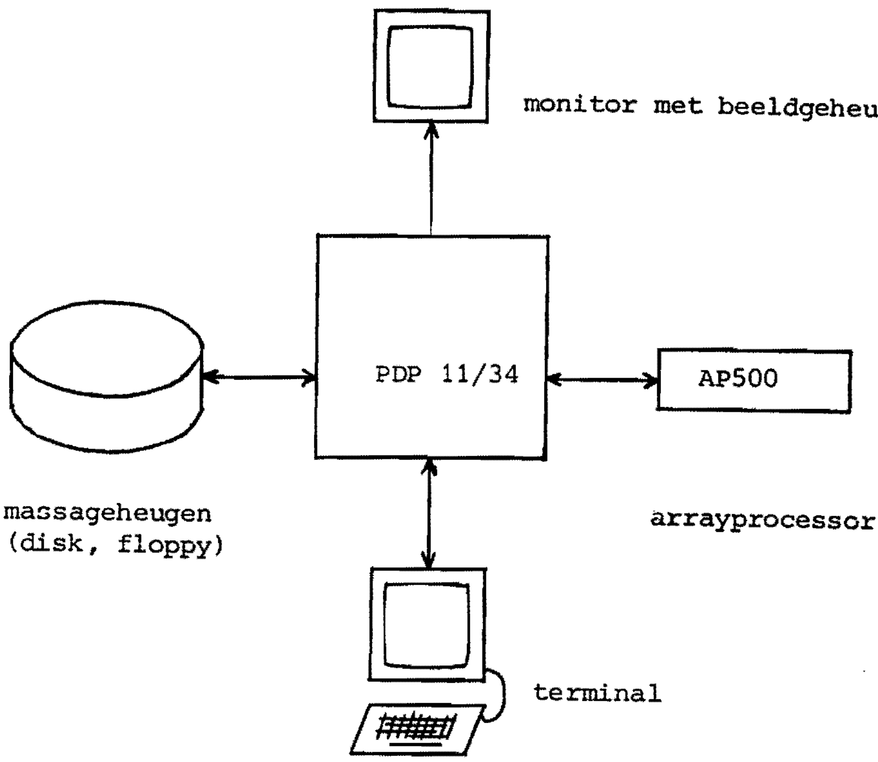


Figuur (4.8). Interpolatie-algoritme.

5. IMPLEMENTATIE

5.1. Inleiding

De in hoofdstuk 3 beschreven rekonstruktiefilters, met de randwaarden zoals besproken in hoofdstuk 4 en overige ondersteunende programmatuur zijn geïmplementeerd op een computer-systeem, waarvan de configuratie in figuur (5.1) schematisch is weergegeven.



Figuur (5.1). Computer-configuratie.

De configuratie bestaat uit een PDP 11/34 "general purpose" computer met massageheugen en terminal, waaraan zijn toegevoegd een arrayprocessor met de PDP 11/34 als "host" en een monitor met beeldgeheugen.
In het beeldgeheugen kunnen via het geheugen van de PDP 11/34 beelden gezet worden die op de monitor getoond worden, zodat rekonstruktieresultaten direkt zichtbaar kunnen worden gemaakt. De AP500 arrayprocessor maakt het mogelijk om grote hoeveelheden data zeer snel te bewerken en is dus bij uitstek geschikt voor beeldrekonstruktie. Het blijkt dat filtertijden van ongeveer 20 minuten per beeld kunnen worden teruggebracht tot 1 à 2 sekonden. In dit hoofdstuk zal daarom aandacht worden besteed aan de AP500-arrayprocessor en het

programmeren hiervan.

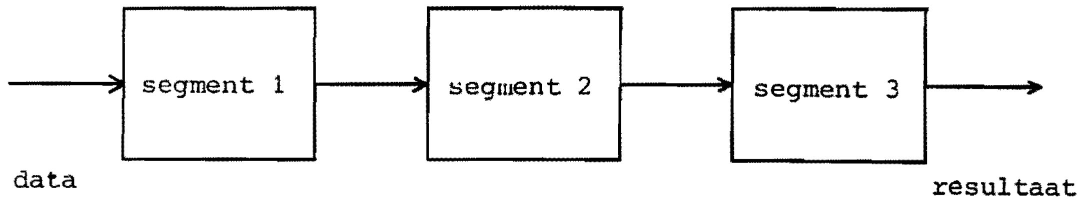
In paragraaf (5.2) wordt via een beknopte beschrijving van de architectuur en het gebruik van de AP500 een introduktie tot de arrayprocessor gegeven.

5.2. De AP500-arrayprocessor

Conventionele computers zijn niet in staat om complexe algoritmes danwel eenvoudige algoritmes op grote datasets naar volle tevredenheid uit te voeren. Dit komt omdat deze computers beperkt zijn door een architectuur waarin de data en de programma-instrukties in hetzelfde geheugen worden opgeslagen en de data bewerkt wordt door sequentiële instrukties te gebruiken. Daarentegen worden bij een arrayprocessor de data en de instrukties in verschillende geheugens opgeslagen en is deze in staat om verschillende operaties parallel uit te voeren. Deze architectuur biedt het hoofd aan de beperkingen van conventionele computers en lost het probleem van herhaalde berekeningen op zeer hoge snelheid op.

De architectuur van de AP500-arrayprocessor van Analogic maakt gebruik van een zogenaamde "pipeline". Dit wil zeggen, dat een uit te voeren funktie, gesplitst wordt in kleinere segmenten. Als een data-element door de pipeline vloeit, dan bezet het slechts één segment tegelijk, waarbij elk segment een rekenkundige of logische bewerking voor kan stellen. Zoals afgebeeld in figuur (5.2), verplaatst een data-element of operand-paar zich na bewerkt te zijn in segment 1, naar segment 2. Op hetzelfde moment treedt een nieuw data-element (of operand-paar) de pipeline binnen bij segment 1. Wanneer alle segmenten van de pipeline gelijktijdig bezet zijn, is de efficiëntie van de pipeline maximaal.

Een vereenvoudigde voorstelling van de pipeline van de AP500 is afgebeeld in figuur (5.3). De AP500 pipeline is veel complexer dan het drie-segmenten model van figuur (5.2). Het drie-segmenten model bijvoorbeeld houdt geen rekening met de mogelijkheid dat een data-element wanneer gewenst één of meerdere segmenten kan overslaan. Zoals afgebeeld in figuur (5.3) bevat de AP500 pipeline twee re-



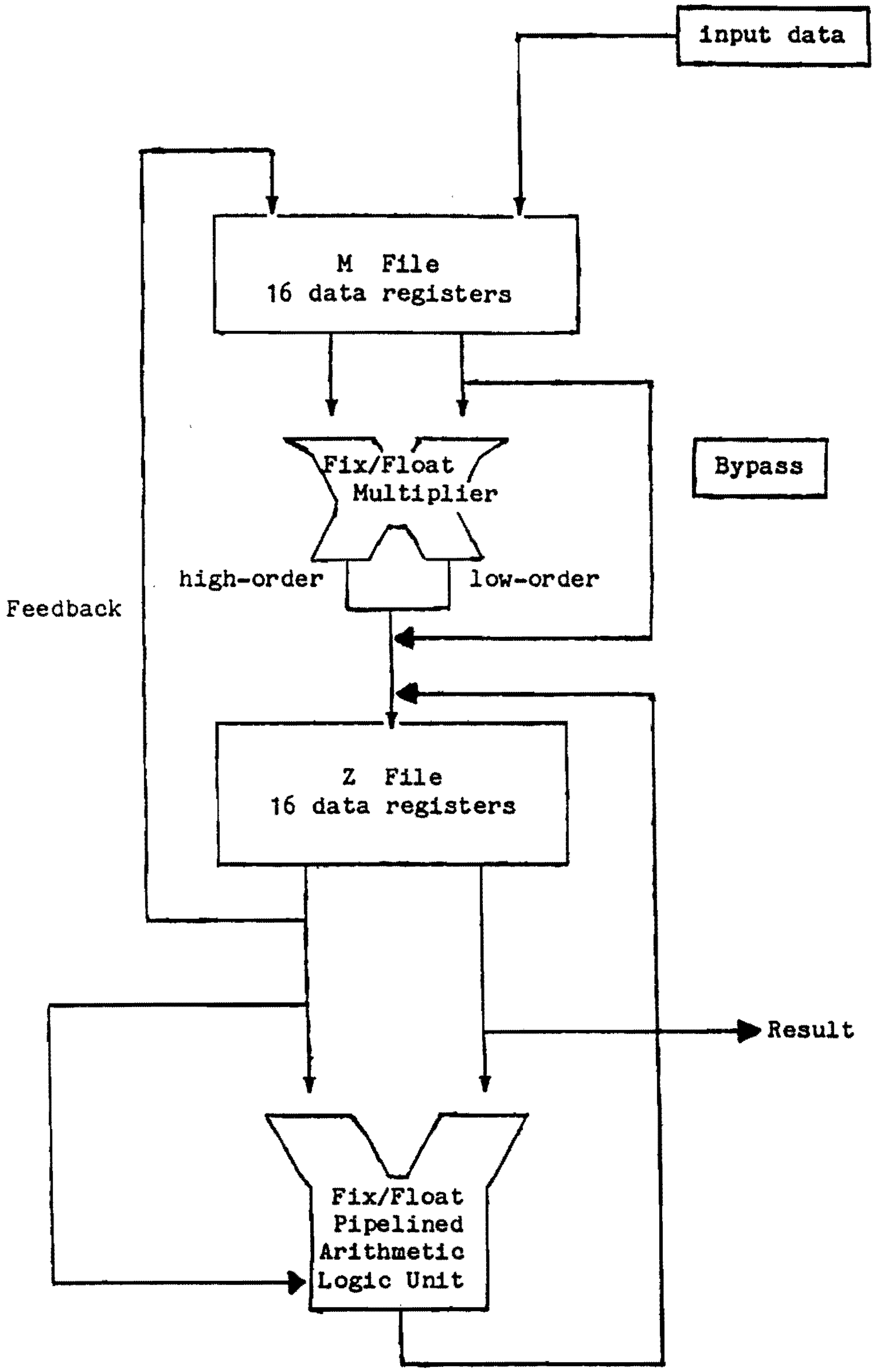
Figuur (5.2). Pipeline concept.

gister-files (Z en M), een vermenigvuldiger, een ALU ("arithmetic logical unit") en een "bypass"-eenheid. De M-file ontvangt de data uit het data-geheugen en levert de operanden voor de vermenigvuldiger. De Z-file bewaart de resultaten van de vermenigvuldigingen en van de ALU-bewerkingen. Ook levert de Z-file de invoer voor de ALU en de "feedback", en bewaart de eindresultaten van de pipeline-berekeningen voordat ze uit de pipeline worden gebracht. Elke pipeline-component staat onder microprocessor besturing. Dit stelt de data in staat om een component over te slaan of om terug gevoerd te worden naar een andere component.

Om een funktie uit te voeren moet een bepaalde volgorde van reken- en logische bewerkingen op een data-verzameling worden aangehouden. Deze volgorde wordt bepaald door de zogenaamde "pipeline-sequencer". Deze bevat microcode-instrukties die elke pipeline-component afzonderlijk kan besturen. Elke funktie correspondeert met een verzameling instrukties die reeds zijn geschreven door Analogic. Deze funkties kunnen apart of in allerhande combinaties worden gebruikt om vele verschillende problemen op te lossen.

De hoge snelheid van de pipeline is deels ook het gevolg van de mogelijkheid om herhaalde bewerkingen uit te voeren op gestruktureerde data-verzamelingen, zoals vectoren en matrices. Deze data-struktuur maakt het mogelijk om een snelle bewerking op alle elementen van een vector of een matrix uit te voeren. De technieken van de lineaire algebra bijvoorbeeld worden gebruikt om alle elementen van een vector of matrix te bewerken, door één enkele instruktie te gebruiken. Dit reduceert in hoge mate het aantal instrukties dat nodig is voor een toepassing.

De meest eenvoudige manier om de AP500 te gebruiken is d.m.v. de host in een hogere programmeertaal subroutines aan te roepen. Voor de meeste toepassingen geeft dit voldoende snelle dataverwerking en vereist de minste software-ontwikkeltijd. Het is ook mogelijk om op host-assembler-niveau te programmeren, of zelfs op AP500 niveau. Hoewel de bewerkingstijd hiermee verkleind wordt, neemt



Figuur (5.3). Pipeline-architectuur.

de benodigde programma-ontwikkelingstijd hiermee evenredig toe. Omdat praktisch alleen met subroutine-calls in een hogere programmeertaal gewerkt wordt, zal de bespreking zich hiertoe beperken.

De voor de AP500 geschreven subroutines, de zg. K- en L-funkties zijn verzameld in een bibliotheek, welke bij het vertalen van een programma op de host meegelinkt moet worden. Programma's opgebouwd m.b.v. K- en L-funkties hebben in het algemeen de volgende struktuur:

(1) initialiseer de AP500
(2) data-transport naar de AP500
(3) data-bewerking
(4) data-transport uit de AP500.

Wanneer alle punten van een vector (of matrix) bewerkt moeten worden, dan wordt een K-funktie aangeroepen, zoals bijvoorbeeld KADD voor vector-optelling. Wanneer een aaneengesloten verzameling van elementen binnen de vector, maar niet de hele vector moet worden bewerkt, wordt een L-funktie gebruikt, zoals LADD. Het in en uit de AP500 transporteren van data wordt gedaan met een K- of L-funktie die in overeenstemming is met het dataformaat (b.v. real of integer) en de lokatie van de data. Indien een vector bewerkt moet worden, wordt deze in z'n geheel door een datatransportfunktie in een databuffer in het datageheugen van de arrayprocessor geplaatst. Het is daarom noodzakelijk dat de lokaties van de verschillende vectoren in het datageheugen worden bijgehouden. Dit wordt gedaan door elke databuffer waarin een vector wordt geplaatst een identifikatienummer te geven.

Tenslotte wordt hieronder nog aangegeven waar de specifieke informatie omtrent de AP500-arrayprocessor te vinden is.

(1) <u>Een introduktie tot de AP500</u>: boek "User Manuals", deel "Introduction".

(2) <u>Een introduktie tot het programmeren van de AP500</u>: boek "User Manuals", deel "RSX-11 M Primer".

(3) <u>Een uitgebreide gebruikshandleiding van de K- en L-funkties</u>: boek "User Manuals", deel "RSX-11 m Host User Manual", hoofdstukken 1, 2 en 3 (plus eventueel 4).

(4) <u>De beschrijving van alle K- en L-funkties</u>: boek "Function Reference Manual", deel "AP500 functions".

LITERATUURLIJST

[1] Abramatic, J.F., "Digital image restoration", *Fundamentals in computer vision*, edited by O.D. Faugeras, 1983.

[2] Albert, A., "*Regression and the Moore-Penrose Pseudoinverse*", Academic Press, New York, New York, 1972.

[3] Andrews, H.C., Hunt, B.R., "*Digital image restoration*", Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1977.

[4] Biemond, J., "Image restoration as an identification and filtering problem", *Pattern Recognition in Practice*, Amsterdam, June 1985.

[5] Biemond, J., Putten, F.G. v.d., "Image restoration using a parallel identification and filtering procedure", *ICASPP-85*, Tampa, Florida, pp. 660-663.

[6] Biemond, J., Rieske, J., Gerbrands, J.J., "A fast Kalman filter for images degraded by both blur and noise", *IEEE Trans. on Acoustics, Speech and Signal Proc.*, vol. ASSP-31, no. 5, oct. 1983.

[7] Bruin, P. de, "*Digitale reconstructie van kleurenbeelden*", afstudeerverslag, TH Delft, aug. 1985.

[8] Burch, S.F., "*A comparison of image restauration methods*", R9671, Materials Physics division, Aera, Harwell, March 1980.

[9] Cannon, T.M., "Blind deconvolution of spatially invariant blurs with phase", *IEEE Trans. on Ac., Speech and Signal Proc.*, vol. ASSP-24, no. 1, Feb. 1976.

[10] Davidson, M., "Perturbation approach to spatial brightness interaction in human vision", *J. Opt. Soc. Amer.*, vol. 58, pp. 1300-1309, 1968.

[11] Gerbrands, J.J., "*Inleiding in de digitale beeldverwerking*", collegedictaat TH Delft, nov. 1982.

[12] Hunt, B.R., "The application of constrained least-squares estimation to image restoration by digital computer", *IEEE Trans. on Comp.*, vol. C-22, no. 9, sept. 1973.

[13] Hunt, B.R. "Bayesian methods in nonlinear digital image restoration", *IEEE Trans. on Comp.*, vol. C-26, no. 3, March 1977.

[14] Hunt, B.R., "Digital Image Processing", *Proc. of IEEE*, vol. 63, no. 4, April 1975.

[15] Legault, R., "Aliasing problems in two-dimensional sampled imagery", in *Perception of displayed information*, L. Biberman, Ed. New York: Plenum Press, ch. 7, 1973.

[16] Mees, C.A.K., *"The theory of the photographic process"*, New York, Mac Millan, 1954.

[17] Philips, D.L., "A technique for the numerical solution of certain integral equations of the first kind", *JACM*, vol. 9, 1962, pp. 97-101.

[18] Putten, F. v.d., *"Beeldrekonstruktie m.b.v. een parallel identificatie- en filterschema"*, afstudeerverslag, TH Delft, aug. 1984.

[19] Reys, A.E.M., *"Een onderzoek naar de mogelijkheid de contrastgevoeligheidsfunctie van het menselijk visuele systeem toe te passen in de digitale beeldrekonstruktie"*, afstudeerverslag, TH Delft, febr. 1980.

[20] Rosenfeld, A., Kak, A.C., *"Digital picture processing"*, Academic Press, New York, 1976.

[21] Stockham, T.G., "Image processing in the context of a visual model", *IEEE Proc.*, vol. 60, pp. 828-841.

[22] Walkup, J.F., Choens, R.C., "Image processing in signal-dependent noise", *Opt. Eng.*, vol. 13, May 1974, pp. 258-266.

[23] Westerink, P.H., *"Vervormingsidentifikatie in het tijd/plaats-domein m.b.v. autoregressieve modelvorming"*, taakverslag, TH Delft, jan. 1984.

[24] Woods, J., Biemond, J., Tekalp, A.M., "Boundary value problem in image restoration", *ICASSP-85*, Tampa, Florida, pp. 692-695.

[25] Eerland, K.K., *"Kalmanfilteren op vervormde en verruiste beelden"*, afstudeerverslag, TH Delft, jan. 1981.

BIJLAGE A

Afleiding Wiener filter

Kort samengevat kan het minimaliseringsprobleem zoals dat voor het Wiener-filter is opgesteld, worden genoteerd als:

criterium	minimaliseer $\min_{\hat{F}} E\{\|F - \hat{F}\|^2\}$	(A.1)
model	$G = H.F + N$	(A.2)
oplossingsvorm	$\hat{F} = L.G.$	(A.3)

Gevraagd wordt dus een lineaire oplossing L in de vorm van vergl. (A.3). Ga daartoe uit van vergl. (A.1) en vul hierin achtereenvolgens vergl. (A.3) en (A.2) in:

$$\min_{\hat{F}} E\{|F - \hat{F}|^2\} \Rightarrow$$

$$\min_{L} E\{|F - LG|^2\} \Rightarrow$$

$$\min_{L} E\{|F - L(HF+N)|^2\}. \tag{A.4}$$

Door te bedenken dat in het algemeen $|A|^2 = AA^* = A^*A$, kan vergl. (A.4) worden geschreven als

$$\min_{L} E\{FF^*-FF^*H^*L^*-FF^*HL + (FF^*HH^*+NN^*)LL^*+ + F^*N(H^*LL^*-L) + FN^*(HLL^*-L^*)\}. \tag{A.5}$$

Omdat in het model de termen f en n ongecorreleerd zijn verondersteld, zijn de termen F^*N en FN^*, die de kruiscorrelatie tussen f en n representeren, nul, en kunnen we vergl. (A.5) schrijven als

$$\min_{L} E\{FF^*-(FF^*H^*)L^* - (FF^*H)L + (FF^*HH^* + NN^*)LL^*\}. \tag{A.6}$$

Om te minimaliseren mag vergl. (A.6) niet zomaar naar L worden gedifferentieerd, omdat de funktie $f(L) = L^*$ niet analytisch is. We splitsen daarom H en L in hun reëele en imaginaire delen:

$$H = h_1 + ih_2 \qquad (H^* = h_1 - ih_2), \tag{A.7}$$

$$L = \ell_1 + i\ell_2 \qquad (L^* = \ell_1 - i\ell_2).$$

Vergl. (A.6) gaat dan over in

$$\min_L E\{FF^*-FF^*(h_1-ih_2)(\ell_1-i\ell_2) - FF^*(h_1+ih_2)(\ell_1+i\ell_2) + $$
$$+(FF^*HH^*+NN^*)(\ell_1{}^2+\ell_2{}^2)\} \Rightarrow$$

$$\min_L E\{FF^*-2FF^*(h_1\ell_1-h_2\ell_2) + (FF^*HH^*+NN^*)(\ell_1{}^2+\ell_2{}^2)\} \Rightarrow$$

$$\min_L E\{FF^*+[(FF^*HH^*+NN^*)\ \ell_1{}^2 - 2FF^*h_1\ell_1] +$$
$$+ [(FF^*HH^*+NN^*)\ \ell_2{}^2 + 2FF^*h_2\ell_2]\}. \tag{A.8}$$

Zoals het (reëele) minimaliseringsproblcem nu is genoteerd, kan dit gesplitst worden in de minimalisering voor ℓ_1 resp. ℓ_2:

$$\min_L E\{FF^*+ [\text{funktie van } \ell_1] + [\text{funktie van } \ell_2]\} \Rightarrow$$

$$E\{FF^*\} + \min_{\ell_1} E\{[\text{funktie van } \ell_1]\} + \min_{\ell_2} E\{[\text{funktie van } \ell_2]\}. \tag{A.9}$$

De oplossingen voor ℓ_1 en ℓ_2 volgen nu direkt door differentiëren van vergl. (A.9) naar resp. ℓ_1 en ℓ_2:

$$\min_{\ell_1} E\{(FF^*HH^*+NN^*)\ \ell_1{}^2 - 2FF^*h_1\ell_1\} \Rightarrow$$

$$\frac{\partial}{\partial \ell_1}\{(FF^*HH^*+NN^*)\ \ell_1{}^2 - 2FF^*h_1\ell_1\} = 0 \Rightarrow$$

$$2(FF^*HH^*+NN^*)\ \ell_1 + 2FF^*h_1 = 0$$

$$\ell_1 = \frac{FF^*}{FF^*HH^*+NN^*}\, h_1. \tag{A.10}$$

Op dezelfde wijze vinden we

$$\ell_2 = -\frac{FF^*}{FF^*HH^*+NN^*}\ h_2. \tag{A.11}$$

Combineren we tenslotte vergl. (A.10) en (A.11), dan levert dit

$$L = \ell_1 + i\ell_2 = \frac{FF^*}{FF^*HH^*+NN^*}\ (h_1-ih_2) \Rightarrow$$

$$L = \frac{FF^*H^*}{FF^*HH^*+NN^*}, \tag{A.12}$$

hetgeen met $S_n(u,v) = N(u,v)\ N^*(u,v)$ en $S_f(u,v) = F(u,v)\ F^*(u,v)$ leidt tot de overdrachtsfunktie $L(u,v)$ van het Wiener filter:

$$L(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \dfrac{S_n(u,v)}{S_f(u,v)}}. \tag{A.13}$$

BIJLAGE B

(1) Afleiding kleinste-kwadraten filter met randvoorwaarden

Het minimaliseringsprobleem voor het kleinste-kwadraten filter met randvoorwaarden luidt kort samengevat als volgt:

criterium minimaliseer $\underset{\hat{F}}{E\{|C\hat{F}|^2\}}$ (B.1)

randvoorwaarde $E\{|G - H\hat{F}|^2\} = E\{|N|^2\}$. (B.2)

Zoals gezegd in par. 3.6 wordt de (lineaire) oplossing gevonden m.b.v. de methode van Lagrange. Definiëer daartoe de funktie

$$J(\hat{F}) = E\{|C\hat{F}|^2\} + \lambda[E\{|G - H\hat{F}|^2\} - E\{|N|^2\}], \quad (B.3)$$

waarin λ een z.g. Lagrange vermenigvuldiger is. M.b.v. $|A|^2 = AA^* = A^*A$ kan deze funktie uitgeschreven worden tot

$$J(\hat{F}) = E\{CC^*\hat{F}\hat{F}^* + \lambda[GG^* - GH^*\hat{F}^* - G^*H\hat{F}^* + HH^*\hat{F}\hat{F}^* - NN^*]\}$$

$$= E\{(CC^* + \lambda HH^*)\hat{F}\hat{F}^* - (\lambda GH^*)\ \hat{F}^* - (\lambda G^*H)\hat{F} + \lambda(GG^* - NN^*)\}. \quad (B.4)$$

We willen nu $J(\hat{F})$ minimaliseren, maar kunnen dit niet doen door te differentiëren naar $\hat{F}$, omdat de funktie $f(\hat{F}) = \hat{F}^*$ niet analytisch is. Splits daarom $\hat{F}$, G en H in hun reëele en imaginaire delen:

$$\begin{aligned} \hat{F} &= \hat{f}_1 + i\hat{f}_2, \\ G &= g_1 + ig_2, \\ H &= h_1 + ih_2. \end{aligned} \quad (B.5)$$

Na uitschrijven gaat $J(\hat{F})$ over in de funktie

$$J(\hat{f}_1,\hat{f}_2) = E\{(CC^* + \lambda HH^*)(\hat{f}_1^{\,2} + \hat{f}_2^{\,2}) + \lambda(GG^* - NN^*) +$$

$$-2\lambda[(h_1g_1 + h_2g_2)\ \hat{f}_1 - (h_2g_1 - h_1g_2)\ \hat{f}_2]\}. \quad (B.6)$$

Deze reëele funktie kunnen we nu wel minimaliseren, door de partiële afgeleiden naar resp. $\hat{f}_1$ en $\hat{f}_2$ gelijk aan nul te stellen

$$\min_{\hat{F}} J(\hat{F}) \Rightarrow \begin{cases} \dfrac{\partial}{\partial \hat{f}_1} J(\hat{f}_1,\hat{f}_2) = 0 \\ \dfrac{\partial}{\partial \hat{f}_2} J(\hat{f}_1,\hat{f}_2) = 0 \end{cases} \tag{B.7}$$

De oplossingen voor $\hat{f}_1$ en $\hat{f}_2$ worden m.b.v. vergl. (B.6):

$$\hat{f}_1 = \frac{\lambda}{CC^* + \lambda HH^*} (h_1 g_1 + h_2 g_2) \tag{B.8}$$

$$\hat{f}_2 = \frac{\lambda}{CC^* + \lambda HH^*} (h_1 g_2 - h_2 g_1), \tag{B.9}$$

waarmee we m.b.v. $\hat{F} = \hat{f}_1 + i\hat{f}_2$ krijgen:

$$\hat{F} = \frac{\lambda}{CC^* + \lambda HH^*} \{(h_1 g_1 + h_2 g_2) + i(h_1 g_2 - h_2 g_1)\}, \tag{B.10}$$

en wordt dus de kleinste-kwadraten schatter met randvoorwaarden gegeven door

$$\hat{F}(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \gamma |C(u,v)|^2} G(u,v), \tag{B.11}$$

waarin $\gamma = 1/\lambda$.

(2) Iteratieve bepaling van de parameter γ

De parameter γ moet in het "constrained least-squares filter" van vergl. (B.11) zó worden ingesteld, dat aan de randvoorwaarde van vergl. (B.2) is voldaan. Vaak wordt hiervoor een Newton-Raphson-achtige methode gebruikt, die kwadratisch convergeert [12].

Deze methode werkt als volgt: indien een variabele x moet voldoen aan de vergelijking $f(x) = c$, dan kan x iteratief worden benaderd met de formule

$$x_{i+1} = x_i - \frac{f(x_i) - c}{f'(x_i)}, \tag{B.12}$$

waarin $f'(x_i)$ de afgeleide van $f(x_i)$ naar x_i is.

We kunnen de bepaling van γ vertalen naar de methode van Newton-Raphson, door de funktie

$$R(\gamma) = E\{|G - H\hat{F}|^2\} \tag{B.13}$$

te beschouwen, en een oplossing te zoeken voor

$$R(\gamma) = E\{|N|^2\}. \tag{B.14}$$

Vullen we vergl. (B.11) in vergl. (B.13) in, dan kunnen we $R(\gamma)$ schrijven als

$$R(\gamma) = E\{(\frac{\gamma CC^*}{HH^* + \gamma CC^*})^2 \; GG^*\}. \tag{B.15}$$

Indien we nu $R(\gamma)$ differentiëren naar γ, dan volgt na uitschrijven, dat

$$R'(\gamma) = \frac{2}{\gamma^2} E\{(\frac{\gamma CC^*}{HH^* + \gamma CC^*})^3 \; (\frac{HH^*}{CC^*}) \; GG^*\}. \tag{B.16}$$

Hiermee kan dus m.b.v. de methode van Newton-Raphson de waarde van γ iteratief bepaald worden, met

$$\gamma_{K+1} = \gamma_K - \frac{R(\gamma_K) - E\{NN^*\}}{R'(\gamma_K)}, \tag{B.17}$$

uitgaande van een beginwaardc γ_0. De iteratie wordt gestopt zodra een relatieve nauwkeurigheid $\varepsilon > 0$ is gehaald:

$$|\frac{\gamma_{K+1} - \gamma_K}{\gamma_{K+1}}| < \varepsilon. \tag{B.18}$$

BIJLAGE C

2-D Fast Fouriertransformatie van een beeld m.b.v. de AP500

Een beeld, dat in de AP500 rijgewijs is opgeslagen, wordt door de 2-D FFT als volgt behandeld:

(1) een reëele 1-D FFT wordt uitgevoerd over alle reëele rijen,

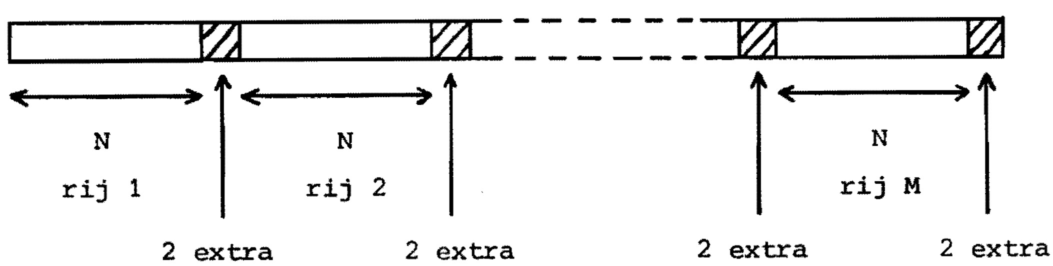
(2) vervolgens wordt een complexe 1-D FFT uitgevoerd over alle verkregen complexe kolommen.

De reëele 1-D FFT heeft echter twee geheugenplaatsen meer nodig voor het resultaat dan er plaatsen zijn in een beeldrij. Dit kan men als volgt inzien: de 1-D FFT van een reëele funktie $f(k)$ heeft als eigenschap dat voor zijn Fourier-getransformeerde $F(\omega)$ geldt:

$$\left.\begin{aligned} \mathrm{Re}\{F(N-\omega)\} &= \mathrm{Re}\{F(\omega)\} \\ \mathrm{Im}\{F(N-\omega)\} &= -\,\mathrm{Im}\{F(\omega)\}. \end{aligned}\right. \qquad (\omega=0,\ldots,N-1) \qquad (C.1)$$

Hieruit volgt dat we slechts de eerste $N/2+1$ complexe waarden van alle N waarden nodig hebben; de overige $N/2-1$ waarden kunnen hieruit berekend worden. De AP500 slaat alleen de $N/2+1$ complexe waarden op, zodat $N+2$ geheugenplaatsen nodig zijn.

Het gevolg is, dat bij een rijgewijs opgeslagen beeld met M rijen en N kolommen, achter elke rij twee extra geheugenplaatsen opengelaten moeten worden, zoals dit is afgebeeld in figuur (C.1).



Figuur (C.1). Wijze van opslag van een beeld t.b.v. de 2-D FFT.

Dit betekent dat een beeld waarop de 2-D FFT wordt uitgevoerd, in de AP500 opgeslagen dient te worden in een databuffer met een lengte van $M(N+2)$ punten. Het 2-D Fouriergetransformeerde beeld bestaat uit M rijen met in elke rij N/2+1 complexe datapunten.

Het frekwentienulpunt bevindt zich hierbij linksboven in de hoek van het Fouriergetransformeerde beeld, dus het eerste complexe data-element van de databuffer.

DIGITALE SIGNAALBEWERKING

deel II.7

Switched Capacitor Filters

door

J.A. Hegt

Technische Universiteit Eindhoven

SWITCHED CAPACITOR FILTERS

dr.ir. J.A. Hegt
Technische Universiteit Eindhoven

1. INLEIDING

Nadat A. Fettweis zijn theorie van de resonant-transfer circuits [1] in 1968[1] publiceerde, gevolgd door een beroemd artikel van Fried [2] in 1972 en een aantal jaren van stilte, is de ontwikkeling van switched capacitor (SC) filters van basisconcept tot veelvuldig toegepast commercieel produkt uitzonderlijk snel verlopen.

Er is een aantal redenen te geven voor hun populariteit.
SC filters

- zijn goedkoop,
- gebruiken weinig oppervlak,
- zijn low power,
- bezitten een voor analoge filters hoge mate van nauwkeurigheid en hoeven na produktie i.h.a. dan ook niet afgeregeld te worden,
- anderzijds kan de overdracht van het filter worden getransformeerd naar een ander frekwentiegebied door eenvoudigweg de klokfrekwentie te wijzigen; dit kan bijvoorbeeld attraktief zijn in toepassingen bij adaptieve filters,
- zijn volledig implementeerbaar in standaard MOS processen,
- kunnen gekombineerd worden met digitale circuits op één en dezelfde chip.

Toch zijn er ook wel effekten te noemen die het werken met SC filters bemoeilijken.
De bouwstenen waar deze filters uit zijn opgebouwd: condensatoren, schakelaars (meestal MOS-transistoren) en versterkers zijn behept met

---

[1] *In feite wordt door J.C. Maxwell het principe van geschakelde condensatoren al beschreven in zijn in 1873 gepubliceerde boek "A treatise on electricity and magnetism".*

vele niet-idealiteiten, zoals

- parasitaire capaciteiten,
- eindige open-loop DC versterking en bandbreedte van de gebruikte versterkers,
- eindige aan-geleiding van de schakelaars,
- eindige geleiding van de uitgangen van de versterkers en begrensde current sourcing en sinking,
- niet-lineariteit van de (parasitaire) capaciteiten, schakelaars en versterkers,
- overspraak van het kloksignaal,
- thermische en 1/f ruis in de MOS-fets van de versterkers en schakelaars,
- etc.

Het zijn o.a. deze niet-idealiteiten die de analyse en synthese van SC filters bemoeilijken.

In zeer kort bestek worden hier de "equivalentie" tussen geschakelde condensatoren en weerstanden bekeken en enige analyse- en synthesemethoden besproken. Voor een uitgebreider en diepgaander behandeling van deze onderwerpen wordt verwezen naar de literatuur, zoals [3-7].

2. "EQUIVALENTIE" TUSSEN GESCHAKELDE CONDENSATOREN EN WEERSTANDEN

In fig.1 wordt een condensator C middels twee schakelaars, gemerkt "o" (welke geleidt in de oneven klokfase) en "e" (geleidend in de even klokfase) beurtelings verbonden met een spanningsbron met spanning U_1 dan wel met spanning U_2.
Op tijdstip $t=t_0$ is de schakelaar "e" aan en "o" uit, condensator C is opgeladen tot de spanning $v_c(t_0)= U_2$ en op de condensator bevindt zich een hoeveelheid lading $q(t_0)= CU_2$. Vervolgens gaat ook schakelaar "e" uit en daarna schakelaar "o" aan. Condensator C wordt ogenblikkelijk opgeladen, zodat op $t=t_0+T/2$ zich hierover een spanning heeft opgebouwd ter grootte $v_c(t_0+T/2)= U_1$ en $q(t_0+T/2)= CU_1$. De bron U_1

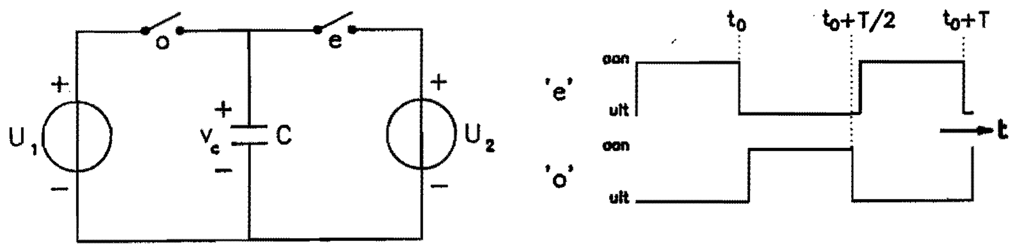


fig.1: voorbeeld geschakelde condensator

heeft hiertoe een ladingshoeveelheid $\Delta q^o = q(t_0+T/2) - q(t_0) = C(U_1-U_2)$ afgestaan. Schakelaar "o" gaat weer uit en vervolgens schakelaar "e" aan. Condensator C ontlaadt zich nu, zodat op $t=t_0+T$ de spanning over de condensator is afgenomen tot $v_c(t_0+T) = U_2$. Aan de bron U_2 is hiertoe een ladingshoeveelheid $\Delta q^e = q(t_0+T) - q(t_0+T/2) = C(U_1-U_2)$ afgestaan. Hiermee is een complete cyclus van deze schakeling beschreven. Netto wordt er per tijdsinterval T door middel van de geschakelde condensator een ladingshoeveelheid $C(U_1-U_2)$ getransporteerd van de bron U_1 naar de bron U_2. De per tijdseenheid getransporteerde hoeveelheid lading bedraagt $C(U_1-U_2)/T$, overeenkomend met de stroom door een weerstand met waarde T/C, aangesloten tussen de spanningsbronnen U_1 en U_2. Men kan dan ook stellen dat de equivalente weerstandswaarde van de geschakelde condensator in dit voorbeeld gelijk is aan

$$R_{eq} = \frac{T}{C} = \frac{1}{f_s C}$$

waarbij $f_s = 1/T$ de klokfrekwentie representeert. De eigenschap dat de grootte van de equivalente weerstandswaarde mede bepaald wordt door de klokfrekwentie biedt de mogelijkheid om een SC filter op eenvoudige wijze in zijn geheel te verstemmen. Merk op dat een grote equivalente weerstandswaarde correspondeert met een kleine geschakelde capaciteit, die dan ook weinig chipoppervlak in beslag neemt. Tevens kan aan de hand van bovenstaande formule worden ingezien dat waar in tijdcontinue filters de eigenschappen bepaald worden door produkten van R en C, deze bij SC filters juist bepaald worden door *verhoudingen* van geschakelde en ongeschakelde capaciteiten. Alhoewel de absolute nauwkeurigheid van geïntegreerde capaciteiten slecht is (±10%) heeft

men de onderlinge verhouding van capaciteiten op één en dezelfde chip veel beter in de hand: toleranties van ±0.1% zijn heel normaal. Dit is de reden waarom geïntegreerde SC filters i.h.a. veel nauwkeuriger zijn dan geïntegreerde filters, opgebouwd met onnauwkeurige weerstanden (±15%) en condensatoren.
De "equivalentie" van geschakelde condensator enerzijds en weerstand anderzijds is slechts ten dele korrekt. Een kenmerkende eigenschap van SC filters is, dat zij behoren tot de klasse van tijddiscrete schakelingen. De getransporteerde ladingshoeveelheden worden o.a. bepaald door samples van knooppuntspanningen, genomen op het einde van de klokfasen. Het gedrag van SC filters wordt dan ook i.h.a. beschreven in termen van z-getransformeerden van spanningen op het einde van klokfasen en z-getransformeerden van per klokfase getransporteerde ladingshoeveelheden.

3. ANALYSE VAN SC FILTERS

Elke analysemethode voor SC filters maakt (evt. impliciet) gebruik van ladingsbehoud binnen bepaalde contouren.
Alhoewel het direkt uitschrijven van alle op ladingsbehoud gebaseerde vergelijkingen per klokfase een krachtige analysemethode is, is dit ook een erg omslachtige werkwijze. Vooral voor grotere SC schakelingen wordt i.h.a. gebruik gemaakt van methoden waarbij ladingsbehoud impliciet wordt verdisconteerd. Veelal zijn deze methoden alleen geldig bij toegestane verwaarlozing van een groot aantal niet-idealiteiten en hebben dus een beperkt toepassingsgebied. Blijken een aantal niet-idealiteiten een signifikante invloed te hebben op het filtergedrag, dan is men voor analyse van deze schakelingen meestal aangewezen op een van de vele simulatieprogramma's zoals DIANA, DOSCA, DONES, PRSC, SCAPN, SCANAL, SCNAP, SCNET, SCYMBAL, SCOP, SWAP, SWICAP, SWINET, SWITCAP, WASCAP, WATSCAD, etc.
Van de "handmatige" analysemethoden wordt hier, zij het summier, een tweetal behandeld.
Bij de eerste methode wordt met equivalente admittanties een schakeling opgebouwd die een spanning/stroom-gedrag vertoont overeenkomstig een beschrijving in het z-domein van de spanningen en

getransporteerde ladingen in het SC circuit. Vervolgens wordt deze equivalente schakeling geanalyseerd.
De tweede methode gebruikt een graph-voorstelling van de vergelijkingen waarmee deze z-getransformeerde spanningen en getransporteerde ladingen in elkaar worden uitgedrukt. Deze graph kan dan met knooppunteliminatie [8] of m.b.v. de "regel van Mason" [9] worden doorgerekend.

3.1 ANALYSE M.B.V. EQUIVALENTE SCHAKELINGEN IN HET Z-DOMEIN

Deze methode [10] verloopt als volgt.

- Eerst wordt het te analyseren circuit opgesplitst in elementaire bouwblokken. Alhoewel in de oorspronkelijke methode deze "elementaire" bouwblokken nogal wat componenten bevatten, is het ook mogelijk de schakeling op te splitsen in zijn losse componenten: condensatoren, schakelaars en versterkers.
- Voor deze elementaire bouwblokken worden equivalente schakelingen in het z-domein opgesteld, waarvoor dezelfde poortrelaties voor de spanningen en getransporteerde ladingen in het z-domein per klokfase gelden als voor het oorspronkelijke circuit. Deze equivalente circuits worden samengesteld uit equivalente admittanties in het z-domein, die de relaties representeren tussen spanningen en getransporteerde ladingen in het z-domein, op dezelfde wijze als "normale" admittanties de relaties tussen spanningen en stromen beschrijven in het Laplace s-domein.
- Vervolgens worden deze equivalente deelschakelingen weer met elkaar verbonden overeenkomstig de topologie van het oorspronkelijke circuit.
- Als laatste stap wordt deze samengestelde equivalente schakeling in het z-domein geanalyseerd, waarbij gebruik wordt gemaakt van de "normale" Kirchhoff spanningswet voor de (z-getransformeerde) spanningen en de Kirchhoff stroomwet voor de (z-getransformeerde) ladingsstromen.

Zoals reeds toegelicht is het niet nodig een uitgebreide tabel van elementaire circuits met hun equivalenten aan te leggen, maar volstaat het de equivalente schakelingen te vinden voor condensatoren, schakelaars en versterkers.

Beschouw een condensator C (fig.2) welke wordt verondersteld deel uit te maken van een SC circuit met een tweefasen klok, met fasen "e" en "o". (Uitbreiding tot meer dan twee klokfasen is eenvoudig mogelijk.)

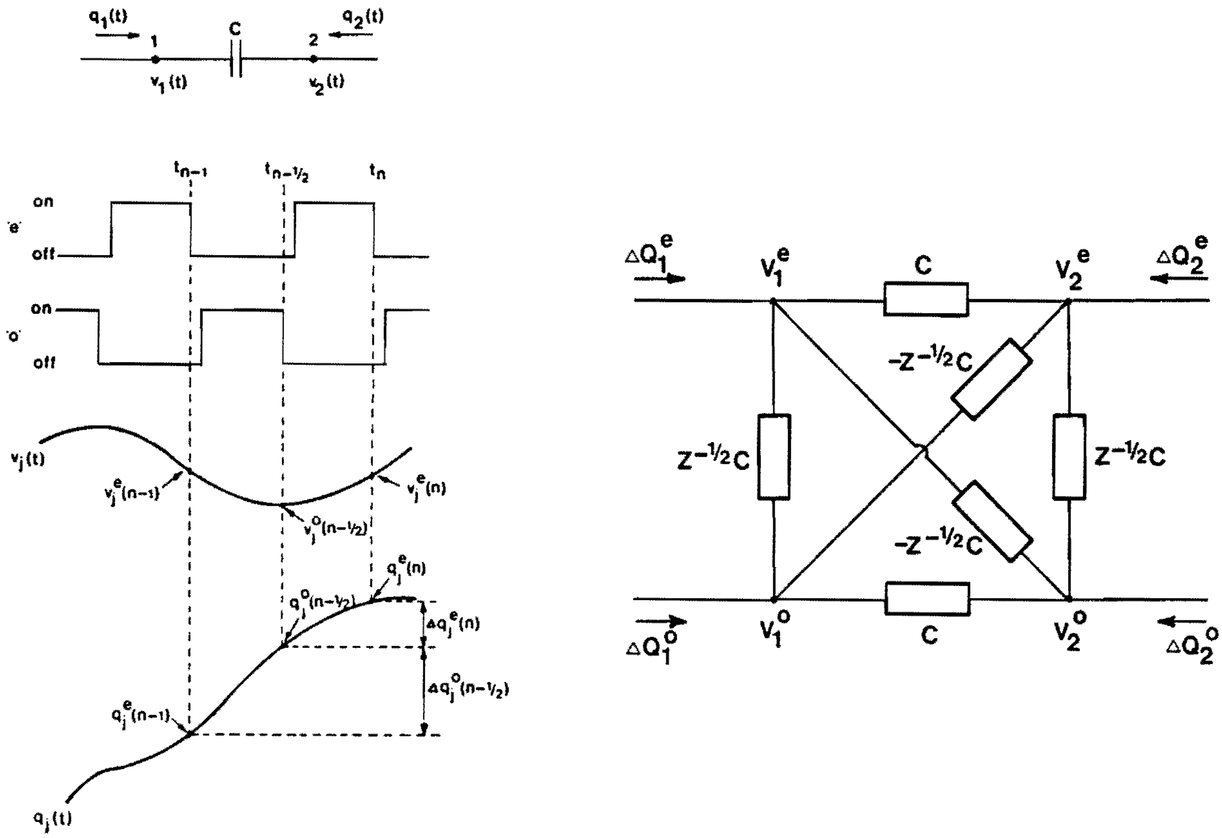


fig.2: Condensator met corresponderende equivalente schakeling in het z-domein

Gebruikmakend van ladingsbehoud geldt:

$$\Delta q_1^e = q_1^e(n)-q_1^o(n-1/2) = C\ [v_1^e(n)-v_2^e(n)-v_1^o(n-1/2) + v_2^o(n-1/2)]$$

$$\Delta q_1^o(n-1/2) = q_1^o(n-1/2)-q_1^e(n-1) = C\ [v_1^o(n-1/2)-v_2^o(n-1/2)-v_1^e(n-1)+v_2^e(n-1)]$$

$\Delta q_2^e(n) = -\Delta q_1^e(n)$

$\Delta q_2^o(n-1/2) = -\Delta q_1^o(n-1/2)$

(waarbij $\Delta q_1^e(n)$de hoeveelheid lading is welke door de buitenwereld aan klem 1 wordt toegevoerd in het interval $t_{n-1/2}<t\leq t_n$,etc.).

Of, na z-transformatie:

$$\Delta Q_1^e = CV_1^e - CV_2^e - z^{-1/2}\,CV_1^o + z^{-1/2}\,CV_2^o$$

$$\Delta Q_1^o = CV_1^o - CV_2^o - z^{-1/2}\,CV_1^e + z^{-1/2}\,CV_2^e$$

$$\Delta Q_2^e = -\Delta Q_1^e$$

$$\Delta Q_2^o = -\,\Delta Q_1^o$$

Deze relaties tussen ladingshoeveelheden ΔQ_i^j en spanningen V_k^l kunnen worden gerepresenteerd door het equivalente circuit in het z-domein volgens fig.2.

De equivalente schakeling voor een schakelaar die "aan" is gedurende klokfase "e" en "uit" is in fase "o" wordt gegeven in fig.3.

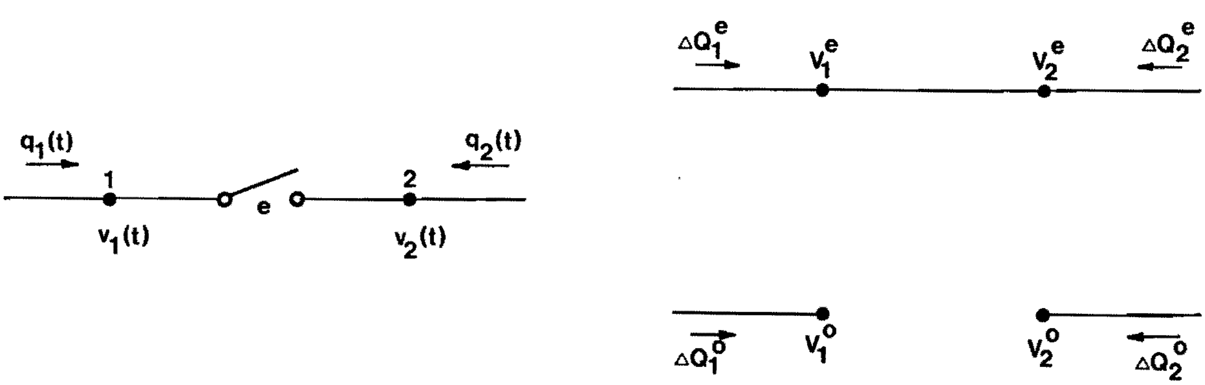


fig.3: Schakelaar met corresponderende equivalente schakeling in het z-domein

Vanzelfsprekend mogen fasen "e" en "o" onderling verwisseld worden.

Het equivalent voor een spanningsgestuurde spanningsbron (of versterker) met (frekwentie onafhankelijke) versterking A_0 bestaat uit twee spanningsgestuurde spanningsbronnen, één voor elke klokfase, zoals geschetst in fig.4.

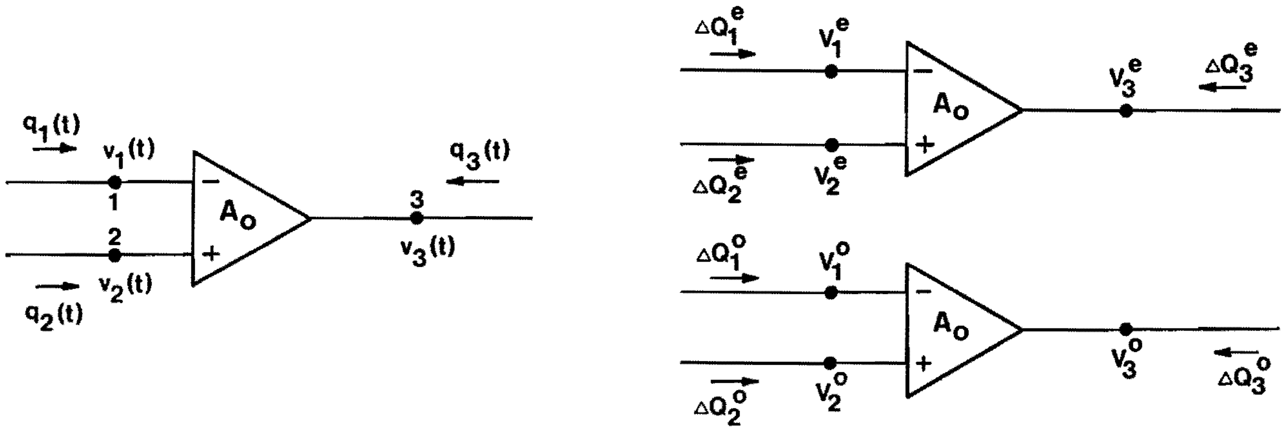


fig.4: Spanningsgestuurde spanningsbron met equivalente schakeling in het z-domein

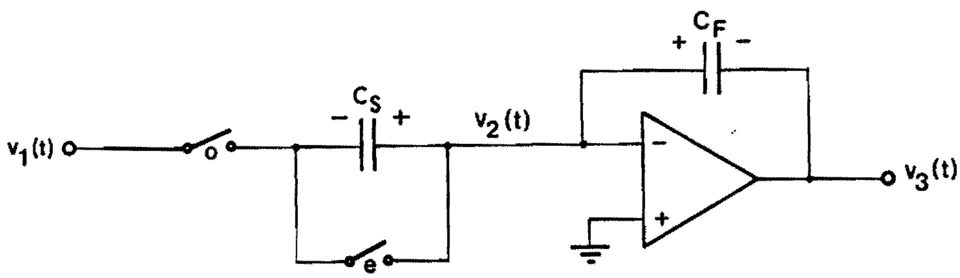
Wanneer als voorbeeld de schakeling van fig.5a wordt geanalyseerd, dan levert dit de equivalente schakeling van fig.5b op. Hierbij is aangenomen dat, afgezien van een eindige versterking A_0 van de opamp, de schakeling van fig.5a ideaal is. De gestippeld aangegeven admittanties zijn dan redundant, resulterend in de overdrachten

$$H^{oo}(z) = \frac{V_3^o(z)}{V_1^o(z)} = \frac{-C_s}{(1-z^{-1})\ (1+1/A_o)C_F + C_s/A_o}$$

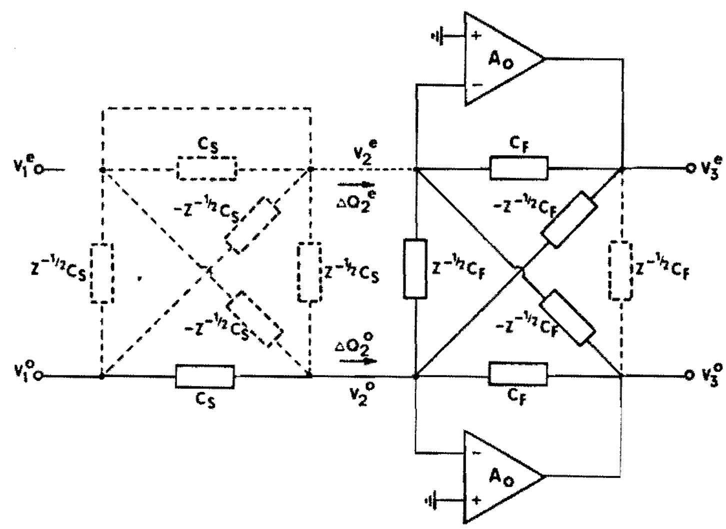
$$H^{oe}(z) = \frac{V_3^e(z)}{V_1^o(z)} = \frac{-z^{-1/2}\ C_s}{(1-z^{-1})\ (1+1/A_o)C_F + C_s/A_o}$$

hetgeen respektievelijk een gedempte inverterende Backward Euler en een gedempte inverterende LDI integrator voorstelt.
In het geval ook de opamp ideaal mag worden verondersteld ($A_0 \rightarrow \infty$) wordt het equivalente circuit van fig.5c verkregen, resulterend in

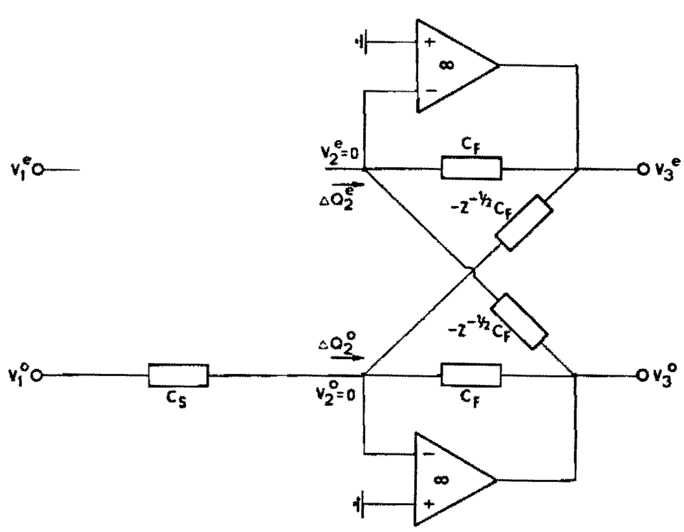
$$H^{oo}(z) = \frac{V_3^o(z)}{V_1^o(z)} = \frac{-C_S}{(1-z^{-1})C_F}$$



A



B



C

fig.5: Voorbeeld analyse m.b.v. equivalente schakeling in het z-domein

a: voorbeeld schakeling

b: equivalente schakeling in het z-domein

c: equivalente schakeling in het z-domein in het geval $A_0 \rightarrow \infty$

$$H^{oe}(z) = \frac{V_3^e(z)}{V_1^o(z)} = \frac{-z^{-1/2}C_S}{(1-z^{-1})C_F}$$

In dit ideale geval zijn de BE en LDI integrator dus ongedempt.
Merk op dat in dit ideale geval, waarvoor $V_2^e = V_2^o = 0$ (virtueel aarde), de ladingsstroom ΔQ_2^o volledig bepaald wordt door V_1^o en C_S volgens

$$\Delta Q_2^o = C_S V_1^o$$

Verder kunnen de spanningen V_3^e en V_3^o uitgedrukt worden in ΔQ_2^o , ΔQ_2^e en C_F:

$$V_3^e = -\frac{1}{(1-z^{-1})C_F}\Delta Q_2^e - \frac{z^{-1/2}}{(1-z^{-1})C_F}\Delta Q_2^o$$

en

$$V_3^o = -\frac{1}{(1-z^{-1})C_F}\Delta Q_2^o - \frac{z^{-1/2}}{(1-z^{-1})C_F}\Delta Q_2^e$$

3.2 ANALYSE M.B.V. SIGNAL FLOW GRAPHS

Bovenstaande relaties kunnen worden gerepresenteerd d.m.v. een Signal Flow Graph [11], zoals geschetst in fig.6.

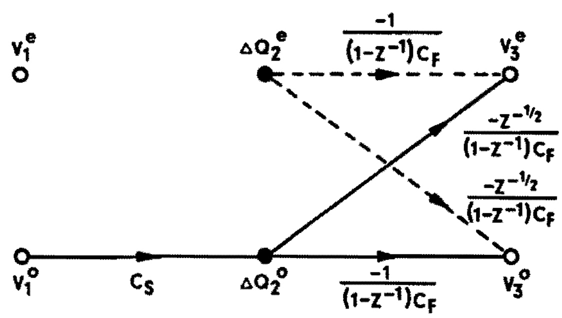


fig.6: Signal Flow Graph voor de schakeling van fig.5a (mits $A_0 \rightarrow \infty$)

Analyse van SC filters m.b.v. SFG's werkt in de praktijk bijzonder plezierig. Nadeel is dat deze methode slechts voor een beperkte klasse van configuraties toepasbaar is, waar gelukkig het overgrote deel van de in de praktijk toegepaste SC filterschakelingen volledig uit zijn opgebouwd. Geëist wordt, dat elk SC deelnetwerk (opgebouwd uit condensatoren en schakelaars) in het totale filter zich bevindt tussen een ideale spanningsbron (bijv. de uitgang van een ideale opamp) en een virtueel aardpunt. Aangezien de ladingsstroom die onttrokken wordt aan een ideale spanningsbron een redundante grootheid is en de spanning op een virtueel aardpunt bekend is, kan de signaal flow in een dergelijk SC netwerk volledig beschreven worden in termen van de spanning over de spanningsbron en de ladingsstroom in het virtueel aardpunt. De relaties tussen deze grootheden kunnen d.m.v. een Signal Flow Graph worden gerepresenteerd.

In de praktijk kan een tabel worden gehanteerd van SC netwerken met hun SFG, waarvan in fig.7 een voorbeeld wordt gegeven.

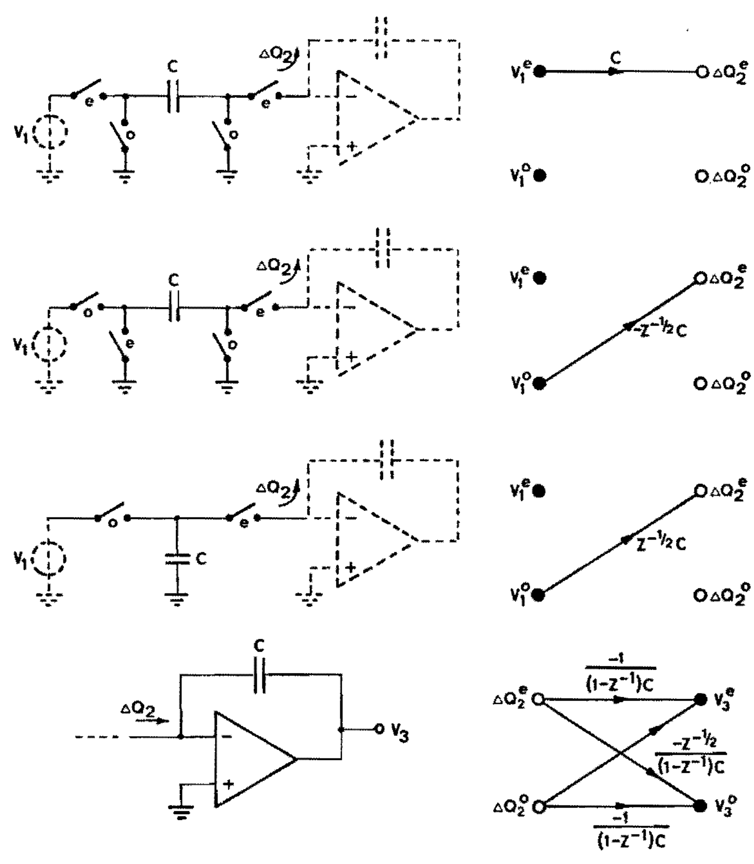


fig.7: Enige SC deelschakelingen met bijbehorende SFG representaties

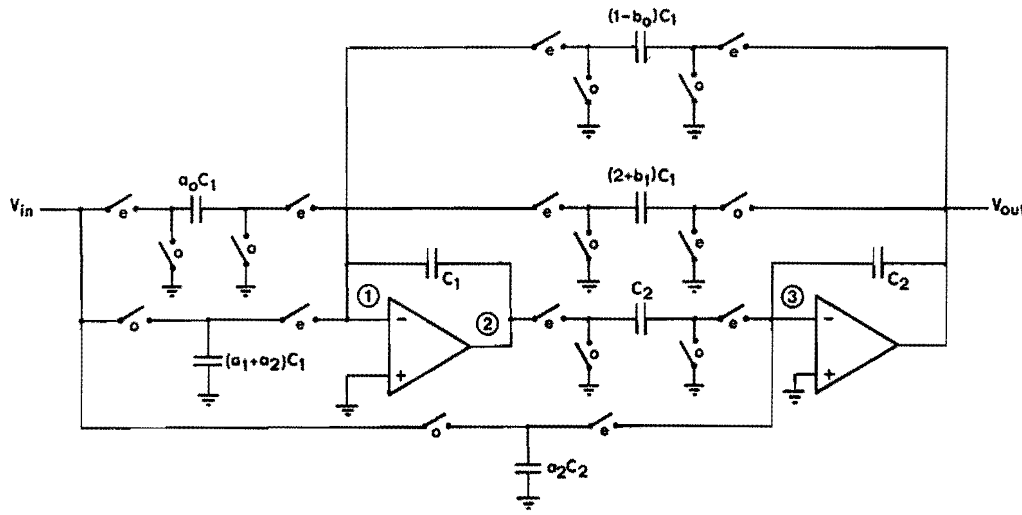


fig.8: Voorbeeld schakeling voor SFG analyse

Door de schakeling van fig.8 op te delen in subcircuits, kan gebruikmakend van de tabel van fig.7 de bijbehorende SFG worden afgeleid, zoals weergegeven in fig.9.

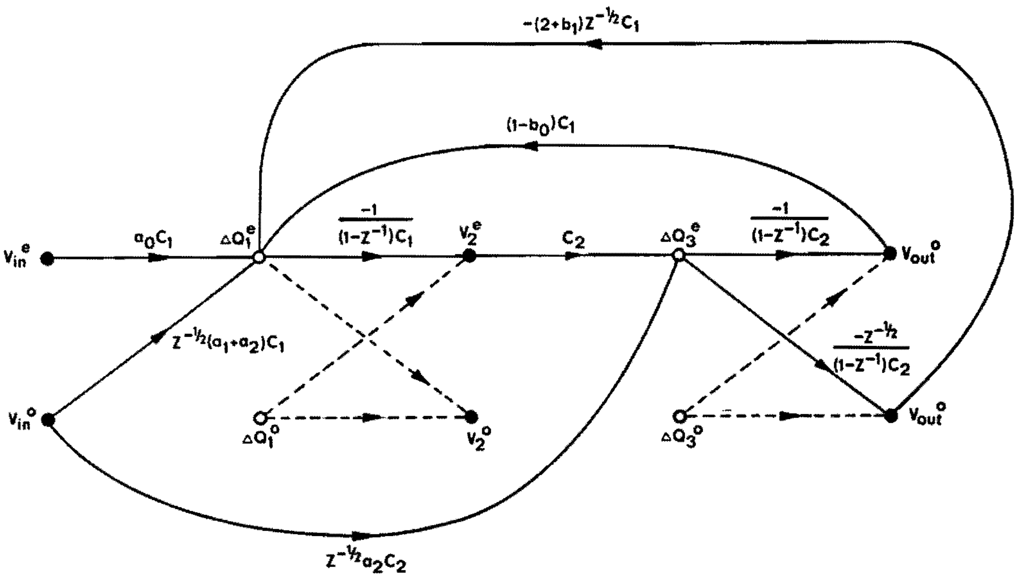


fig.9: Signal Flow Graph, behorend bij de schakeling van fig.8

Uit deze graph kan eenvoudig [8-9] de overdracht van het filter worden afgeleid:

$$H^{ee} = \frac{V^{e}_{out}}{V^{e}_{in}} = \frac{a_0}{b_0 + b_1 z^{-1} + z^{-2}}$$

$$H^{eo} = \frac{V^{o}_{out}}{V^{e}_{in}} = z^{-1/2}\, H^{ee}$$

$$H^{oe} = \frac{V^{e}_{out}}{V^{o}_{in}} = z^{-1/2}\, \frac{a_1 + a_2 z^{-1}}{b_0 + b_1 z^{-1} + z^{-2}}$$

$$H^{oo} = \frac{V^{o}_{out}}{V^{o}_{in}} = z^{-1/2}\, H^{oe}$$

Als het ingangssignaal niet verandert gedurende een volledige klokperiode, zodanig dat $V^{o}_{in} = z^{-1/2}\, V^{e}_{in}$, dan geldt

$$H^{ee} = z^{1/2}\, H^{eo} = z^{-1/2}\, \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{b_0 + b_1 z^{-1} + z^{-2}}$$

4. SYNTHESE VAN SC FILTERS

Voor de synthese van SC filters staan de ontwerper verschillende methodieken ter beschikking, waarvan er twee zeer veelvuldig worden toegepast:

- synthese gebaseerd op het in cascade schakelen van 2^{e} orde sekties (biquads),
- leap-frog synthese.

4.1 SYNTHESE MET GECASCADEERDE SC BIQUADS

Bij deze synthesemethode worden allereerst van de te realiseren overdrachtsfunktie teller en noemer ontbonden in tweede orde faktoren (plus evt. een eerste orde faktor).

Vervolgens wordt gezocht naar een optimale pool-/nulpuntpaar combinatie per sektie. Aspecten die hierbij een rol kunnen spelen zijn bijv. gevoeligheid van het totale filter voor componentvariaties, totale benodigde capaciteit, dynamisch bereik. De keuze wordt o.a. bepaald door de topologie van de biquad sektie, die sterk van invloed is op de samenhang van pool- en nulpuntpaar. Bij de keuze van de biquad sektie kan evt. gebruik gemaakt worden van een catalogus, bijv. [12].

Nu kunnen de filtersekties gedimensioneerd worden, waarbij veelal een aantal vrijheidsgraden aanwezig zijn, die later weer gebruikt kunnen worden voor het optimaliseren van de dynamiek en het minimaliseren van de capaciteit.

Daarna wordt de optimale volgorde bepaald om de filtersekties in cascade te schakelen. Deze keuze kan sterk van invloed zijn op de dynamiek van het filter.

Vervolgens worden per opamp *uitgang* alle daarmee verbonden capaciteiten met een zodanige factor geschaald, dat elke opamp bij hetzelfde spanningsniveau aan de ingang van het filter zal verzadigen.

Als laatste stap worden per opamp *ingang* alle daarmee verbonden capaciteiten met een zodanige factor geschaald, dat de kleinste capaciteit, verbonden met die ingang, gelijk is aan de kleinste voldoend nauwkeurig realiseerbare waarde.

4.2 LEAP-FROG SYNTHESE

Bij de leap-frog synthese wordt geprobeerd de extreem goede gevoeligheidseigenschappen van tweezijdig ohms afgesloten LC ladderfilters te behouden bij het omzetten naar een SC struktuur.

Allereerst wordt gezocht naar een prototype LC ladderfilter dat aan de eisen voldoet. Vervolgens worden voor dit filter om en om de knooppuntspanningen in de takstromen uitgedrukt en vice versa. Dit wordt op een dusdanige wijze gedaan, dat uitsluitend uitdrukkingen in integraalvorm worden verkregen. Deze uitdrukkingen worden vervolgens geïmplementeerd d.m.v. een SC structuur, opgebouwd uit gekoppelde integratoren.

Het prototype laagdoorlaatfilter van fig.10 levert bijv. volgend

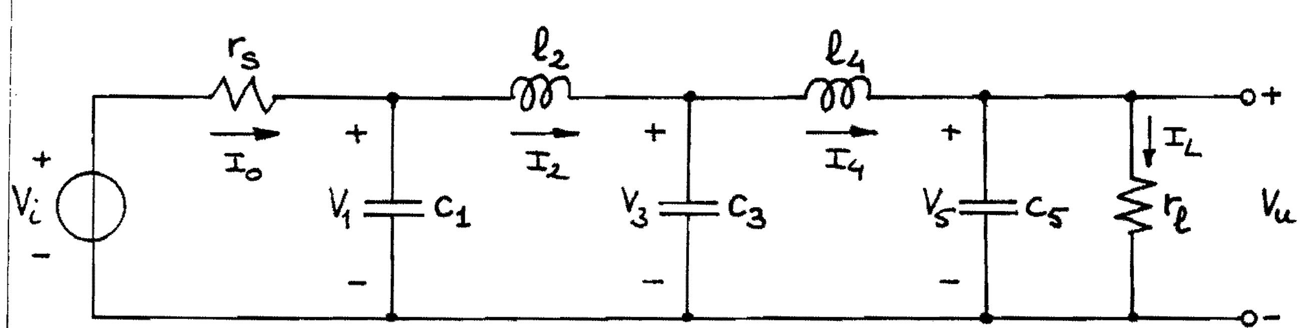


fig.10: Prototype LC ladder laagdoorlaatfilter

stelsel vergelijkingen op:

$$V_1 = \frac{I_0 - I_2}{sc_1} \qquad V_3 = \frac{I_2 - I_4}{sc_3} \qquad V_5 = \frac{I_4 - I_1}{sc_5} \qquad V_u = V_5$$

$$I_0 = \frac{V_i - V_1}{r_s} \qquad I_2 = \frac{V_1 - V_3}{sl_2} \qquad I_4 = \frac{V_3 - V_5}{sl_4} \qquad I_1 = \frac{V_u}{r_1}$$

Dit stelsel kan worden gerepresenteerd door de graph van fig.11a, of na schaling van alle stroomknooppunten met een factor $r_s = r_1 = r$ door die van fig.11b.

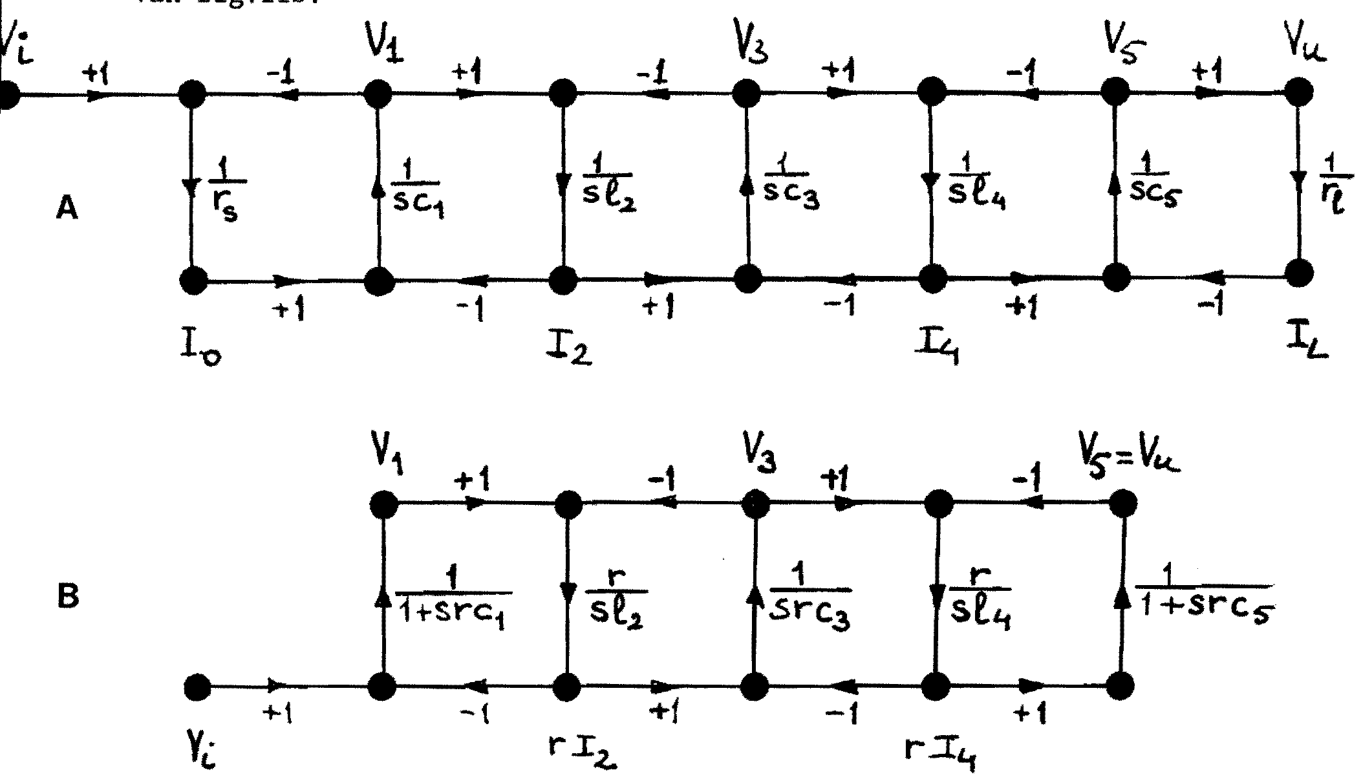


fig.11: Graph behorend bij fig.10

a: vóór schaling b: ná schaling van de stroomknooppunten

16

Alle vertikale takken stellen integratoren voor, waarbij alleen de buitenste gedempt zijn.

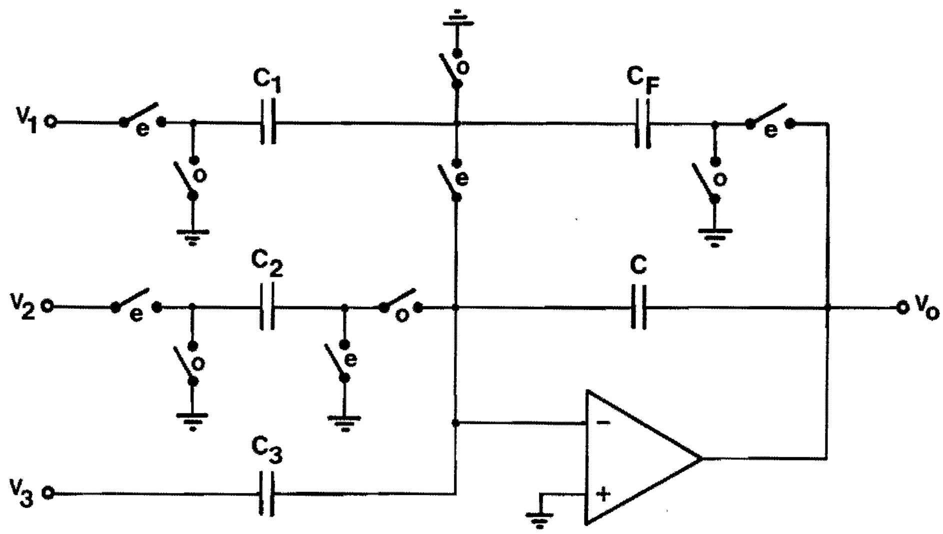


fig.12: Universele SC integratorschakeling

Bij de SC implementatie wordt als universele integratorschakeling het circuit van fig.12 gebruikt, met overdracht

$$V_0 = \frac{-C_1 z^{1/2} V_1 + C_2 z^{-1/2} V_2 - C_3 (z^{1/2} - z^{-1/2}) V_3}{C(z^{1/2} - z^{-1/2}) + C_f z^{1/2}}$$

(alle spanningen gesampled in klokfase "e")

of met $\gamma := \dfrac{z^{1/2} - z^{-1/2}}{2} \approx s$ (bij klokperiode genormeerd op 2)

en $\mu := \dfrac{z^{1/2} + z^{-1/2}}{2} \approx 1$

$$V_0 \approx \frac{-C_1 z^{1/2} V_1 + C_2 z^{-1/2} V_2 - s2C_3 V_3}{s(2C+C_f) + C_f}$$

Door op de circuitvergelijkingen de volgende substituties toe te passen:

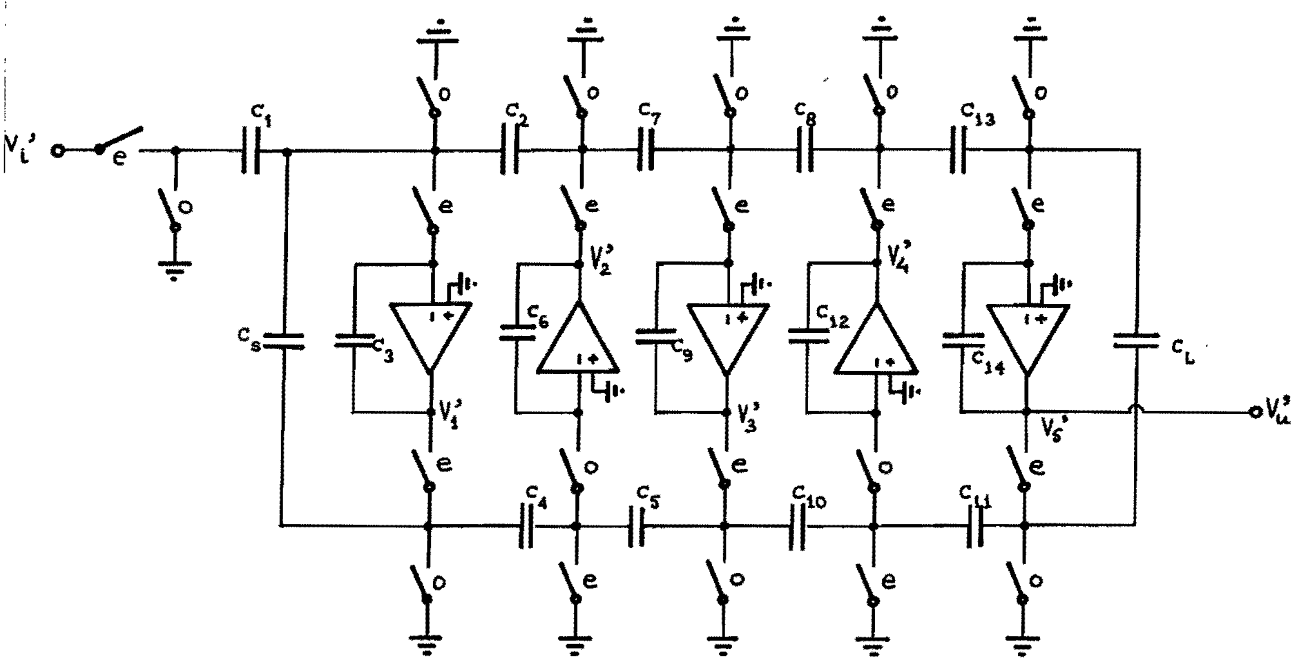


fig.13: SC leap-frog filter

$V_i' = -V_i \qquad V_1' = z^{1/2}V_1 \qquad V_2' = rI_2$

$V_3' = -z^{1/2}V_3 \qquad V_4' = -rI_4 \qquad V_5' = z^{1/2}V_5$

wordt het volgende stelsel verkregen:

$$V_1' = \frac{-z^{1/2}C_1V_i' - z^{1/2}C_2V_2'}{s(2C_3+C_s)+C_s} \qquad \text{met} \qquad \begin{aligned} C_1 &= C_2 = C_s = 1 \\ C_3 &= (rc_1-1)/2 \end{aligned}$$

$$V_2' = \frac{z^{-1/2}C_4V_1' + z^{-1/2}C_5V_3'}{s2C_6} \qquad \begin{aligned} C_4 &= C_5 = 1 \\ C_6 &= \frac{l_2}{2r} \end{aligned}$$

$$V_3' = \frac{-z^{1/2}C_7V_2' - z^{1/2}C_8V_4'}{s2C_9} \qquad \begin{aligned} C_7 &= C_8 = 1 \\ C_9 &= \frac{rc_3}{2} \end{aligned}$$

$$V_4' = \frac{z^{-1/2}C_{10}V_3' + z^{-1/2}C_{11}V_5'}{s2C_{12}} \qquad \begin{aligned} C_{10} &= C_{11} = 1 \\ C_{12} &= \frac{1_4}{2r} \end{aligned}$$

$$V_5' = \frac{-z^{1/2}C_{13}V_4'}{s(2C_{14}+C_1) + C_1} \qquad \begin{aligned} C_{13} &= C_1 = 1 \\ C_{14} &= (rc_5-1)/2 \end{aligned}$$

hetgeen m.b.v. de schakeling van fig.12 als bouwblok kan worden gerealiseerd, zie fig.13.

Net als bij de beschreven synthesemethode met SC biquads kan de dynamiek worden geoptimaliseerd door per opamp *uitgang* alle daarmee verbonden capaciteiten te schalen. Ook kan hier vervolgens de totale capaciteit geminimaliseerd worden door weer per opamp *ingang* alle daarmee verbonden capaciteiten te schalen.

LITERATUUR

[1] A. Fettweis, "Theory of resonant-transfer circuits and their application to the realization of frequency selective networks," Proc. Summer School on Circ. Th., Czechoslovak Academy of Sciences, 1968.

[2] D.L. Fried, "Analog sample-data filters," IEEE J. Solid-State Circuits, vol. SC-7, pp. 302-304, Aug. 1972.

[3] M.S. Ghausi, en K.R. Laker, "Modern filter design - active RC and switched capacitor," Englewood Cliffs: Prentice-Hall, 1981.

[4] P.E. Allen, en E. Sánchez-Sinencio, "Switched capacitor circuits," New York: Van Nostrand Reinhold, 1984.

[5] G.S. Moschytz (ed.), "MOS switched-capacitor filters: analysis and design," New York: IEEE Press, 1984.

[6] J.A. Hegt, "Filters met geschakelde capaciteiten," diktaat 5696 en 5697, Technische Universiteit Eindhoven, 1987.

[7] J.A. Hegt, "Contributions to switched capacitor filter synthesis," proefschrift Technische Universiteit Eindhoven, 1988.

[8] S.J. Mason, "Feedback theory - some properties of signal flow graphs," Proc. IRE, vol. 41, pp. 1144-1156, Sept. 1953.

[9] S.J. Mason, "Feedback theory - further properties of signal flow graphs," Proc. IRE, vol. 44, pp. 920-926, July 1956.

[10] K.R. Laker, "Equivalent circuits for the analysis and synthesis of switched capacitor networks," Bell Syst. Tech. J., 58, pp. 727-767, March 1979.

[11] G.S. Moschytz, en U.W. Brugger, "Signal-flow graph analysis of SC networks," Proc. Inst. Elec. Eng., vol. 131, pt. G, pp. 72-85, Apr. 1984.

[12] J.C.M. Bermudez, en B.B. Bhattacharyya, "A systematic procedure for generation and design of parasitic insensitive SC biquads," IEEE Trans. Circuits Syst., vol. CAS-32, no. 8, pp. 767-783, Aug. 1985.

DIGITALE SIGNAALBEWERKING

deel II.8

Spraaksynthese

door

L.L.M. Vogten

IPO Eindhoven

Inhoudsopgave

Lineaire Predictie LPC: Toepassing van digitale adaptieve filters bij spraaksynthese

L.L.M. Vogten, IPO-Eindhoven

De techniek van digitale signaalbewerking speelt ook in het spraakonderzoek een belangrijke rol. Toepassingen vinden we o.m. bij de ontwikkeling van systemen voor spraaksynthese, waarbij we ons ten doel stellen om goed verstaanbare en natuurlijk klinkende spraak op te wekken zonder tussenkomst van een menselijke stem. In deze bijdrage aan de PATO-cursus "Digitale Signaalbewerking" zullen we laten zien hoe adaptieve filters (v.d. Enden & Verhoeckx, 1987, hfdst. 7.6) in de spraaksynthese worden toegepast via de techniek van "Linear Predictive Coding", LPC.

Spraakgeluid kan worden gemodelleerd door een signaal dat tot stand is gekomen door een bronsignaal op een (onbekende) manier te filteren. Dit is het zogeheten bron-filtermodel voor spraakproduktie en de parameters van dat model variëren in de tijd. De LPC-techniek is een methode om de coëfficiënten van het adaptieve filter van dat model automatisch uit het spraakgeluid te bepalen als funktie van de tijd.

De indeling van deze syllabus is als volgt. Eerst geven we in het kort aan hoe de produktie van spraakgeluid fysisch beschreven kan worden met een lineair bron-filtermodel. Dit model, in de vorm van een spraaksynthetisator met meerdere tweede-orde deelfilters blijkt zeer geschikt te zijn voor het spraakonderzoek. Vervolgens zetten we uiteen hoe we de parameters van het model, dus de stuurgetallen van de spraaksynthetisator, automatisch uit natuurlijk spraakgeluid kunnen bepalen via de techniek van Lineaire Predictie. We besluiten met een illustratie van enkele toepassingen van deze LPC-techniek van analyse en (re)synthese ten behoeve van manipulatie en zuinige codering van spraak.

1 Bron-filtermodel voor de spraakproduktie

Om te fysisch beschrijven hoe een spraakgeluid gemaakt wordt onderscheiden we een *geluidsbron*, waardoor het *brongeluid* wordt opgewekt, en een *filter*, waardoor dat brongeluid wordt gekleurd. Er zijn in normale spraak twee typen brongeluiden. Het eerste type is periodiek en ontstaat door het trillen van de stembanden bij het maken van klinkergeluiden en van medeklinkers als bijv. /m/, /n/, /w/ en /j/. Het tweede type brongeluid is ruis die in de mondholte ontstaan doordat de luchtstroom uit de longen door een sterke vernauwing wordt geperst. Dit ruisgeluid treedt op bij medeklinkers als /f/, /s/, /p/ en /t/.

Het filter, dat zorgt dat het spraakgeluid het gewenste timbre of de gewenste 'kleur' krijgt, wordt gevormd door dat deel van het spraakkanaal dat zich tussen de geluidsbron en de buitenlucht bevindt. Dus wanneer de geluidsbron de stembandtrilling in het strottehoofd is, wordt het verkleurende filter gevormd door het hele spraakkanaal, de mond-keelholte (en bij nasale geluiden tevens de neusholte) samen. Wanneer de geluidsbron zich in de mondholte bevindt, zoals bij de /k/ en de /t/, wordt het brongeluid verkleurd door de werking van de holte die zich tussen de afsluiting en de buitenlucht bevindt. Wanneer het brongeluid gemaakt wordt met een vernauwing of afsluiting tussen de lippen (die onmiddellijk aan de buitenlucht grenzen), vindt geen of weinig verkleuring plaats. Het filter dat gevormd wordt door het spraakanaal werkt als een resonator; bepaalde frekwenties worden door het filter versterkt, andere worden verzwakt. Dit is in Fig. 1 schematisch weergegeven. Daar komt nog bij het effekt van de uitstraling aan de mond-

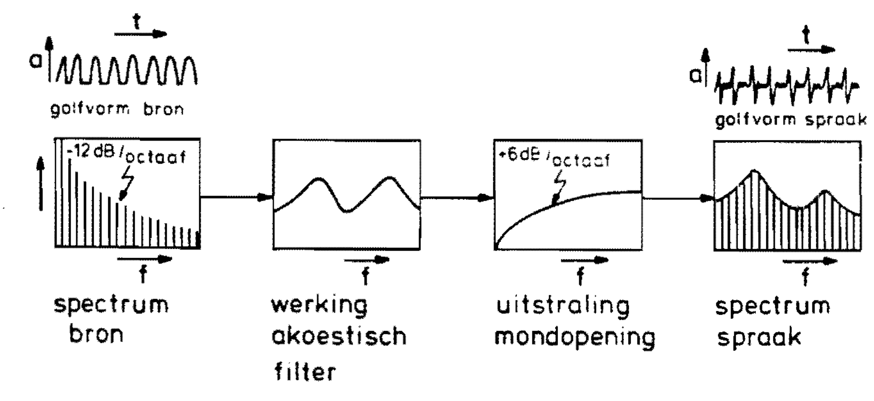


Fig. 1: Schematische voorstelling van het tot stand komen van een klinkergeluid volgens de bronfiltertheorie van spraakproduktie.

opening, waardoor de spectrale omhullende met ongeveer 6 dB afneemt als de frekwentie wordt gehalveerd. Het eindresultaat is het spectrum van b.v. het klinkergeluid. We zien de herhalingsfrekwentie F_0 van de stembandtrilling terug in de afstand tussen de boventonen in het spectrum; ze liggen verder

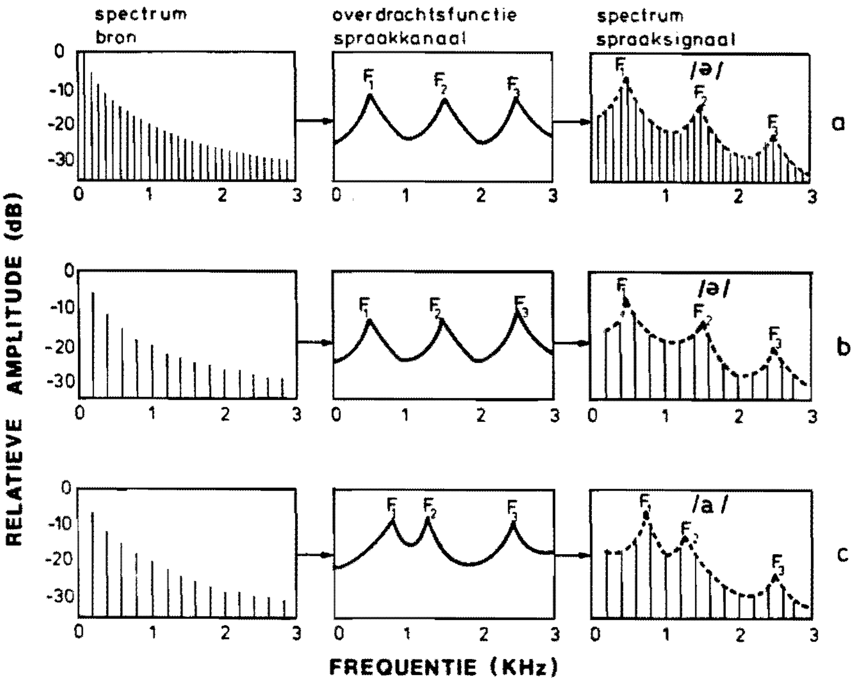


Fig. 2: Drie voorbeelden van het tot stand komen van een klinkergeluid: a) de klinker /ə/ met een grondtoon van 100 Hz; b) idem met een grondtoon van 200 Hz en c) de klinker /a/ met een grondtoon van 200 Hz.

uit elkaar naarmate de grondtoon een hogere frekwentie heeft. Het effekt van de filterende werking van het spraakkanaal zien we vooral terug in de plaatsen van de pieken in de spectrale omhullende. Deze plaatsen hangen samen met de vorm van het akoestisch filter en zijn dus afhankelijk van de stand van de spraakorganen.

De bijdragen van brongeluid, filterwerking en stralingseffekt worden in dit model als onafhankelijke elementen beschouwd, die elkaar niet beïnvloeden (belasten). We hebben deze bijdragen nog eens schematisch weergegeven in Fig. 2. In Fig. 2a heeft het brongeluid een grondtoon van 100 Hz, dat is de grondtoon van een lage mannenstem. De mondkeelholte staat in de stand voor een /ə/, de klank die hoort bij de onbeklemtoonde "e" in een woord als "bedaard", waardoor de filterwerking pieken in het spectrum veroorzaakt die liggen bij 500, 1500 en 2500 Hz, de resonantiefrekwenties van het filter.

In dit geval gedraagt het spraakkanaal zich als een buis die aan een kant open en aan de andere kant gesloten is; te vergelijken met een orgelpijp. In het (geïdealiseerde) geval dat de buis volkomen star is (geen akoestische energie absorbeert) en bij verwaarlozing van het stralingseffekt aan het open eind, gedraagt de buis zich als een perfekte kwart-golflengte-resonator. De resonantiefrekwenties zijn dan betrekkelijk eenvoudig te berekenen. Zij zijn alleen afhankelijk van de lengte L van de buis en van de geluidssnelheid c en gegeven door oneven veelvouden van c/4L. Bij een volwassen man is L ongeveer 0.17 m zodat bij een geluidssnelheid van 340 m/s de resonanties liggen bij 500, 1500, 2500 Hz enz. Bij andere klinkers dan de /ə/ gedraagt het spraakkanaal zich niet als één buis maar als een samenstel van twee of meer buizen van verschillende doorsnede. De berekening van de resonantiefrekwenties wordt dan ingewikkelder.

In Fig. 2b laten we de mondkeelholte in dezelfde stand als in Fig. 2a, maar laten we de stembanden trillen met een tweemaal zo hoge herhalingsfrekwentie, waardoor we een grondtoon krijgen van 200 Hz, en ook de boventonen steeds 200 Hz uit elkaar liggen in het spectrum. Het spraakgeluid is nog steeds een /ə/, met resonanties bij 500, 1500 en 2500 Hz, maar de toonhoogte is tweemaal zo hoog. In Fig. 2c tenslotte houden we de grondtoon vast op 200 Hz maar we veranderen de stand van de mondkeelholte, waardoor we resonantiefrekwenties krijgen in de buurt van 800, 1200 en 3200 Hz. Het geluid klinkt dan als een /a/, met een hoge toonhoogte.

2 Spraaksynthetisator met tweede-orde deelfilters

In het bron-filtermodel wordt de resulterende overdrachtsfunktie gegeven door het produkt van de afzonderlijke bijdragen. Omdat die lineair en onafhankelijk zijn mogen we ze als één geheel opvatten, waarmee een eenvoudige spraaksynthetisator ontstaat zoals weergegeven in Fig. 3.

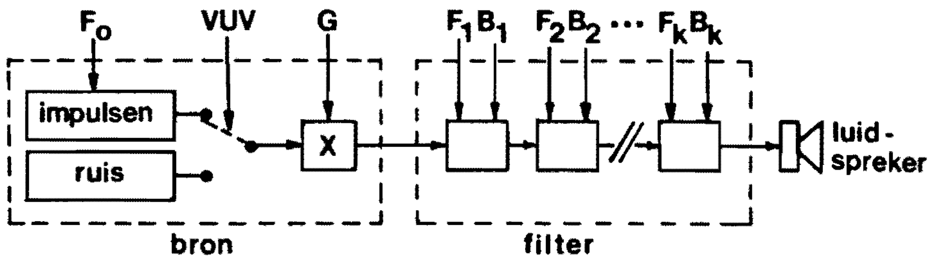


Fig. 3: Vereenvoudigd bron-filtermodel voor de produktie van spraakgeluid. Het bronsignaal wordt bepaald door drie parameters: de herhalingsfrekwentie F_0 van de impulsen, de stem/stemloosparameter VUV en de amplitudeversterkingsfaktor G. Het filter is samengesteld uit een cascade van tweede-orde deelfilters. De parameters van ieder filter zijn aangeduid met afstemfrekwenties F_k en bandbreedtes B_k.

Hierin neemt één (hogere orde) filter de volgende bijdragen voor zijn rekening:

1. het effekt van de afnemende hogere harmonischen van de stembandtrilling, overeenkomend met een, voor stemhebbende klanken vaste, overdrachtsfunktie van -12 dB/oct,

2. het effekt van de variabele akoestische eigenschappen van het spraakkanaal en

3. het effekt van de straling aan de mondopening, overeenkomend met een vaste overdrachtsfunktie van +6 dB/oct.

In dit (sterk vereenvoudigde) bronfiltermodel heeft het bronsignaal dus *altijd* een vlak, 'wit' energiespectrum, dat bij stemhebbende klanken bestaat uit een periodieke impuls met herhalingsfrekwentie F_0 en bij stemloze klanken bestaat uit witte ruis. In de tijd wordt het bronsignaal gespecificeerd door drie parameters: ruisbron of peridodiekbron (voiced/unvoiced, VUV); indien periodiek dan de herhalingsfrekwentie F_0 van de impulsen en de versterkingsfaktor G waarmee de amplitude van de impuls en de ruis versterkt wordt. Voor het filter is hier gekozen voor een cascade van tweede-orde deelfilters waarbij één tweede-orde filter één resonantiepiek voor zijn rekening kan nemen.

Ieder deelfilter heeft een instelbare afstemfrekwentie F (afstemfrekwentie) en een instelbare bandbreedte B. Met de afstemfrekwentie kunen we regelen in welk frekwentiegebied de componenten doorgelaten worden. Door de bandbreedte te wijzigen kunnen we de steilheid van de spectrale flanken van de resonantie besturen: een kleine bandbreedte geeft een zeer scherpe spectrale piek met hoge amplitude; een grote bandbreedte een slappe piek, zoals aangegeven is in Fig. 4. Naast de bandbreedte zullen we in het vervolg

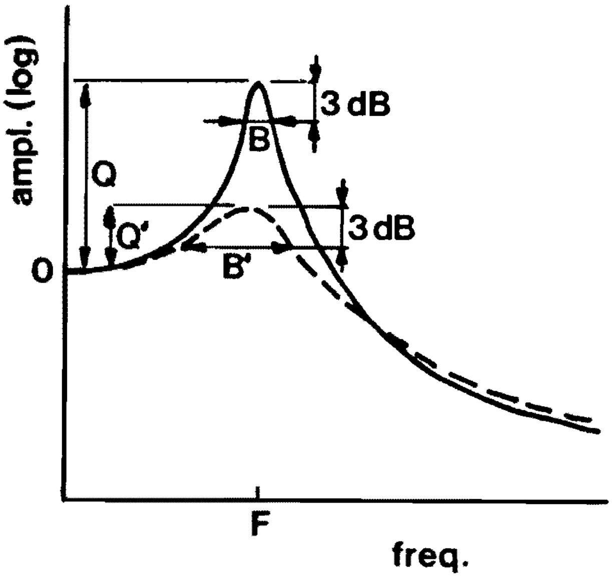


Fig. 4: De invloed van de bandbreedte op het amplitudeverloop van de overdrachtsfunktie van een tweede-orde filter. Bij vergroten van de 3 dB-bandbreedte van B naar B' (of verkleinen van de kwaliteitsfaktor Q=F/B naar Q'=F/B') neemt de piekwaarde (selectiviteit) van de amplitude af en gaat de doorgetrokken curve over in de gestippelde.

ook gebruik maken van de *kwaliteitsfaktor* Q van het filter, hier gedefiniëerd door de verhouding F/B. Deze Q is een (wat meer direkte) maat voor de selektiviteit van het filter.

Deze spraaksynthetisator is het gereedschap waarmee we nu kunnen onderzoeken wat het verband is tussen spraakgeluiden en de spraakklanken die we daarin herkennen. Hij stelt ons in staat om, onafhankelijk van elkaar de volgende waarneembare eigenschappen van geluid te bestuderen:

1. De geluidssterkte en daarmee tevens de aan- of afwezigheid van geluid. Deze wordt bestuurd door de amplitudeversterkingsfaktor G van het brongeluid te veranderen.

2. Het wel of niet *stemhebbend (tonaal)* zijn van het spraakgeluid. Dit wordt bepaald door de keuze van de geluidbron VUV. Bij de hier gekozen synthetisator is niet voorzien in een kombinatie van periodiek en ruizig brongeluid, zodat stemhebbende wrijf- en plofklanken er in principe niet goed mee gemaakt kunnen worden. In de praktijk blijkt deze beperking geen groot bezwaar te zijn.

3. Voor stemhebbende geluiden, de toonhoogte van het geluid. Deze wordt bestuurd via de grondtoonfrekwentie F_0 van het periodieke brongeluid.

4. Het timbre of klankkleur van het geluid. Hiervoor zijn de gezamenlijke afstemfrekwenties en bandbreedtes $/F_k, B_k/$ van de deelfilters verantwoordelijk.

De hier geschetste spraaksynthetisator met 5 deelfilters heeft dus 13 stuurgetallen nodig. Wat we nog niet gespecificeerd hebben is de duur van het spraakgeluid dat gesynthetiseerd wordt. In principe kunnen we deze vrij kiezen, maar vaak willen we een spraakfragment synthetiseren waarvan de eigenschappen in de tijd veranderen. Zulke veranderingen worden in de synthese benaderd door korte stukjes spraakgeluid, gespecificeerd door de hierboven genoemde 13 getallen, achter elkaar te zetten en de specificatie van ieder volgend stukje spraakgeluid te veranderen. In de praktijk blijkt het goed te werken wanneer we de waarden van de getallen 100 keer per seconde veranderen. De opeenvolgende stukjes spraakgeluid waarvan we de eigenschappen dus steeds met de stuurgetallen bijstellen, hebben dus een duur van 10 milliseconden. De vraag is nu hoe we aan de stuurgetallen voor de spraaksynthetisator komen. Om die getallen (de modelparameters) te bepalen kunnen we natuurlijke spraak analyseren. De methode van Lineaire Predictie geeft, zoals we in de volgende sectie zullen zien, die mogelijkheid.

3 LPC-analyse van spraakgeluid

Bij de analyse van spraaksignalen is het de bedoeling om de parameters van het bron-filtermodel te bepalen als funktie van de tijd. Daarbij gaat het om een beschrijving van de energie (geluidssterkte), de spectrale omhullende (klankkleur, timbre), het wel of niet periodiek zijn (stemhebbend/stemloos) en, indien periodiek, de frekwentie F_0 van de grondtoon (toonhoogte). De analyse vindt steeds plaats over een relatief kort segment (het analysevenster), in grootte variërend van 20 tot 40 ms, afhankelijke van de te bepalen parameter. Daarbij nemen we dus aan dat de betreffende parameter binnen dit venster constant is, dus dat het signaal (lokaal) stationair is. Alvorens nu de eigenlijke LPC-analyse te bespreken gaan we eerst even in op een van de basisbegrippen van die analyse: de discrete autocorrelatiefunktie.

3.1 Autocorrelatiefunktie

Basis voor de LPC-analyse vormt de zogeheten *auto-correlatiefunktie* R_k, gedefiniëerd als:

$$R_k = \sum_{n=0}^{N} s_n s_{n-k} \tag{1}$$

Hierin is s_n de verkorte schrijfwijze voor de waarde van het signaal op de discrete tijdstippen n/f_s, waarin f_s de bemonsteringsfrekwentie is en n een geheel getal. Dus s_n is de n^e bemonstering ("sample") en evenzo is s_{n-k} het $(n-k)^e$ "sample", dus het signaal s over k samples verschoven t.o.v. s_n. Voor het berekenen van de autocorrelatie volgens (1) wordt dus het signaal s_n met zichzelf vermenigvuldigd, na eerst over k samples te zijn verschoven, en dan gesommeerd over de N samples binnen het analysevenster. Daarmee verkrijgen we de k-de autocorrelatie en als funktie van de verschuiving k de autocorrelatiefunktie R_k.

Van belang voor de spraakanalyse is in welk gebied van de verschuiving k we de autocorrelatiefunktie berekenen. In het algemeen bevatten de eerste M autocorrelaties $R_1, R_2, \ldots R_M$ informatie omtrent de *omhullende* van het amplitude- (of energie-)spectrum. Hoe meer autocorrelaties we meenemen, dus hoe groter M, des te groter is het "oplossend vermogen" Δf voor details in die omhullende, waarbij globaal het verband geldt:

$$\Delta f = \frac{f_s}{1+2M}, \tag{2}$$

waarin f_s de bemonsteringsfrekwentie is.

Voor het gebied waarover de tijdverschuiving k laten lopen kunnen we globaal drie gebieden onderscheiden. Eerst het geval dat we $k = 0$ nemen. Dan krijgen we R_0, de *energie* (kwadratensom) van het spraaksignaal binnen het analysevenster. Als tweede het gebied van de "korte-termijn" autocorrelatie, waarbij k kleiner is dan, zeg, 20 samples. Met bijv. $M = 10$ leveren de eerste 10 autocorrelaties dan informatie over de spectrale omhullende met een oplossend vermogen van ongeveer 500 Hz, als $f_s = 10$ kHz. Dit is het gebied, met M kleiner dan bijv. 20, dat meestal bij de LPC analyse wordt gebruikt. Het derde gebied bereiken we als we tijdverschuiving k veel groter kiezen, bijv. tussen 20 en 200 samples. We spreken dan van een "lange-termijn" autocorrelatie en krijgen daarmee informatie over de *fijnstruktuur* van het amplitudespectrum. Voor periodieke signalen vinden we een top in de autocorrelatiefunktie op veelvouden van de periode $T_0 = 1/F_0$ van de grondtoon. Het signaal lijkt dan sterk op het over T_0 verschoven signaal. We zien hiervan een voorbeeld in Fig. 5, waarin de periode ongeveer 9

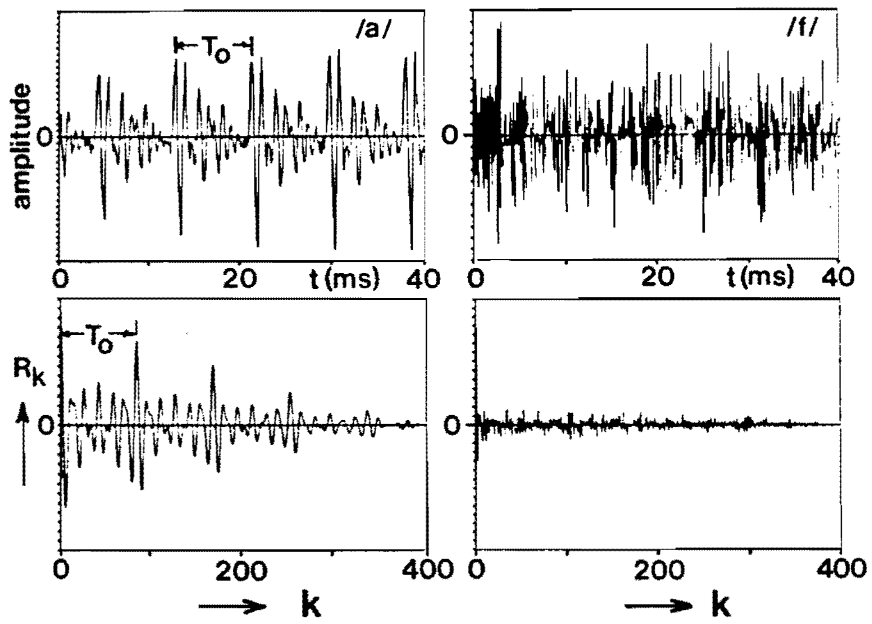


Fig. 5: Voorbeeld van golfvorm (boven) voor de klinker /a/ (links) en medeklinker /f/ (rechts) met bijbehorende autocorrelatiefunkties (onder). We zien bij het periodieke signaal (links) in de correlatiefunktie pieken op veelvouden van de grondtoonperiode, hier ongeveer 9 ms.

ms bedraagt. Voor niet-periodieke signalen ontbreken dergelijke duidelijke pieken. De autocorrelatiefunktie zou dus in principe ook kunnen dienen als basis voor zowel de stemhebbend/stemloos-beslissing als voor het meten van de toonhoogte. In de praktijk echter kan alleen de stem/stemloos-beslissing op redelijk bevredigende wijze op de autocorrelaties worden gebaseerd en met name op de genormeerde eerste autocorrelatie, gedefiniëerd door R_1/R_0, dus de verhouding van eerste autocorrelatie en energie. We komen ook hierop nog terug bij de LPC-analyse.

3.2 LPC-analyse

De methode van Lineaire Predictie kan in het frekwentiedomein worden opgevat als een beschrijving van de spectrale omhullende van spraakgeluid door een klein aantal continu afstembare filters, ieder met een variable bandbreedte. Deze methode leent zich daardoor zeer goed om automatisch de parameters van het bron-filtermodel uit het spraaksignaal te berekenen. Zij levert direkt een beschrijving van de spectrale omhullende in termen van één hogere-orde filter en de amplitude-versterkingsfaktor G. Dat hogere-orde filter kan vervolgens worden omgezet in een cascade van tweede-orde deelfilters.

We hebben eerder gezien hoe het filter in het vereenvoudigde bronfiltermodel de spectrale omhullende bepaalt van het uitgangssignaal. De overdrachtsfunktie van dit filter (verder ook synthesefilter genoemd) bepaalt de omzetting van het vlakke, witte bronspectrum naar het gekleurde, gepiekte spectrum aan de uitgang van het model. Dat het bronspectrum wit is levert ons nu de mogelijkheid om voor een bepaald stukje spraakgeluid, via de techniek van zogeheten "invers filteren", de parameters van het filter rechtstreeks uit het spraaksignaal te bepalen. Stel dat we een signaal met een of ander analysefilter zódanig filteren dat aan de uitgang van dat filter het spectrum een vlakke omhullende heeft, dus wit is. Dan moet de overdrachtsfunktie van dat analysefilter de geïnverteerde zijn van het synthesefilter dat we zoeken, immers dit laatste heeft een wit ingangsspectrum. Dit betekent dat we de parameters van het synthesefilter kunnen vinden door de overdrachtsfunktie van het analysefilter te inverteren; analyse- en synthese-filter zijn elkaars inversen. Hoe we de parameters van het analysefilter zó berekenen dat het spectrum aan de uitgang ervan vlak wordt zullen we nu uiteenzetten.

3.3 Bepaling van de a-parameters van het analysefilter

We nemen aan dat het analysefilter fysisch realiseerbaar (dus causaal) is en dat het lineair en tijdonafhankelijk is voor een korte tijdsduur overeenkomend met de grootte van het analysevenster (b.v. 25 ms). Verder dat het een digitaal filter is waarvan de overdrachtsfunktie louter nulpunten heeft. Voor zo'n filter wordt in het tijddomein het uitgangssignaal e_n (dat is het uitgangssignaal op de tijdstippen n/f_s waarin f_s de bemonsteringsfrekwentie is en n een geheel getal) alleen bepaald door het ingangssignaal s_n op datzelfde tijdstip en een *lineaire combinatie* van M voorafgaande samples aan de ingang:

$$e_n = s_n + \sum_{k=1}^{M} a_k s_{n-k} = \sum_{k=0}^{M} a_k s_{n-k}, \qquad a_0 = 1. \tag{3}$$

Het uitgangssignaal e_n is dus de convolutie van het ingangssignaal s_n en de impulsresponsie gegeven door de coëfficiënten a_k. Alleen de M voorafgaande samples uit het verleden dragen, elk voorzien van een eigen weegfaktor a_k, bij tot het uitgangssample e_n, zie Fig. 6. De orde M van het filter geeft het

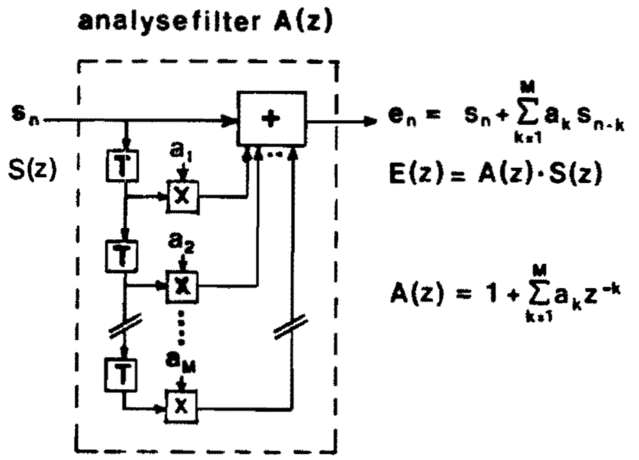


Fig. 6: Digitaal M^e-orde analysefilter. Het uitgangssample e_n op tijdstip nT wordt gegeven door de som van het ingangssample s_n op datzelfde tijdstip en M gewogen daaraan voorafgaande ingangssamples. De weegfaktoren a_k worden bij de analyse zó berekend dat de energie van het uitgangssignaal minimaal is.

aantal coëfficiënten aan, dus het tijdsbereik waarover het filter de samples uit het verleden onthoudt. Voor de (totale) energie E van het uitgangssignaal, gedefiniëerd door

$$E = \sum_n e_n^2, \qquad n = -\infty, \ldots + \infty \tag{4}$$

geldt dan met (3):

$$E = \sum_n \Big(\sum_{k=0}^{M} a_k s_{n-k}\Big)^2 \tag{5}$$

of

$$E = \sum_{i=0}^{M} a_i \sum_{k=0}^{M} a_k \sum_n s_{n-i} s_{n-k}, \tag{6}$$

of

$$E = \sum_{i=0}^{M} a_i \sum_{k=0}^{M} a_k R_{i-k}, \tag{7}$$

waarin

$$R_{i-k} = \sum_n s_{n-i} s_{n-k} \tag{8}$$

de $(i-k)^e$ autocorrelatie van het ingangssignaal s_n definiëert. Hoewel n in principe van $-\infty$ tot $+\infty$ loopt zijn in ons geval per definitie alle ingangssamples s_n buiten het analysevenster, dat uit N samples bestaat, nul. Voor een minimale energie E is dan, na partiële differentiatie van (7) naar de filtercoëfficiënten a_k, af te leiden dat als voorwaarde moet gelden:

$$\sum_{k=0}^{M} a_k R_{i-k} = 0, \qquad i = 1 \ldots M \tag{9}$$

of:

$$\sum_{k=1}^{M} a_k R_{i-k} = -R_i, \qquad i = 1 \ldots M \tag{10}$$

omdat a_0 per definitie 1 is. Dit stelsel (10) van M vergelijkingen met de M filtercoëfficiënten a_k als onbekenden kan recursief worden opgelost na berekening van de autocorrelaties R_{i-k} van het ingangssignaal volgens (8).

Wanneer de filtercoëfficiënten aan (10) voldoen is de energie E van het uitgangssignaal minimaal en deze wordt dan gegeven door:

$$E_m = \sum_{i=0}^{M} a_i R_i. \tag{11}$$

Bij het berekenen van de filtercoëfficiënten a_k wordt alleen gebruik gemaakt van de autocorrelaties (8) van het ingangssignaal. Dat signaal zelf is daarbij niet nader gespecificeerd en mag dus ook ruis zijn. Resultaat is steeds een set filtercoëfficiënten waarvoor geldt dat zij een filter van gegeven orde M definiëren dat de energie van het uitgangssignaal voor het gegeven ingangssignaal zo klein mogelijk maakt. Hoe groot deze minimale energie E_m van het signaal aan de uitgang (het "restsignaal") dan is, hangt af van M. Naarmate het filter meer coëfficiënten heeft zal het beter in staat zijn zijn taak te vervullen en zal de energie van het restsignaal kleiner zijn. E_m neemt monotoon af met toenemende M.

In het frekwentiedomein wordt het analysefilter gekarakteriseerd door zijn (complexe) overdrachtsfunktie $A(z)$, de z−getransformeerde van de impulsresponsie a_k van het filter:

$$A(z) = \sum_{k=0}^{M} a_k z^{-k}. \tag{12}$$

Hierin is z een complexe variabele[1] en z^{-k} op te vatten als een vertraging over k samples, dus voor $k = 0$ géén verschuiving in de tijd, voor $k = 1$ een verschuiving over één bemonsteringsperiode T, voor $k = 2$ over 2 keer T enz., zie Fig. 6.

De z-getransformeerde $E(z)$ van het uitgangssignaal e_n wordt gegeven door:

$$E(z) = A(z).S(z), \tag{13}$$

[1]Door de eenheidscirkel in het z-vlak te doorlopen, dus $z = e^{j\omega}$ te kiezen met $\omega = 2\pi f$, gaat de z-transformatie over in de Fourier-transformatie en kan uit de overdrachtsfunktie $A(z)$ van het digitale filter het frekwentiespectrum worden berekend.

waarin $S(z)$ de z-getransformeerde is van het ingangssignaal s_n. Voor de energie E van het uitgangssignaal geldt dat de kwadratensom (4) in het tijddomein ook geschreven kan worden als integratie in het frekwentiedomein (theorema van Parseval):

$$E = \sum_n e_n^2 = \frac{1}{2\pi} \int_{-\pi}^{+\pi} | E(\omega) |^2 \, d\omega \tag{14}$$

Minimaliseren van E volgens (10) betekent dat, geïntegreerd over het gehele frekwentiegebied, de energie van het uitgangssignaal, bij gegeven M, zo klein mogelijk wordt gemaakt. Voor het betreffende spraaksegment van N samples in het analysevenster is de spectrale omhullende van het uitgangssignaal, door het filter waarvan de coëfficiënten aan (10) voldoen, dan zo vlak mogelijk gemaakt.

Een voorbeeld van een 10^e-orde filter, berekend voor zowel een periodiek als een ruizig ingangssignaal, is weergegeven in Fig. 7. Links staan

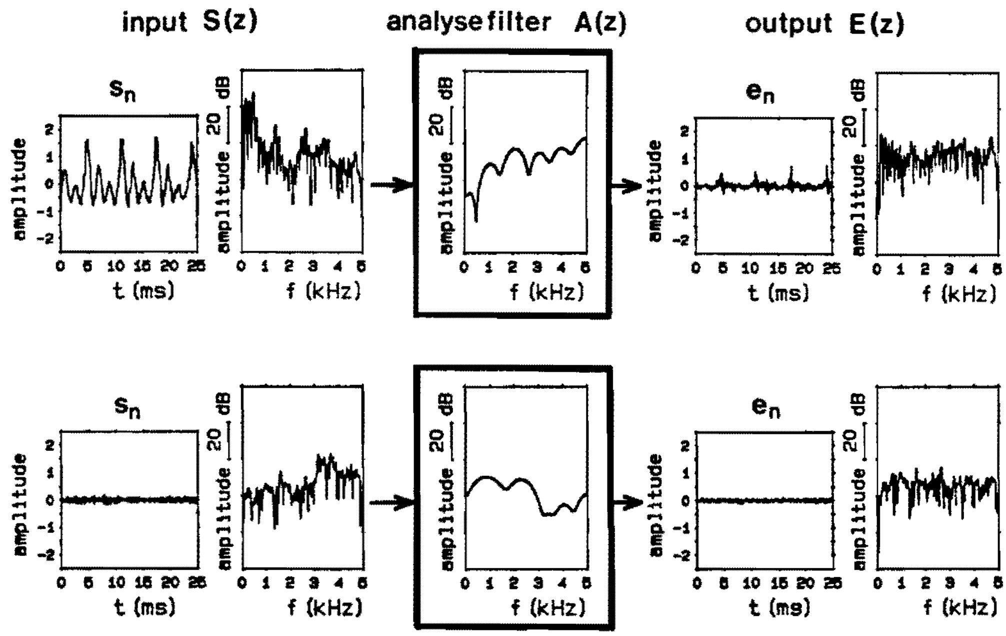


Fig. 7: Voorbeeld van het energiespectrum (midden) van een 10^e-orde analysefilter $A(z)$, berekend voor een periodiek (stemhebbend) ingangssignaal (boven) en een ruizig (stemloos) ingangssignaal (onder). Het spectrum van het restsignaal aan de uitgang van het filter is bij benadering vlak (wit) geworden.

golfvorm en energiespectrum van beide ingangssignalen, rechts die aan de uitgang. We zien hoe het energiespectrum van het restsignaal e_n door het filter is vlakgestreken, maar niet perfekt. Hoe meer coëfficiënten M het filter heeft des te beter kunnen pieken (resonanties) in de omhullende van het ingangsspectrum door dalen (antiresonanties) van het berekende analysefilter worden "geneutraliseerd". Het daarvoor benodigde energiespectrum van het analysefilter is in het midden in Fig. 7 weergegeven voor $M = 10$.

3.4 Synthese met het geïnverteerde analysefilter

We waren op zoek naar de parameters voor het synthesefilter van het bronfilter model om daarmee kunstmatige spraak te genereren. Bij een gegeven ingangssignaal zijn de coëfficiënten van het M^e-orde analysefilter zó bepaald dat de spectrale omhullende van het signaal aan de uitgang van het filter, voor gegeven M, zo vlak mogelijk is geworden. Dat betekent dat deze overdrachtsfunktie $A(z)$ de *geïnverteerde* omhullende van het ingangsspectrum zo goed mogelijk benadert. Dus beschrijft $1/A(z)$ de spectrale omhullende van het ingangssignaal zo goed mogelijk. Als synthesefilter voor het bronfiltermodel definiëren we dan ook:

$$H(z) = \frac{1}{A(z)} \tag{15}$$

Het ingangssignaal voor dit synthesefilter is het bronsignaal u_n met als z-getransformeerde $U(z)$ en het uitgangssignaal is een signaal s'_n, met z-getransformeerde $S'(z)$, dat het spraaksignaal aan de ingang van het analysefilter zo goed mogelijk benadert. In het z-domein geldt dus:

$$S'(z) = U(z).H(z) \tag{16}$$

of met (15) en (12):

$$S'(z) = \frac{U(z)}{A(z)} = \frac{U(z)}{1 + \sum_{k=1}^{M} a_k z^{-k}}. \tag{17}$$

Terugtransformatie naar het tijddomein levert:

$$s'_n + \sum_{k=1}^{M} a_k s'_{n-k} = u_n, \tag{18}$$

of

$$s'_n = u_n - \sum_{k=1}^{M} a_k s'_{n-k}. \tag{19}$$

Het uitgangssignaal wordt dus gegeven door het *verschil* van ingangssignaal op tijdstip n en een lineaire combinatie van M daaraan voorafgaande *uitgangs*samples, zoals is geschetst in Fig. 8. Merk op dat de parameters van het synthesefilter uit die van het analysefilter zijn verkregen door slechts het teken om te keren.

Het ingangssignaal u_n voor dit synthesefilter kunnen we in principe vrij kiezen. Als we daarvoor bijvoorbeeld het restsignaal e_n nemen, dus de output van het analysefilter, dan krijgen we als output van het synthesefilter weer het oorspronkelijke spraaksignaal s_n terug, immers het produkt van analysefilter $A(z)$ en synthesefilter $H(z)$ is per definitie 1. Wij zullen echter als ingangssignaal voor het synthesefilter een sterk "geïdealiseerd" restsignaal

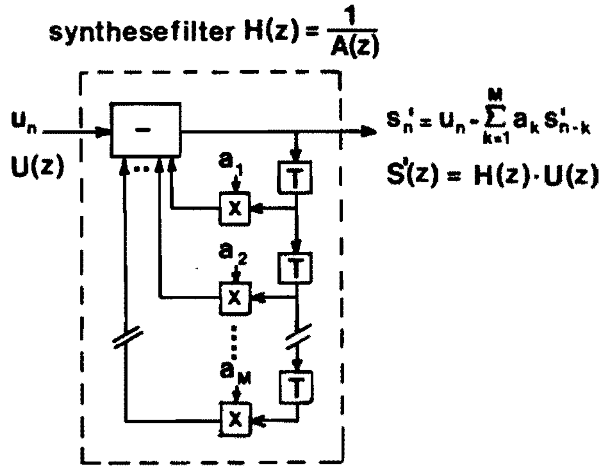


Fig. 8: Digitaal M^e-orde synthesefilter verkregen door inverteren van het analysefilter $A(z)$. Het uitgangssample s'_n op tijdstip nT wordt gegeven door het verschil van het ingangssample u_n op datzelfde tijdstip en M gewogen daaraan voorafgaande uitgangssamples. De weegfaktoren a_k zijn de a-parameters van het berekende analysefilter.

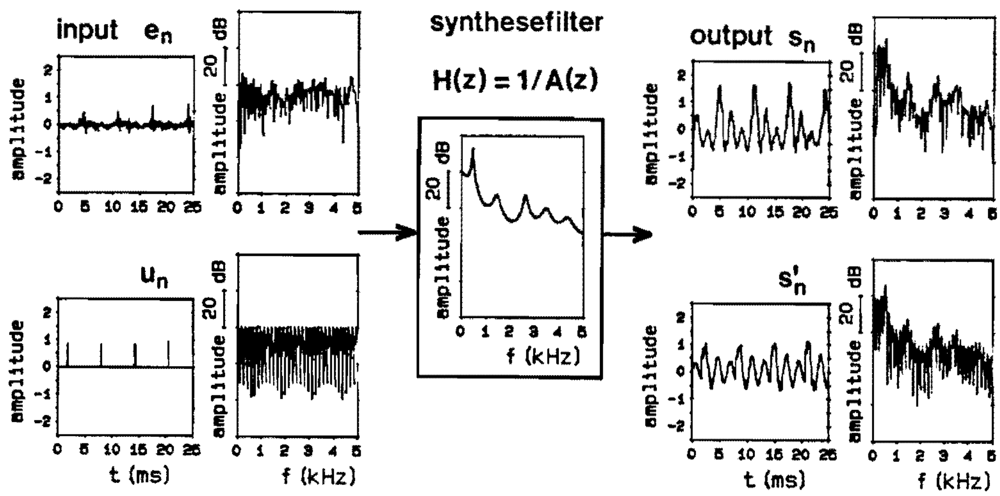


Fig. 9: Voorbeeld van het energiespectrum (midden) van het synthesefilter $H(z)$ verkregen door het analysefilter $A(z)$ van Fig. 7 (voor een periodiek ingangssignaal) te inverteren. Het restsignaal e_n als input (links boven) levert het oorspronkelijke spraaksignaal s_n weer op (rechts boven). Bij de resynthese wordt echter niet e_n maar een impulsreeks u_n (links onder) als ingangssignaal toegepast. Dat levert een synthetisch spraaksignaal s'_n (rechts onder), waarvan de spectrale omhullende die van het oorspronkelijke spraaksignaal benadert.

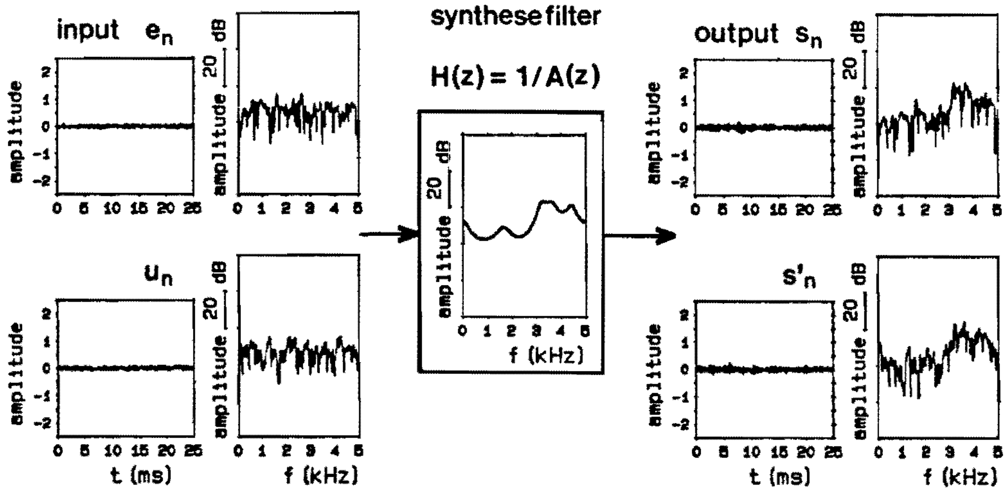


Fig. 10: Als Fig. 9, maar nu voor een ruizig ingangssignaal. Links boven het (ruizige) restsignaal e_n van Fig. 7, dat als input voor het synthesefilter het oorspronkelijke spraaksignaal s_n oplevert. Bij de synthese wordt echter witte ruis als ingangssignaal u_n voor het synthesefilter gebruikt, hetgeen een (zeer goede) benadering s'_n oplevert.

nemen: impulsen of ongecorreleerde (witte) ruis, geheel conform het vereenvoudigde produktiemodel. Dit is schematisch weergegeven in Fig. 9 voor een stemhebbend en in Fig. 10 voor een stemloos stukje spraak.

Rest ons nog het bepalen van de versterkingsfaktor G, waarmee in het bron-filtermodel de energie van het spraaksegment moet worden geregeld. Deze is vastgelegd door de eis dat de energie van het signaal na resynthese gelijk moet zijn aan die van het oorspronkelijk ingangssignaal van het analysefilter. We hebben bij de analyse gezien dat de energie van het restsignaal E_m gegeven is door (11). We kunnen dan afleiden dat aan die eis van gelijke energie wordt voldaan als

$$G^2 = E_m = \sum_{i=0}^{M} a_i R_i. \tag{20}$$

Bij de analyse zijn de autocorrelaties R_k en daaruit de filtercoëfficiënten a_k al berekend. Door tevens het inprodukt van beide uit te rekenen krijgen we dus, als "bijprodukt" van de analyse, tevens het kwadraat van de amplitudeversterkingfaktor G. In het frekwentiedomein betekent dit dat het energiespectrum van de resynthese, dat qua vorm al zo goed mogelijk paste bij het ingangsspectrum, na vermenigvuldiging met een faktor G nu ook zo goed mogelijk tot dekking is gebracht met het ingangsspectrum. We zien daarvan een voorbeeld in Fig. 11. Integratie over het gehele frekwentiegebied levert voor de gladde curve hetzelfde op als voor het oorspronkelijk spectrum; beide energieën zijn gelijk. De amplitudeversterkingsfaktor G wordt dus verkregen volgens (20) uit de energie van het restsignaal. Het restsignaal zelf wordt bij

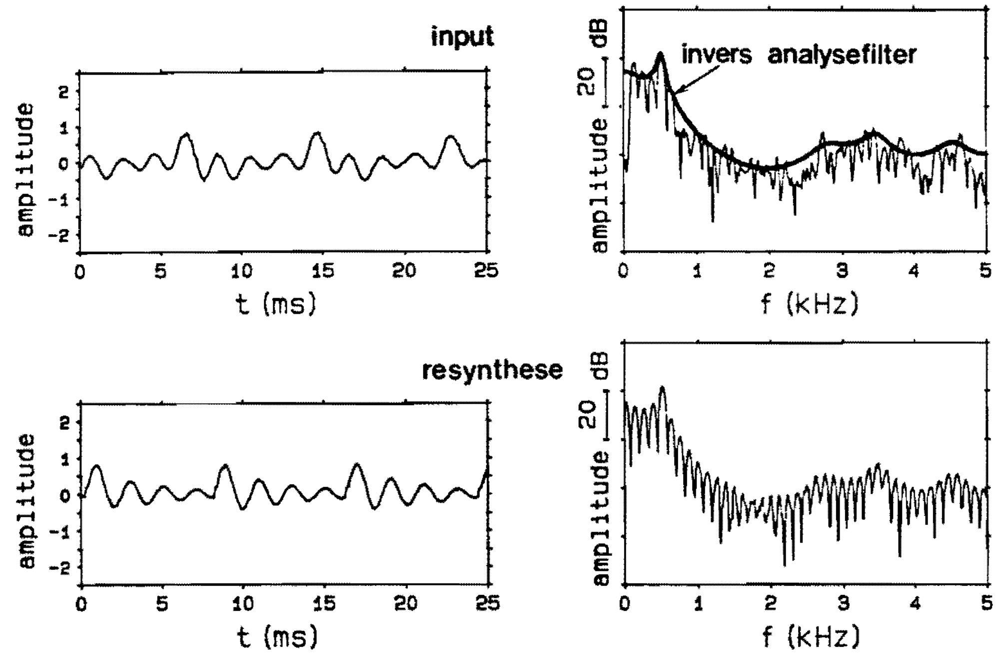


Fig. 11: Voorbeeld van golfvorm (links) en energiespectrum (rechts) van het ingangssignaal (boven) en de resynthese (onder). De dikke gladde curve rechts boven is de overdrachtsfunktie van het geïnverteerde analysefilter dat bij de gegeven orde M (hier 10) zo goed mogelijk past bij het ingangsspectrum.

de analyse niet berekend. We merken in dit verband nog op dat het restsignaal bijna evenveel informatie bevat als het oorspronkelijke spraaksignaal, n.l. alles wat niet door het analysefilter wordt beschreven. Dat is dus het gehele fasespectrum en bovendien die details van het amplitudespectrum die niet door het analysefilter kunnen worden geneutraliseerd omdat er slechts een beperkt aantal van M filtercoëfficiënten beschikbaar is.

De naam "lineaire predictie" voor deze methode van invers filteren heeft de volgende achtergrond. Het minimaliseren van de energie van het restsignaal e_n gegeven door (3) is ook op te vatten als het minimaliseren van de fout die we maken als we voor ieder sample s_n een "voorspelling" $\hat{s}_n$ maken, gedefiniëerd door een lineaire combinatie van M voorafgaande samples, dus:

$$\hat{s}_n = \sum_{k=1}^{M} a_k s_{n-k} \tag{21}$$

De bij deze "lineaire predictie" optredende fout e'_n is het verschil tussen het werkelijke ingangssignaal s_n en de voorspelde waarde $\hat{s}_n$, dus

$$e'_n = s_n - \hat{s}_n = s_n - \sum_{k=1}^{M} a_k s_{n-k} \tag{22}$$

Minimaliseren van dit verschil e'_n over alle samples binnen het analysevenster, met het criterium van kleinste kwadraten, levert dan hetzelfde stelsel vergelijkingen als (10), maar dan met alle coeëfficiënten a_k voorzien van een negatief teken. Daaruit volgt dus precies het synthesefilter $H(z)$.

3.5 Andere uitvoeringsvormen van analyse- en synthesefilter

De beschrijving van de spectrale omhullende van het spraaksignaal met één hogere-orde ("direct form") filter heeft een aantal praktische bezwaren. Ten eerste is een dergelijk filter vrij gevoelig voor onnauwkeurigheden in de parameters en kan aantasting van de coëfficiënten door afbreek-, kwantiseer- of transmissiefouten tot een instabiel filter leiden. Ten tweede leveren de a-parameters nog niet direkt informatie over de ligging en "scherpte" (selectiviteit) van de deelfilters. Daarom wordt in de praktijk uit het hogere-orde filter vaak een andere filterstruktuur afgeleid, waarvan we er hier twee zullen noemen.

3.5.1 Cascade van tweede-orde deelfilters

De eerste mogelijkheid is om over te gaan op een cascade van tweede-orde filters en daarmee het verband tussen filtercoëfficiënten en resonantiefrekwenties expliciet te maken. We kunnen het M^e-orde analysefilter omrekenen naar een cascade van $M/2$ tweede-orde filters. Eén zo'n tweede-orde sectie kan in principe één resonantiepiek in het ingangsspectrum voor zijn rekening nemen. Het M^e-orde filter heeft als overdrachtsfunktie (12):

$$A(z) = 1 + \sum_{k=1}^{M} a_k z^{-k}, \tag{23}$$

en dit polynoom is ook als produkt van kwadratische termen te schrijven. Aannemende dat M even is geldt:

$$A(z) = \prod_{k=1}^{M/2} (1 + p_k z^{-1} + q_k z^{-2}), \tag{24}$$

waarin p_k en q_k de filtercoëfficiënten zijn van de tweede-orde secties (deelfilters) die we verder pq-parameters zullen noemen, zie Fig. 12. Voor het afsplitsen van kwadratische termen van een hogere-orde polynoom bestaan numeriek-iteratieve methoden, die nulpunten van een polynoom bepalen.

Het verband tussen afstemfrekwentie F en bandbreedte B van een analoog tweede-orde filter en de coëfficiënten p en q van het overeenkomstige digitale tweede-orde filter wordt gegeven door de relatie tussen het s-vlak (Laplace-transformatie) en het z-vlak (z-transformatie). Voor een filter met toegevoegd complexe nulpuntenparen in de overdrachtsfunktie is dan af te leiden dat

$$F = \frac{1}{2\pi T} \arccos(\frac{-p}{\sqrt{2q}}), \qquad q > \frac{p^2}{4} \tag{25}$$

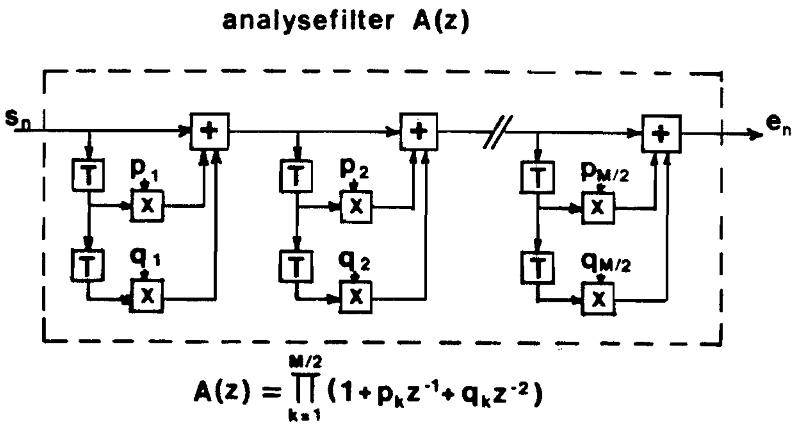


Fig. 12: Het M^e-orde analysefilter $A(z)$ weergegeven in de vorm van een cascade van $M/2$ tweede-orde deelfilters, met als parameters de filtercoëfficiënten p_k en q_k.

en

$$B = \frac{-1}{\pi T} \ln(\sqrt{q}), \qquad q > 0, \tag{26}$$

waarin T weer de bemonsteringsperiode $1/f_s$ is. Hiermee hebben we het M^e-orde analysefilter weergegeven met (ten hoogste) $M/2$ antiresonanties, in de vorm van pq-parameters (de coëfficiënten van de tweede-orde digitale filters). Die kunnen we vervolgens omrekenen naar zogeheten FB-parameters of FQ-parameters, met F de afstemfrekwentie, B de bandbreedte en Q de kwaliteitsfaktor van de afzonderlijke antiresonanties. Deze beschrijving in tweede-orde secties is niet alleen aantrekkelijk omdat we daaruit in principe de afstemfrekwenties en bandbreedtes van de afzonderlijke resonanties in het amplitudespectrum kunnen afleiden. Zij heeft ook als voordeel dat daarmee een robuuster filter ontstaat, dat minder gevoelig is voor instabiliteiten door b.v. kwantiseerfouten. De pq- of FB-parameters kunnen dan ook zeer zuinig gecodeerd worden. We komen daar nog op terug.

3.5.2 Ladderfilter: reflectiecoëfficiënten

De tweede mogelijkheid is om uit het hogere-orde filter een tralie- of ladderfilter te berekenen. Dit kan opgevat worden als het elektrisch model voor een (verliesvrije) akoestische buis die uit M stukken bestaat waarvan de doorsnede stapsgewijs variëert, zoals is geschetst in Fig. 13. Op de overgangen tussen de secties wordt de akoestische energie gedeeltelijk gereflecteerd en de coëfficiënten van zo'n ladderfilter worden dan ook *reflectiecoëfficiënten* genoemd. De buis vormt een ruwe, sterk vereenvoudigde benadering van de vorm van het spraakkanaal en kan, onder bepaalde voorwaarden, een redelijke beschrijving geven van de geometrie van het spraakkanaal. Ook deze reflectiecoëfficiënten lenen zich goed voor kwantisering, zuinige codering en

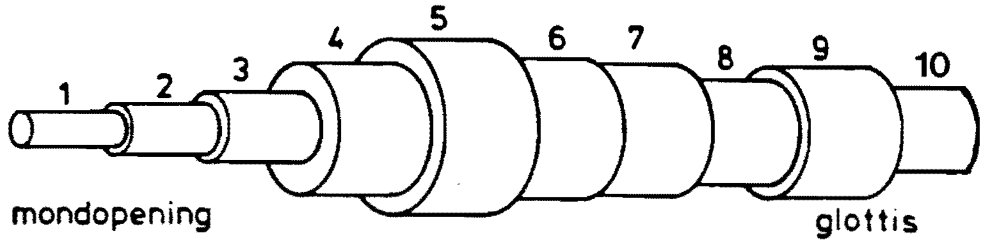


Fig. 13: Schematische voorstelling van het spraakkanaal als een akoestische buis bestaande uit M (hier 10) secties van vesrchillende doorsnedes.

transmissie omdat door het opleggen van eenvoudige beperkingen aan de parameters instabiele filters vermeden kunnen worden.

3.6 Bepaling van de overige modelparameters

De standaard LPC-analyse zoals hiervoor is uiteengezet wordt uitgevoerd voor een spraaksegment van 25 ms en levert zowel de parameters van het filter als de amplitudeversterkingsfaktor G. De stem/stemloos parameter VUV kan worden berekend uit de globale helling van de omhullende van het energiespectrum. Het spectrum van de stembandtrillingen heeft een helling van ongeveer -12 dB/oct. Samen met het stralingseffekt van de mondopening van +6 dB/oct resulteert dit in een helling van -6 dB/oct voor stemhebbende klanken. Voor stemloze klanken vervalt de -12 dB/oct. Het ligt dus voor de hand om de stem/stemloosbeslissing te baseren op de globale helling van de omhullende van het energiespectrum. Dat kan door berekening van de genormeerde eerste autocorrelatie van het spraaksignaal, gedefiniëerd door:

$$\frac{R_1}{R_0} = \frac{\sum_n s_n s_{n-1}}{\sum_n s_n^2} \tag{27}$$

Dit quotiënt is op te vatten als de coëfficiënt van een analysefilter van orde een, immers met M=1 wordt (10):

$$a_1 = \frac{-R_1}{R_0}. \tag{28}$$

Dus is $-a_1$ de parameter van een eerste-orde filter dat de spectrale omhullende zo goed mogelijk beschrijft. Bij stemhebbende klanken is R_1/R_0 bijna 1; er is een hoge correlatie tussen twee opeenvolgende samples, terwijl bij stemloze klanken die correlatie klein of negatief is.

Vervolgens moeten we nog de herhalingsfrekwentie F_0 van de grondtoon bepalen. Daartoe wordt een analysevenster gebruikt van 40 ms, dus aanzienlijk langer dan de 25 ms die bij de overige parameters is toegepast. Dit hangt samen met het feit dat we ook bij zeer lage F_0 van b.v. 50 Hz nog minstens twee periodes van de grondtoon in het analysevenster willen hebben, teneinde die periode voldoende nauwkeurig te kunnen bepalen. Voor het meten van de grondtoon is in de literatuur een grote hoeveelheid toonhoogtemeters en -algoritmes beschreven. Geen van alle werkt foutloos, maar er zijn voor de praktijk zeer goed bruikbare methoden.

Daarmee hebben we alle parameters van het bron-filtermodel voor het betreffende spraaksegment berekend en de zo verkregen set parameters noemen we een frame. Dan wordt het volgend stukje spraak in het analysevenster geplaatst. De tijdsduur, ook frameduur genaamd, waarover we het venster opschuiven in de tijd, kan in principe vrij worden gekozen. In de praktijk blijkt een frameduur van 10 ms goed te voldoen; dat is voldoende om ook de relatief snel veranderende parameters in het spraakgeluid te kunnen volgen. Aldus wordt de gehele uiting in vaste stappen doorlopen en kunnen de opeenvolgende frames met bij elkaar horende parameters worden opgeslagen. Een voorbeeld van het resultaat van zo'n analyse vinden we in Fig. 14, voor de zin "weet je wié de sleutel gevonden heeft", uitgesproken door een mannenstem. De opeenvolgende frames zijn hierin weergegeven als funktie van

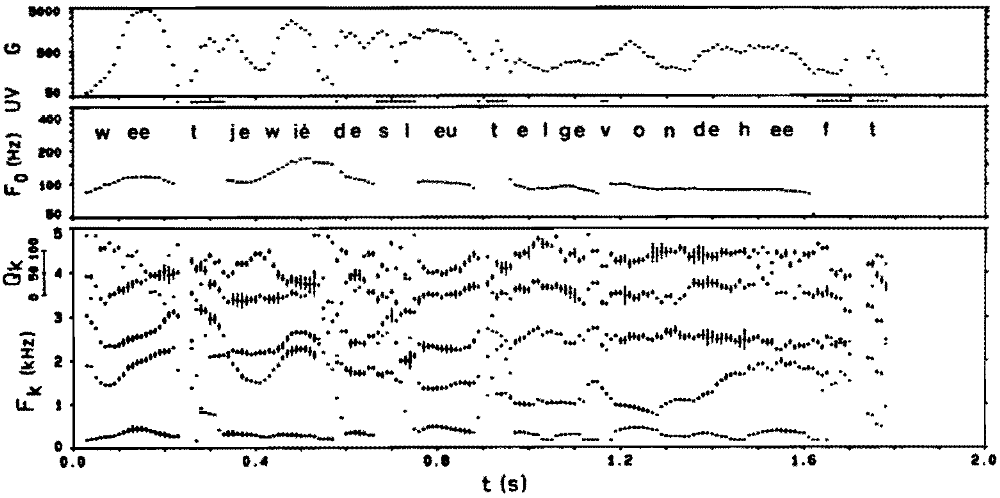


Fig. 14: Voorbeeld van de analyseresultaten voor een mannenstem, met $M = 10$ filtercoëfficiënten geanalyseerd. Van boven naar beneden: amplitudefaktor G, stemloosindicatie UV (zwart=stemloos) en grondtoonfrekwentie F_0. Daaronder de 5 afstemfrekwenties F_k met eromheen de 5 kwaliteitsfaktoren Q_k. De totale lengte van de vertikale streepjes geeft de waarde van Q_k volgens de aparte schaal links.

de tijd; 100 frames voor 1 seconde spraak. In de bovenste helft zijn de bronparameters getekend; de amplitude G en de grondtoonfrekwentie F_0 beide op logaritmische schaal. De onderste helft, het zogeheten *resogram*, toont de FQ-parameters van de individuele tweede-orde filters: de afstemfrekwenties F met daaromheen gecentreerd de kwaliteitsfaktoren Q. Lange verticale strepen betekenen dus een scherpe, selectieve resonantie met een piekwaarde die relatief hoog boven de rest van het spectrum uitsteekt, dus een smalle bandbreedte. Voor de Q-waarden geldt de aparte dimensieloze schaal links;

de verticale frekwentie-as voor de afstemfrekwenties heeft voor de Q-waarden geen betekenis. Dit resogram is enigszins te vergelijken met een spectrogram maar dan zonder grijstinten.

Als we met deze parameters het spraakgeluid weer synthetiseren volgens het bronfiltermodel, kunnen we vaststellen dat de perceptieve overeenkomst tussen de oorspronkelijke versie en de resynthese groot is. Soms horen ongetrainde luisteraars geen verschil, maar meestal zijn wel verschillen hoorbaar. Hoe groot die zijn hangt niet alleen af van de condities waaronder de oorspronkelijke spraak is opgenomen (ruis, nagalm, achtergrondlawaai, meerdere sprekers enz.) en van de luistercondities (koptelefoon versus luidspreker) maar vooral ook van de spreekstem.

Bekijken we de golfvorm van de resynthese en vergelijken we die met de oorspronkelijke dan kunnen we een aantal aspecten opmerken, zie Fig. 15. Globaal gekeken zien we grote overeenkomsten en in het algemeen blijft de

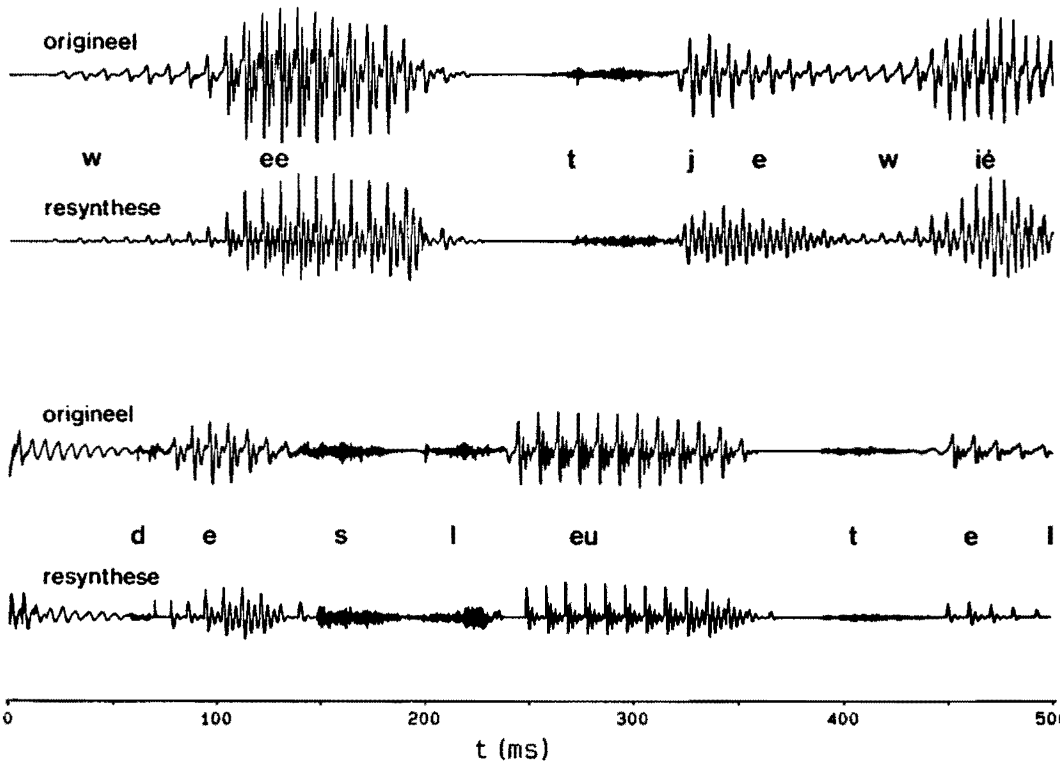


Fig. 15: Voorbeeld van originele en geresynthetiseerde golfvorm voor het eerste stuk van de zin "weet je wié de sleutel gevonden heeft", uitgesproken door een mannenstem.

structuur van het signaal goed bewaard. De sterke amplitudeveranderingen tengevolge van de stembandtrillingen blijven goed bewaard, evenals de afwisselingen tussen stemhebbende en stemloze stukken, stiltes en de temporele opbouw van de verschillende klanken. Maar als we in detail gaan kijken zien we wel kleine verschillen. Ten eerste heeft de golfvorm bij stemhebbende klanken na resynthese een iets ander verloop gekregen. Dat komt in hoofdzaak omdat bij de LPC-analyse alle fase-informatie uit het oorspronkelijke signaal verloren is gegaan. Na resynthese is de energie vaak sterk opeengehoopt in het begin van iedere periode. Dat maakt dat de geresynthetiseerde spraak in de stemhebbende stukken ook soms wat "zoemend" klinkt. Een tweede verschil is te zien bij vooral de plofklanken en bij de overgangen tussen stemhebbende en stemloze klanken en tussen signaal en stilte. We zien hier dat het signaal een beetje uitgesmeerd wordt in de tijd, en dat niet alle details precies bewaard blijven. Zo is de stemhebbende plofklank /d/ van "de sleutel" in het originele signaal een combinatie van periodiek en ruizig signaal en dat wordt bij resynthese vervangen door een geheel stemloos frame, zoals ook in Fig. 14 is te zien. Tenslotte zien we ook een duidelijke fout in de stem/stemloosbeslissing bij de /s/ van "de sleutel". Het eerste frame van de /s/ is ten onrechte stemhebbend geresynthetiseerd.

4 Manipulatie en zuinige codering van spraak

Met de hierboven besproken LPC-analyse en resynthese hebben we de mogeljkheid gekregen om, alvorens het spraakgeluid te synthetiseren, in te grijpen in het parameterbestand en het effekt van zo'n verandering op het spraakgeluid te bestuderen. Dit is van belang voor het spraakonderzoek, omdat we dan inzicht kunnen verwerven dat nodig is om b.v. kunstmatige spraak te maken vanuit tekst. Daarvoor moet het spraakgeluid beheersbaar zijn, zodat we kunnen begrijpen wat de samenhang is tussen taal en waarneembare fysische eigenschappen van spraak.

Een voorbeeld van een dergelijke manipulatie is het vervangen van de oorspronkelijke toonhoogte-contour van een uiting door een gestileerde versie, die voor het oor (van ongeoefende luisteraars) nauwelijks is te onderscheiden van de originele. In Fig. 16a is de oorspronkelijke F_0-contour geschetst (gestippelde curve) voor het zinnetje "vannacht is de vorst ingevallen". Lang niet alle details in het verloop van F_0 zijn van belang voor de perceptie en die kunnen dus worden gladgestreken zonder de correcte perceptieve indruk te verstoren. Dan krijgen we bijvoorbeeld de gestileerde, doorgetrokken contour van Fig. 16a. Na resynthese volgens beide curves horen ongetrainde luisteraars geen verschil tussen beide versies. In de gestileerde curve van Fig. 16a zijn nog twee grote F_0-bewegingen overgebleven, die bij "nacht" en bij "vorst". Deze zijn wel degelijk van perceptief belang; zij zorgen ervoor dat het accent (klemtoon) op de juiste lettergrepen wordt waargenomen. Dat kunnen we demonstreren door de gestileerde contour op één plaats te veranderen en de F_0-beweging bij "vorst" ongeveer 200 ms naar rechts te verschuiven, zodat hij bij "in" komt te te liggen. Dat is weergegeven in Fig. 16b. Na resynthese blijkt daarmee de betekenis van de zin totaal te zijn

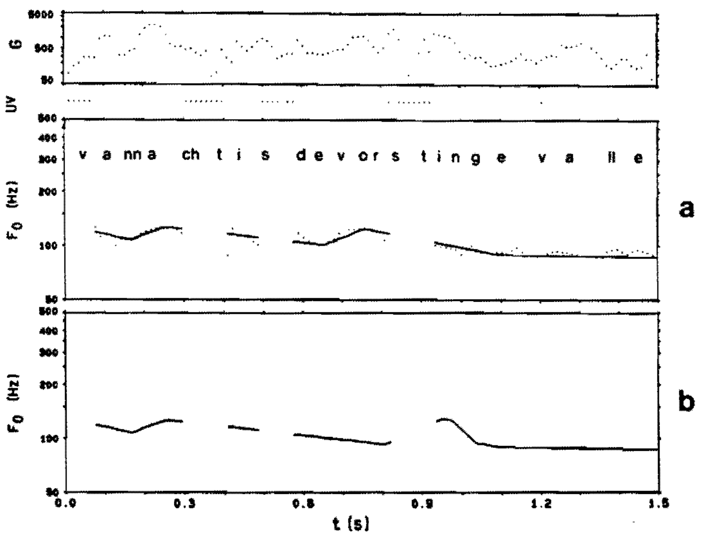


Fig. 16: Voorbeeld van manipulatie van de toonhoogte F_0. Bovenste (gestippelde) curve is de amplitudefaktor G en daaronder de stemloos- indicatie UV. Daaronder in a): (gestippeld) de oorspronkelijke, gemeten toonhoogtecontour en (doorgetrokken) een gladgestreken F_0-contour die dezelfde perceptieve indruk geeft. In b) een veranderde contour, waarbij de F_0-beweging bij "vorst" is verplaatst naar "in", en de zin daardoor een heel andere betekenis krijgt.

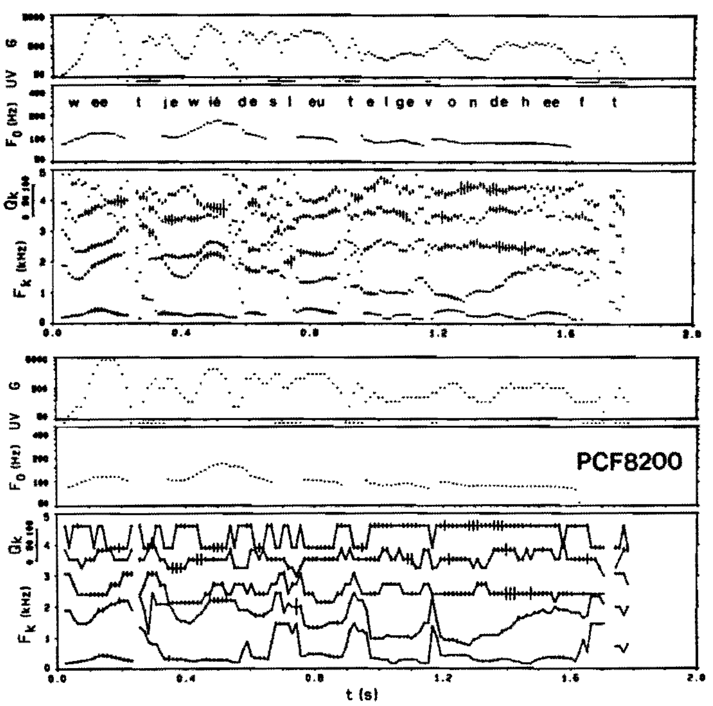


Fig. 17: Voorbeeld van zuinige codering van de parameters voor uitgifte met een spraaksynthesechip. Boven: oorspronkelijke analyseresulaten van Fig. 14. Onder: een codering voor de PCF8200, waarbij ieder frame is gecodeerd met 40 bits, bij een frameduur van 12.8 ms.

veranderd; de uiting wordt niet meer waargenomen als het "invallen van de vorst" maar als het "vallen van de vorstin".

Met het analyse-resynthesesysteem kunnen we iedere willekeurige F_0- contour aanbrengen die voor het intonatie-onderzoek gewenst is.

Een tweede toepassing van het systeem die we hier zullen noemen is zuinige codering. Bij golfvormcodering ligt de ondergrens bij enkele tientallen kbit/s. Met parametercodering kunnen we in principe nog een flink stuk lager komen. De modelparameters worden slechts om de 10 ms berekend. Codering van ieder frame in 120 bits (de stem/stemloosparameter 1 bit, de overige met ieder 10 bit) heeft geen hoorbaar effect op de geresynthetiseerde spraak. We zitten dan op 12 kbit/s, een faktor 10 zuiniger dan de oorspronkelijke golfvormcodering met linPCM (12 bit per sample, f_s =10 kHz). Maar voor een aantal praktische toepassingen kunnen de individuele parameters nog aanzienlijk zuiniger worden beschreven. Ons oor is weinig gevoelig voor kleine afwijkingen in de resonantiefrekwenties en zeer tolerant voor afwijkingen in de bandbreedte. Ook de precieze ligging van de hogere resonanties draagt relatief weinig bij tot de verstaanbaarheid en de resonanties F_4 en F_5 mogen dus ook vrij ruw worden gekwantiseerd. Daarnaast levert een wat langere frameduur doorgaans weinig achteruitgang op in spraakkwaliteit. We krijgen dan een kwantisering zoals is weergegeven in Fig. 17. Dit is een voorbeeld voor de codering voor de PCF8200 spraaksynthesechip met ongeveer 3 kbit/s. Op zo'n chip is een complete spraaksynthetisator geïmplementeerd volgens het bronfiltermodel. Door samenvoeging met de geheugenchip(s) waarin de gecodeerde parameters zijn opgeslagen, kunnen zeer compacte en robuuste systemen voor spraakuitgifte worden gerealiseerd.

5 Referenties

Enden A.W.M. van den, Verhoeckx N.A.M. (1987) *Digitale signaalbewerking*, Delta Press BV., Amerongen.

Flanagan J.L. (1972) *Speech analysis, synthesis and perception*, 2e druk, Springer Verlag, Heidelberg.

Markel J.D., Gray A.H. (1976) *Linear prediction of speech*, Springer Verlag, Heidelberg.

Nooteboom S.G., Cohen A. (1984) *Spreken en verstaan*, 2e druk, Van Gorcum, Assen.

Rabiner L.R., Schafer R.W. (1978) *Digital processing of speech signals*, Prentice Hall, Englewood Cliffs, New Jersey.

Sickert K. (1983) *Automatische Spracheingabe und Sprachausgabe*, Markt & Technik Verlag, München-Haar.

Sluyter R.J.(1983) Digitalisering van spraak. *Philips Technisch Tijdschrift 41*, 209-233.

Witten I.H. (1982) *Principles of computer speech*, Academic Press, London.

DIGITALE SIGNAALBEWERKING

deel II.9

Signaalbehandeling bij de CD

door

R.W.C. Groen

Philips C.E. Eindhoven

Inhoudsopgave

1 Algemene inleiding

In de huidige compact disc spelers wordt reeds een behoorlijke hoeveelheid digitale signaalbewerking toegepast. Het compact disc systeem kent in principe een volledig digitaal datapad, d.w.z. dat de data die van de plaat wordt gelezen zowel tijddiscreet als amplitude discreet (tweewaardig) is. In de huidige spelers wordt er nagenoeg geen digitale signaalbewerking toegepast op het regeltechnisch vlak, d.w.z. bij het aansturen van de actuatoren om de lichtstraal te manipuleren.

Het eerste hoofdstuk in dit dictaat is gewijd aan het compact disc systeem zelf. In dit hoofdstuk komen globaal alle belangrijke systeemelementen aan de orde. Allereerst komt de plaat aan de orde, waarbij o.a. de manier van dataopslag wordt aangegeven. Daarbij worden ook de systeem parameters (bijv. de spoorbreedte) genoemd zodat men zich een voorstelling kan maken over de eisen die we aan de regelsystemen moeten stellen. Vervolgens komen globaal de actuatoren aan de orde die het systeem bezit om de laserstraal "in het spoor" te houden.

In het volgende hoofdstuk wordt een deel van het audiodatapad behandeld. Hierbij gaan we uit van de (fouten)gecorrigeerde data die naar een digitaal analoog omzetter moet worden gestuurd. Er zal worden gesproken over de voor- en nadelen van analoge postfiltering versus digitale prefiltering.

Hoofdstuk vier is gewijd aan een speciale vorm van A/D en D/A conversie, n.l. de sigma delta modulator en de noiseshaper. Deze typen convertoren hebben een aantal interessante eigenschappen, waarvan een van de belangrijkste is dat de afmetingen op een IC zeer klein zijn.

In hoofdstuk vijf wordt een voorbeeld gegeven van het ontwerp van een digitale regeling. Als vehikel hiervoor nemen we de focusregeling uit het compact disc systeem. Allereerst worden de eisen geformuleerd waaraan de regeling moet voldoen. Uitgaande van dit eisenpakket wordt een structuur ontworpen en vervolgens wordt een realisatie getoond. In deze realisatie worden de componenten zoals behandeld in hoofdstuk vier toegepast.

2 Het CD-systeem

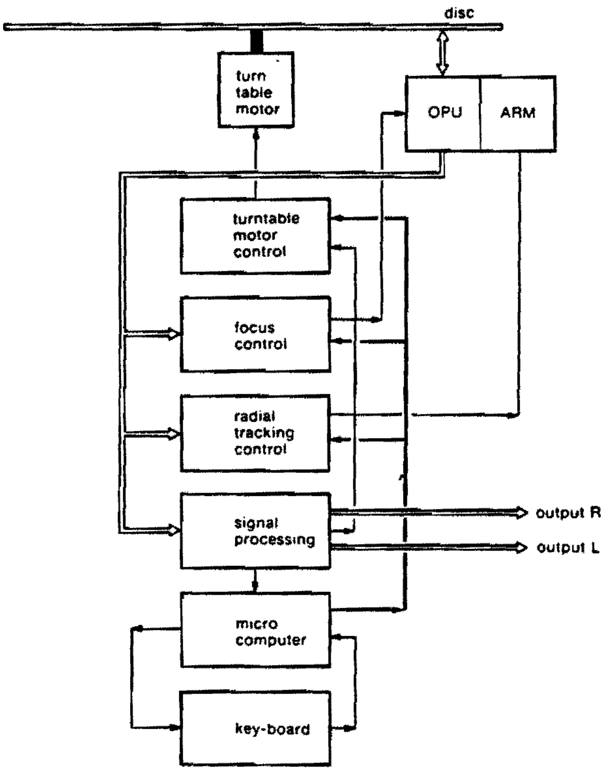
2.1 Inleiding

Een bijzonder kenmerk van het Compact Disc systeem is dat de informatie is opgeslagen in een reflecterende schijf, die wordt uitgelezen door middel van een optisch systeem [1]. Alvorens de informatie op de plaat op te schrijven worden er foutencorrectie-bits aan toegevoegd en dit geheel wordt gemoduleerd. Het modulatiesysteem (EFM = Eight to Fourteen Modulation) zorgt ervoor dat de informatie wordt aangepast aan het *kanaal*, in dit geval de plaat. In de speler wordt de informatie gedemoduleerd en eventuele fouten worden gecorrigeerd, alvorens het signaal aan een digitaal analoog omzetter aan te bieden. Deze omzetter maakt uit het digitale signaal een reproductie van het oorspronkelijke analoge audiosignaal.

In figuur 1 staat blokschematisch de CD-speler afgebeeld. Deze figuur geeft aan hoe de verschillende functies (waarvan er enkele in dit hoofdstuk aan de orde komen) met elkaar verbonden zijn. De optische uitleesunit (OPU zie figuur 1) is samengesteld uit een laser, een set lenzen en de licht gevoelige diodes (fotodiodes). Het lenzensysteem draagt zorg voor een correcte focussering van de spot op het plaatoppervlak, terwijl de fotodiodes het gereflecteerde licht omzetten in elektrische signalen. De informatie op de plaat is aanwezig in de vorm van putten en dijken, en is aangebracht door deze in het reflecterende plaatoppervlak te persen. Het laserlicht dat in de putten terecht komt wordt zodanig verstrooid, dat de intensiteit van het gereflecteerde licht, welk wordt opgevangen door de fotodiodes, veel lager is dan wanneer het licht op een dijk valt. Dit resulteert dan in een variatie van het elektrische signaal dat de fotodiodes verlaat.

Om de plaat te kunnen uitlezen moet de laser lichtbundel correct op het plaatoppervlak worden gefocusseerd. Om te kunnen focusseren kan de objectief lens naar het plaatoppervlak toe- of van het plaatoppervlak af worden bewogen. Gedurende het uitlezen van de plaat moet de optische uitleesunit zeer nauwkeurig het spoor volgen. Er is echter geen enkel mechanisch contact tussen de OPU en de plaat. We zullen dan ook in de loop

[1] Een deel van de tekst en figuren van dit hoofdstuk is ontleend aan de manual "basis cursus compact disc" uitgegeven door de Philips concern service afdeling.



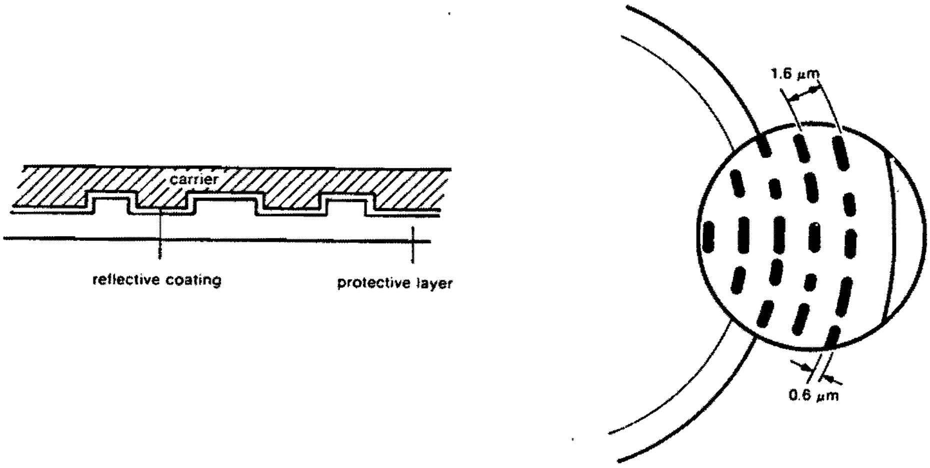
Figuur 1: Blokschema compact disc speler.

van dit hoofdstuk zien dat door gebruik te maken van meerdere fotodiodes, en die op een speciale manier op te stellen, er signalen af te leiden zijn voor de spoorvolging en de focussering van de lichtbundel.

De snelheid waarmee de plaat draait moet ook gecontroleerd worden, om ervoor te zorgen dat het verkrijgen van de informatie (nullen en enen) gebeurt met een constante lineaire snelheid. Om de motor met een constante lineaire snelheid te laten draaien wordt de informatiesnelheid van de plaat gemeten en vergeleken met een referentie. Het resulterende regelsignaal wordt gebruikt om de motor mee te sturen.

2.2 De plaat

Het CD plaatje is samengesteld uit een plastic drager waar de informatie in geperst is in de vorm van putten (zie fig. 2a). Vervolgens wordt er een reflecterende laag aangebracht op de put/dal structuur (bijv. aluminium of zilver). Om het geheel te beschermen wordt er een coating laag over de reflecterende laag aangebracht. De plaat heeft een maximale speelduur van 72 *minuten*.

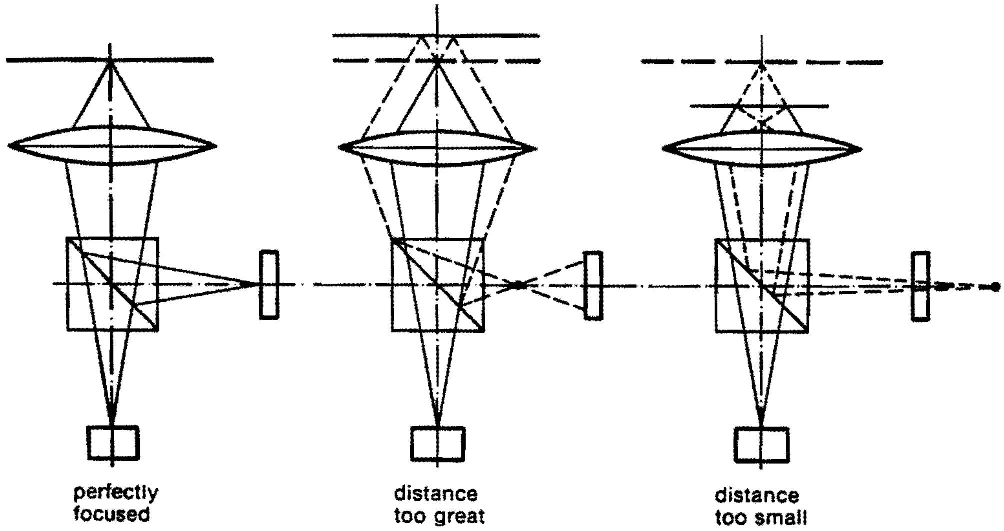


Figuur 2: Het plaatoppervlak a) doorsnede b) loodrecht.

De diameter van de plaat is 120 mm. De gemiddelde steek van de sporen is $1,6\,\mu m$ terwijl de spoorbreedte $0,6\,\mu m$ is (zie figuur 2b). De plaat kan maar van een zijde uitgelezen worden (dit is zo in de plaatstandaard vastgelegd). De plaatsnelheid wordt zodanig geregeld dat de tangentiele snelheid waarmee de sporen worden uitgelezen $1,25\,m/s$ bedraagt. Bij deze afspeelsnelheid hoort een optische afsnijfrequentie van ongeveer $1,4 Mhz$. Deze afsnijfrequentie wordt bepaald door de optiek, infeite de spotdiameter. Boven deze afsnijfrequentie is de output van de detector absoluut nul. Omdat de tangentiële snelheid constant is zal het toerental variëren van ongeveer $500\,rpm$ aan de binnenkant (waar de informatie begint) tot ongeveer $200\,rpm$ aan de buitenkant bij een volgeschreven plaat.

2.3 De focussering

Om een betrouwbaar signaal te verkrijgen op de fotodetector is het een vereiste dat de lichtbundel goed gefocusseerd is. Dat wil zeggen dat het focuspunt van de bundel op de reflecterende laag van de plaat ligt. De toleranties van plaat en draaitafel veroorzaken variaties in de afstand van plaat tot objectief. Dit is schematisch aangegeven in figuur 3. In deze figuur

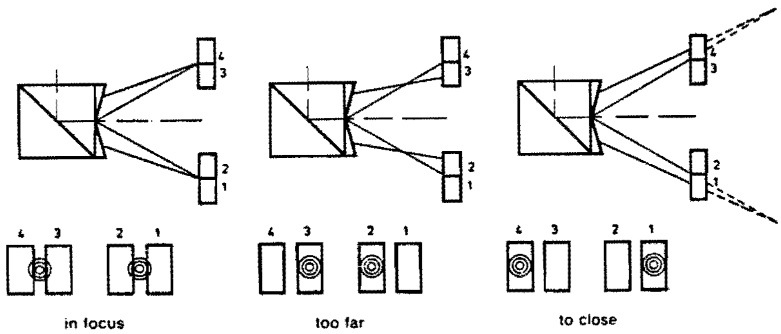


Figuur 3: Variaties in plaat-objectief afstand.

zien we onderaan de laser die licht uitzendt, dat via een halfdoorlatende spiegel en objectief de plaat bereikt. Het gereflecteerde licht komt via het objectief terug op de halfdoorlatende spiegel en vervolgt zijn weg naar rechts naar de detector. We zien dat bij de meest linkse figuur de afstand correct is, en dus het focuspunt van de laserbundel op de reflecterende laag ligt, waardoor het gereflecteerde licht exact wordt gefocusseerd op de detector. In de midden figuur is de plaat-objectief afstand te groot waardoor het focuspunt van het gereflecteerde licht vóór de detector ligt. Dit betekent dat er een *onscherpe afbeelding* op de detector wordt gemaakt en daarmee is een betrouwbare uitlezing onmogelijk. Een dito geval krijgen we als de plaat objectief afstand te klein is, waardoor de afbeelding *scherp is achter* de detector.

Om nu een constante (en correcte) afstand te kunnen garanderen tussen plaat en objectief, is het objectief beweegbaar opgehangen in een richting loodrecht op het plaatoppervlak (de zogenaamde optische as). Deze ophanging kan men vergelijken met die van een luidsprekerspoel. Wanneer men nu de stroom in de spoel varieert kan men het objectief naar de plaat toe of van de plaat af laten bewegen.

Om deze stroom te kunnen bepalen, hebben we een regelsysteem nodig dat uitgaande van de gemeten focusfout een stroom in de spoel doseert. In figuur 4 kunnen we zien hoe de focusfout kan worden opgewekt met behulp van de detector.



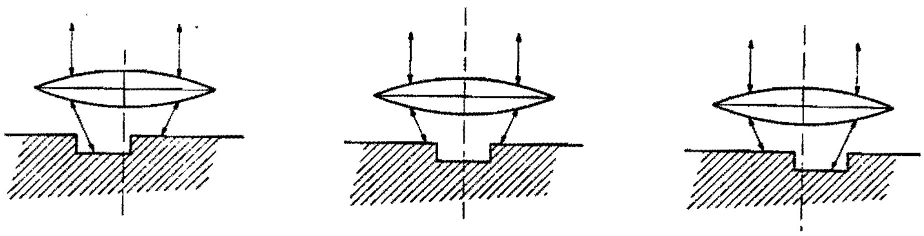
Figuur 4: Focusfout opwekking.

We hebben hier de opstelling van de detector, die dus in werkelijkheid uit vier afzonderlijke detectoren bestaat. Verder zien we dat naast de half-doorlaatende spiegel ook nog een wigvormige scheiding is aangebracht om de terugkerende bundel in tweeën te splitsen. Ook hier zijn weer drie situaties geschetst, nl. correcte focus, te ver en te dicht gefocuseerd. Uitgaande van de nummering van de detectoren zoals in figuur 4 is aangegeven kunnen we nu het focusfoutsignaal FE (Focus Error) formuleren als volgt:

$$FE = (i_1 + i_4) - (i_2 + i_3) \tag{1}$$

2.4 De spoorvolging

Voor de spoorvolging geldt ook dat het spoor zeer nauwkeurig moet worden gevolgd om een betrouwbaar detector signaal te kunnen garanderen. Er zijn altijd imperfecties in de plaat geometrie, zoals onrondheid en een middengat dat niet exact in het midden zit. Hierdoor zal het spoor zich excentrisch rond het "middengat" bewegen en bovendien nog lokale deviaties t.o.v. het midden vertonen.



Figuur 5: Radiele spoorvolging.

Omdat er geen enkel mechanisch contact met de plaat bestaat moet er een elektronisch foutsignaal worden opwekt dat maatgevend is voor de volgfout. Beschouwen we de linkse tekening in figuur 5 dan blijkt dat wanneer de spot gecentreerd is op het spoor, de intensiteit van het gereflecteerde licht links en rechts van de gestippelde lijn, dezelfde diffractie aan het plaatoppervlak heeft ondergaan. Bewegen we nu echter de plaat iets naar rechts (of links), dan zal het gereflecteerde licht t.g.v. diffractieverschillen in de linker en rechter pupilhelft [2] van de lens anders zijn. Deze verschillen kunnen worden gedetecteerd met behulp van de detector opstelling zoals die hiervoor al is beschreven. Dit leidt dan tot de volgende uitdrukking voor het spoorvolgsignaal RE (Radial Error):

$$RE = (i_1 + i_2) - (i_3 + i_4)$$

[2] Dit wordt het Philips één spots of push-pull systeem genoemd.

2.5 De rotatiemotor

De informatiestroom van data van de plaat moet zo goed mogelijk constant worden gehouden. In het hoofdstuk over de plaat is er reeds gesteld dat de tangentiële snelheid (ook wel lineaire snelheid genoemd) ongeveer $1,25m/s$ bedraagt. Het ongeveer slaat hier niet op lokale variaties, maar meer op de gebruikte putgeometrie tijdens het produktieproces van de plaat. Verder weten we ook dat plaat plus draaitafel excentriciteit kunnen vertonen, waardoor de tangentiële snelheid sinusvormig kan varieren met een frequentie gelijk aan de omwentelingsfrequentie van de plaat. Bovendien is het zo dat de eis die we stellen t.a.v. constantheid (jitter eis) van de bemonsteringsfrequentie van het uitgangssignaal (lees: het geregenereerde kloksignaal) zo groot is, dat de rotatie motor dat alleen niet kan bewerkstelligen.

Daarvoor is de regeling van de informatiesnelheid, zoals die van de plaat komt, in twee trappen uitgevoerd, n.l. de motor regelt de gemiddelde lineaire snelheid, terwijl er een elektronische correctie (FIFO= First In First Out) is voorzien om de lokale afwijkingen (bijv. t.g.v. excentriciteit) te compenseren. Met behulp van het tijdsverschil (faseverschil) tussen de overeenkomstige inlees- en uitleesmomenten van het FIFO kan het motortoerental worden geregeld. Wanneer er verder naar het motorregelsignaal wordt gerefereerd dan wordt daarmee dit faseverschilsignaal uit het FIFO bedoeld.

3 Het Audio datapad

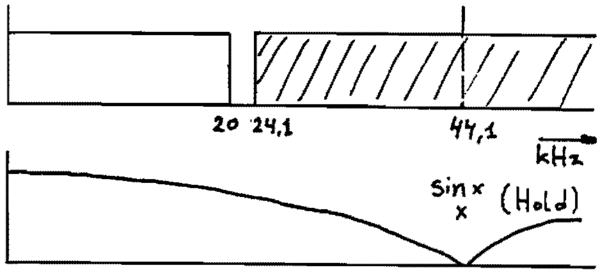
Dit hoofdstuk handelt over de signaalbewerking van de audio signalen nadat de transmissiefouten gecorrigeerd zijn en de tijdbasis hersteld is. Dat betekent dat op dit nivo reeds 16 *bits* samples beschikbaar zijn die naar een digitaal naar analoog omzetter (D/A) kunnen worden gestuurd. Hoe hier de digitale signaalbewerking kan worden ingezet is het onderwerp van dit hoofdstuk.

3.1 Overbemonstering

Zoals reeds vermeld neemt het digitale signaal van een audio kanaal na decodering en error correctie de vorm aan van $16 - bits\, lineare\, PCM$ monsters. De bemonsteringsfrequentie van dit signaal is $44.1\, kHz$. Figuur 6 toont het audio spectrum met een bemonsteringsfrequentie f_s van $44.1\, kHz$. Het spectrum aan de ingang (tijdens opname) wordt begrensd tot maximaal $20\, kHz$, vandar de ruimten van $\approx 4\, kHz$ tussen de spectrale herhalingen. Deze ruimten zijn nodig om de transitie band van het postfilter te leggen. Wanneer de D/A omzetter op $44.1\, kHz$ wordt bedreven, dan zien we dat de sample and hold functie van de D/A omzetter een $\sin x/x$ distorsie geeft. Het valt op dat de sample and hold een aanzienlijke verzwakking veroorzaakt in het basisband spectrum. Verder moet er achter de D/A omzetter een postfilter worden geschakeld om de resterende frequentie componenten boven $20\, kHz$ voldoende te onderdrukken (gearceerde deel).

De eis voor de onderdrukking is ten minste $50\, dB$ t.o.v. de doorlaatband. Dit om ervoor te zorgen dat versterkers en luidsprekerboxen geen niet lineaire effecten kunnen vertonen t.g.v. dit niet hoorbare deel van het spectrum. Om deze getallen te bereiken met een analoog postfilter, dat bovendien ook nog faselineair is, is geen sinecure.

Het is daarom ook dat reeds in de eerste generatie Philips CD spelers een digitale oplossing voor dit probleem is toegepast. Allereerst wordt de bemonsteringsfrequentie f_s een factor vier omhoog gebracht (we noemen dit f_{4s}). Dit wordt gerealiseerd door het toevoegen van monsters die de waarde nul hebben. Het signaal spectrum ziet er dan uit als aangegeven in figuur 7.

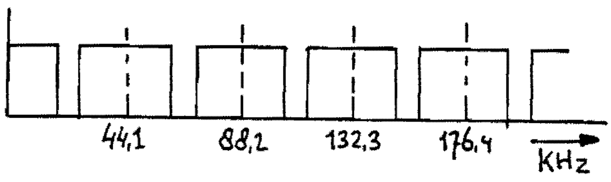


Figuur 6: postfilter eis.

Met behulp van een digitaal filter met een bemonsteringsfrequentie van f_{4s} kunnen we nu het gewenste spectrum eruit filteren. Dit filter moet zoals eerder vermeld een minimale onderdrukking van $50\,dB$ hebben boven $20\,kHz$ en bovendien faselineair zijn. Figuur 8 toont de frequentieresponsie van dit filter. Dit filter is uitgevoerd als een *FIR filter* met 120 coëfficiënten. Allereerst merken we op dat in de doorlaatband de responsie iets toeneemt. Dit is de compensatie voor de $\sin x/x$ distorsie van de sample and hold, en de amplitudedistorsie van het analoge postfilter (dit filter wordt later besproken). Verder valt op dat de responsie in de sperband nogal onregelmatig is, hetgeen te wijten is aan de eindige woordbreedte (12-bits) van de coëfficiënten. Overigens kan men wel zien dat over de gehele sperband de gewenste onderdrukking wordt gehaald.

3.2 Implementatie van een interpolerend FIR filter

We noemen het soort filters, waarbij de bemonsteringsfrequentie van de uitgang hoger is als die van de ingang, ook wel interpolerende filters. We staan eerst even stil bij het rekenwerk dat nodig is om zo'n type filter te implementeren voor het linker en rechter audio kanaal in de compact disc speler. Voor het totaal aantal vermenigvuldigingen N (en optellingen) per



Figuur 7: Viervoudig overbemonsterd audio spectrum.

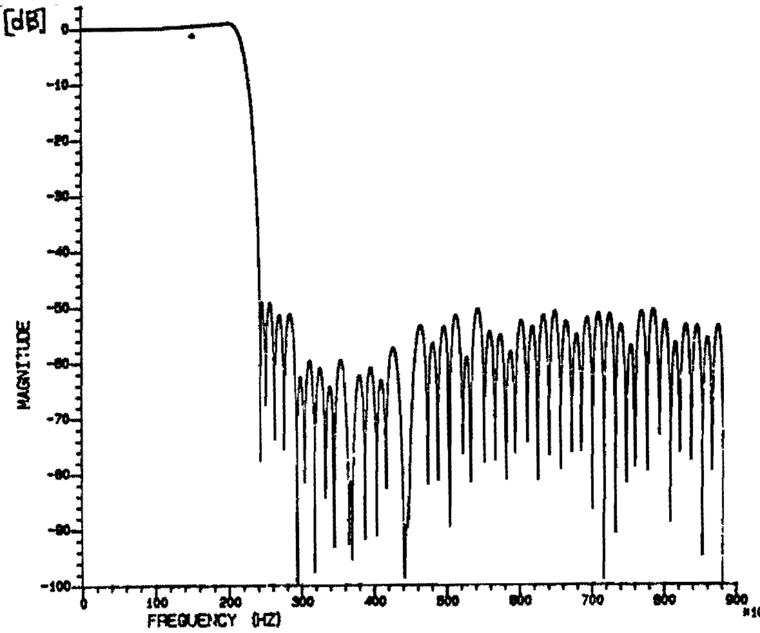
seconde krijgen we:

$$N = A * B * (44.1 * 10^3) * C \approx 42 * 10^6$$

met $A = 2$ (linker en rechter kanaal), $B = 4$ van de overbemonstering en $C = 120$ vanwege de filterlengte. Het zal blijken dat de hoeveelheid rekenwerk niet zo afschrikwekkend hoeft te zijn als het hier lijkt. Want men kan overbemonsteren en filteren ineenweven waardoor het aantal vermenigvuldigingen reduceert. Het blijkt dat men de slechts $\frac{1}{4}$ van het aantal vermenigvuldigingen hoeft uit te rekenen. Een gedetailleerde beschrijving van dit principe kan men vinden in het boek [6] dat bij deze cursus wordt uitgereikt.

3.3 Multi bit D/A omzetter

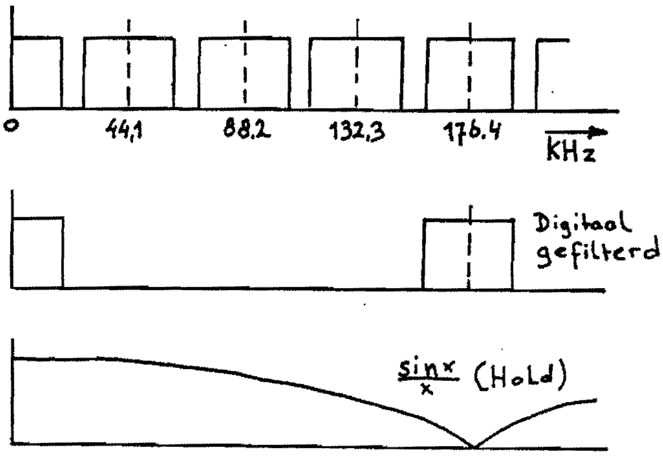
De uitgang van het digitale filter geeft 16 bit monsters af met een frequentie van f_{4s}. Dat uitgangssignaal kan worden toegevoerd aan een multibit D/A omzetter. Het spectrum na filtering ziet er uit als aangegeven in figuur 9. Een eigenschap van het overbemonsteren is dat de kwantisatieruis die wordt toegevoegd aan de uitgang van het filter (er worden 16 bits aan de D/A omzetter toegevoerd) over frequentiegebied $0 \ldots f_{4s}/2$ is verdeeld. Beschouwen we de ruis in de band $0 \ldots f_s/2$ dan is het ruisvermogen in het geval van vier keer overbemonsteren $6\,dB$ beter t.o.v. bemonsteren met f_s.



Figuur 8: Frequentieresponsie van het Fir filter.

In deze figuur is duidelijk te zien dat het resterende stoorspectrum veel eenvoudiger analoog af te filteren is bij de gegeven eisen omdat het veel verder weg ligt in de frequentieband. Verder blijkt dat t.g.v. het hold effect (zoals aangegeven in figuur 9) in D/A omzetter ook nog een extra bijdrage in de onderdrukking wordt geleverd. Als analoog postfilter blijkt een 3^{de} orde Bessel filter met een $3\,dB$ punt op $30\,kHz$ uitstekend te voldoen. De amplitude distorsie die wordt veroorzaakt door het $\sin x/x$ effect en het Bessel filter kan in het digitale filter worden gecorrigeerd door de coëfficiënten een extra inverse weegfactor mee te geven.

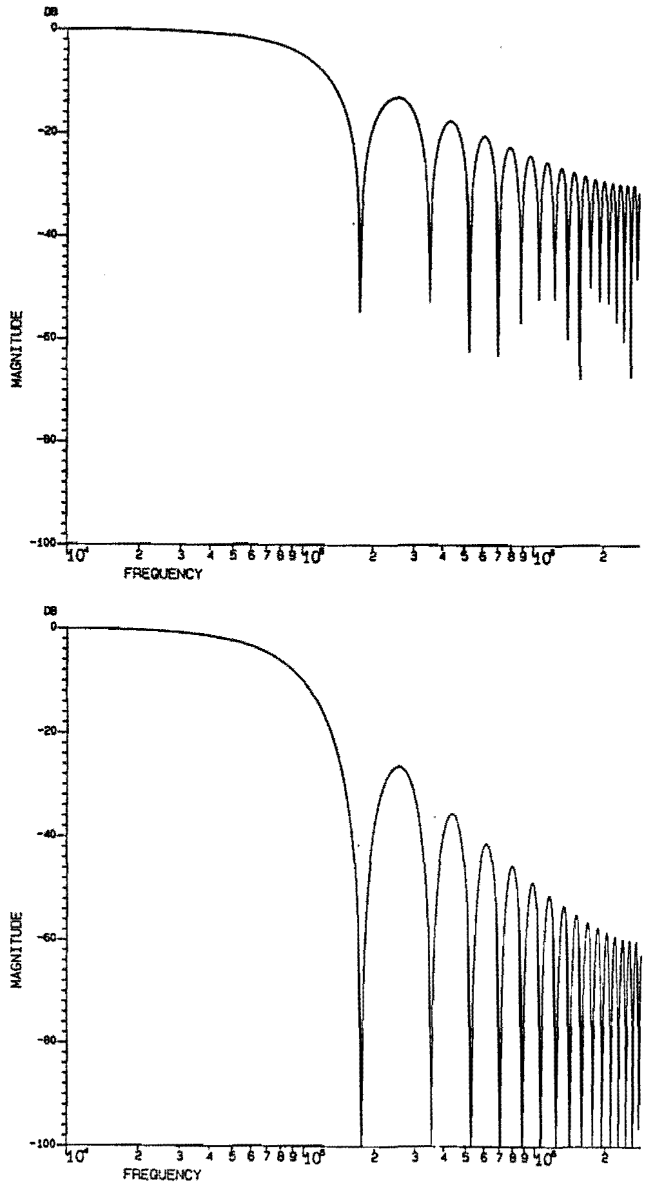
Met de methode van overbemonstering en digitale filtering kunnen we een gecompliceerd analoog postfilter vermijden. De prijs die ervoor moet worden betaald is dat er extra digitale processing nodig is en dat de D/A omzetter op een hogere frequentie moet kunnen werken.



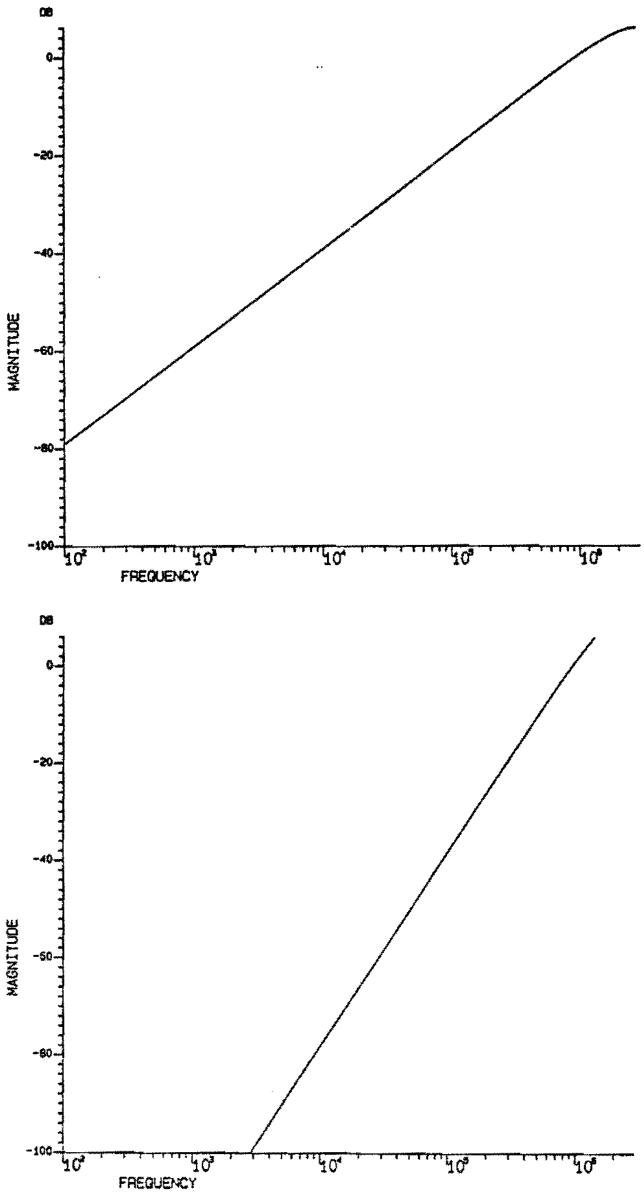
Figuur 9: boven: overbemonsterd met nullen toevoegen, midden: gefilterd audio spectrum en onder: sample and hold distorsie van de D/A omzetter.

3.4 Een bits D/A omzetter

Een ander soort D/A omzetters die men tegenwoordig meer en meer gaat toepassen zijn 1 bit D/A omzetters. Bij deze typen van omzetters wordt een ultra hoge bemonsterings frequentie gekozen, bijv. $128 f_s$. Enige voordelen van dit soort D/A omzetters zijn dat ze zeer klein zijn en geen lineariteits problemen hebben. In het kort zal nu worden beschreven hoe dit soort omzetters wordt aangestuurd uitgaande van bijv. het signaal uit het digitale filter op f_{4s}. Allereerst wordt de bemonsteringsfrequentie verhoogd van f_{4s} tot bijv. de reeds genoemde f_{128s}. Dit kan men doen door een eenvoudig eerste orde houd circuit. Hierdoor bereikt men dat het signaal voor de hoge frequenties reeds wordt gefilterd (zie Figuur 9). Bovendien wordt de amplitude distorsie in de doorlaatband door het digitale filter gecorrigeerd. Een andere methode van de bemonsteringsfrequentie te verhogen is d.m.v. een lineare interpolator. De lineare interpolator in het tijddomein heeft een frequentieresponsie volgens een $(\sin x/x)^2$ functie zoals in Figuur 15 van hoofdstuk 24. De lineare interpolator geeft een betere onderdrukking van de hogere frequentiebanden t.o.v. een nulde orde houd circuit. De beide filteracties zijn geschetst in Figuur 10. Het signaal dat wordt verkregen m.b.v. een der boven beschreven interpolaties wordt dan toegevoegd aan een noiseshaper. Het principe van een eerste orde noiseshaper wordt beschreven op blz. 30 en volgenden. De noiseshaper, of ruisherverdeler, moet nu de 16-bits monsters reduceren naar 1-bit. De daarbij geïntroduceerde kwantisatieruis krijgt een karakter zoals aangegeven in Figuur 11 Achter deze 1-bit noiseshaper volgt een 1-bit D/A omzetter. deze D/A omzetter, meestal uitgevoerd met een geschakkelde capaciteit. Het aldus verkregen uitgangssignaal dient nog gefilterd te worden om de kwantisatieruis weg te nemen. Dit gebeurt net als bij de multibit D/A omzetter met een 3de orde Besselfilter.



Figuur 10: boven: Filtering door N.O.H. actie. onder: Filtering door lineare interpolator



Figuur 11: Ruis spectra van de noise shaper. boven: eerste orde lusfilter. onder: tweede orde lusfilter

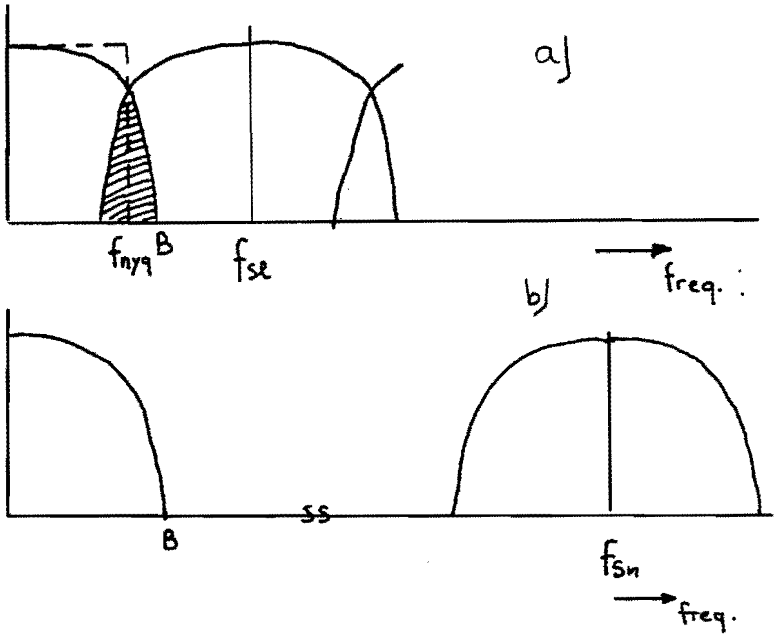
4 Speciale A/D en D/A systemen

In dit hoofdstuk zullen we een systeem behandelen voor A/D conversie de *Sigma Delta* (afgekort als: $\Sigma\Delta$) en een systeem voor D/A conversie de *Noiseshaper*. Verder zullen we zien dat beide systemen functioneel equivalent zijn en derhalve met dezelfde theorie kunnen worden beschreven [2].

4.1 De $\Sigma\Delta$ modulator

Wanneer we een conventioneel A/D systeem bekijken dan bestaat dat meestal uit een bandbegrenzend prefilter met daarachter een sample and hold schakeling gevolgd door de A/D converter. Men kan natuurlijk ook de bemonsteringsfrequentie hoog genoeg kiezen zodat er geen vouwvervorming meer kan optreden, en het bandbegrenzend prefilter achterwege kan blijven. Om in het conventionele geval een degelijke onderdrukking van de vouwvervorming te verkrijgen, moet het prefilter voldoende demping hebben boven de nyquist frequentie f_{nyq}. In figuur 12a is een willekeurig signaal aangenomen met een bandbreedte B (boven de nyquist frequentie) en een bemonsteringsfrequentie f_{sl}. Wanneer er geen prefilter wordt toegepast dan zullen de spectra na bemonstering overlapping vertonen en vouwvervorming leveren (gearceerd gebied). Deze vouwvervorming kan worden voorkomen als een prefilter wordt toegepast, zoals gestippeld in figuur 12a is aangegeven. Het andere alternatief is om de bemonsteringsfrequentie f_{sh} hoog genoeg te kiezen, zoals aangegeven in figuur 12b. Door nu een $\Sigma\Delta$ modulator toe te passen kunnen we de (te) hoge bemonsteringsfrequentie op een andere manier benutten.

Stel we hebben een bandbegrensd signaal met een bandbreedte $B \leq f_{sl}/2$. Indien dit wordt bemonsterd met $2f_{sl}$ en daarna netjes digitaal gefilterd, dan wordt er een winst van $3\,dB$ in signaal-ruis verhouding verkregen. Een dergelijk principe wordt toegepast in een $\Sigma\Delta$ modulator. In figuur 13 is een A/D omzettingssysteem op basis van een $\Sigma\Delta$ modulator geschetst. We nemen aan dat het signaal $i[n]$ tijddiscreet maar amplitude continu is (hier wordt later nog op teruggekomen). De omzetter bestaat uit een loopfilter $G(z)$ en een kwantisator Q (voor de toepassing zoals in dit diktaat beschreven gaan we uit van een 1-bits kwantisator). Verder zien we



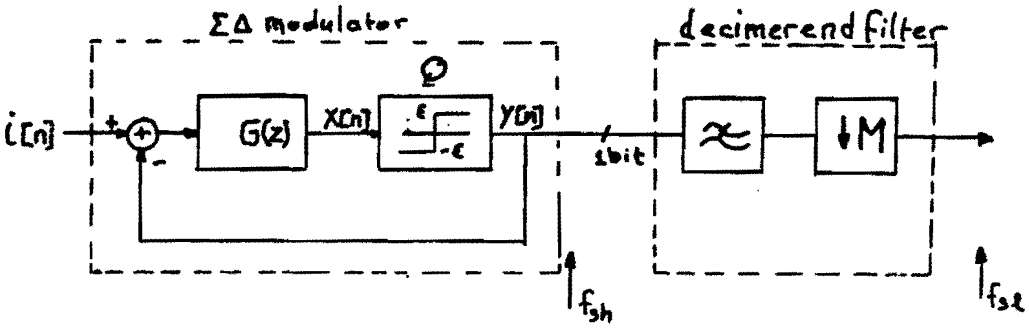
Figuur 12: a) conventioneel A/D systeem b) $\Sigma\Delta$ modulator systeem.

dat de $\Sigma\Delta$ modulator nog wordt gevolgd door een filter en een bemonsteringsfrequentieverlager M. Deze combinatie wordt ook wel decimerend filter genoemd en komt in de volgende paragraaf aan de orde. Bij een $\Sigma\Delta$ modulator wordt het spectrum van de kwantiseringsruis via een loopfilter gefilterd (shaping). Deze filtering is zodanig dat de ruis in de gewenste frequentieband (bijv. $0 \ldots f_{sl}/2$) wordt verminderd ten koste van de ruis buiten deze band. De kwantisator Q kan gemodelleerd worden door:

$$y[n] = cx[n] + r[n] \tag{2}$$

Hierin is $x[n]$ het ingangssignaal van de kwantisator, $y[n]$ het uitgangssignaal en $r[n]$ de kwantisatieruis die er wordt geinjecteerd. De factor c stelt de globale versterking van de kwantisator voor. Deze factor c is wel aannemelijk te maken wanneer men eerst uitgaat van een meerbits "magnitude truncation" kwantisator [1]. We kunnen voor het uitgangssignaal $Y(z) = Z\{y[n]\}$ van de $\Sigma\Delta$ modulator (vóór het decimerend filter) schrijven dat

$$Y(z) = \frac{cG(z)}{1 + cG(z)} I(z) + \frac{1}{1 + cG(z)} R(z). \tag{3}$$



Figuur 13: blokdiagram van een overbemonsterd A/D systeem.

Bovenstaande vergelijking geeft het signaal de uitgang van de overbemonsterde $\Sigma\Delta$ modulator. In praktische realisaties is het ingangssignaal $i[n]$ niet tijddiscreet en het lusfilter G evenmin, immers we willen een analoog digitaal omzetter bouwen. In feite is uitdrukking (3) dus een benadering. Uit simulaties blijkt echter dat bovenstaande vergelijking een zeer goede benadering van het systeem geeft. Het frequentiegebied waarin we geinteresseerd zijn ligt factoren lager dan de bemonsteringsfrequentie f_{sh}. Om naar het gewenste PCM formaat te gaan wordt de bemonsteringsfrequentie van het 1-bits signaal door een decimerend filter met een factor M teruggebracht zoals in figuur 13 is aangegeven. Om een eis te kunnen formuleren voor $G(z)$ voeren we de substitutie $z = \mathrm{e}^{\mathrm{j}\theta}$ in. Voor het frequentiegebied waarin we geinteresseerd zijn nemen we aan dat $\theta_1 \leq \frac{\pi}{M}$. Als we er nu voor zorgen dat $G(\mathrm{e}^{\mathrm{j}\theta}) \gg 1$ voor $\theta \leq \theta_1$ dan kunnen we schrijven voor (3)

$$Y(\mathrm{e}^{\mathrm{j}\theta}) = I(\mathrm{e}^{\mathrm{j}\theta}) + \frac{1}{1 + cG(\mathrm{e}^{\mathrm{j}\theta})} R(\mathrm{e}^{\mathrm{j}\theta})$$

en we zien dat het signaal praktisch onvervormd door de modulator heen gaat, terwijl de kwantisatieruis een zodanige shaping moet ondergaan dat in het gewenste frequentiegebied weinig ruisenergie overblijft.

Een klasse van loopfilters die aan de gestelde eis kunnen voldoen zijn te schrijven als

$$G(z) = \frac{(z-a)^N - (z-b)^N}{(z-b)^N}$$

waarvoor geldt dat $-1 < a \leq b \leq 1$ en N de orde van het filter aangeeft. Wanneer we ons beperken tot een eerste orde $\Sigma\Delta$ modulator ($N = 1$) en als we kiezen $a = 0$ en $b = 1$ dan krijgen we een eenvoudige integrator als loopfilter. Voor de globale versterking c mogen we in dit geval $\approx$ 1.3 aannemen (zie [2]). De systeemfunctie wordt gegeven door:

$$G(z) = \frac{1}{(z-1)} = \frac{z^{-1}}{1-z^{-1}} \tag{4}$$

Beschouwen we tenslotte nog grootte van het ruisvermogen dat er wordt geintroduceerd bij de kwantisator, en daarmee gekoppeld de signaal-ruisverhouding aan de uitgang van de $\Sigma\Delta$ modulator. Wanneer we de $\Sigma\Delta$ modulator bekijken dan is het moeilijk te stellen dat we die ruisbron als "wit" en ongecorreleerd mogen beschouwen. Het is echter zó dat simulaties hebben aangetoond [3] dat deze aanname toch goede gemiddelde ruisgetallen geeft. Het ruisvermogen dat we kunnen associëren met deze ruisbron is

$$P_{noise} = \frac{q^2}{3} \tag{5}$$

waarbij $2q$ de kwantisatiestap aangeeft. Figuur 14 geeft het spectrum van de ruis aan de uitgang van de modulator, uitgegaan van (4) en (5) met $c = 1$.

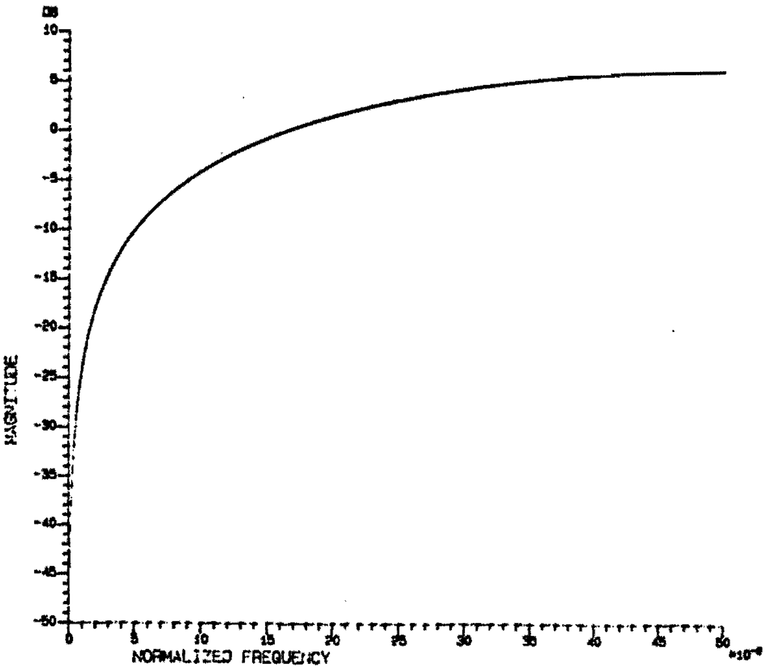
Indien we de $\Sigma\Delta$ modulator exciteren met een sinus dan wordt de signaal-ruisverhouding in de band van $(0 \ldots 0,5f_{sh})$ bepaald door

$$SNR = 0,5q^2 / \frac{q^2}{3} \frac{1}{2\pi j} \oint |1 - z^{-1}|^2 dz$$

en bedraagt :

$$SNR = -1.2\,dB$$

De signaal-ruisverhouding in het gewenste frequentiegebied $(0 \ldots f_{sl}/2)$ is echter veel groter. Bij een $\Sigma\Delta$ modulator met een eerste orde loopfilter geeft een factor twee overbemonsteren telkens een extra winst in signaal-ruisverhouding van $6\,dB$. Hiermee komt de totale winst op $9\,dB$ per factor



Figuur 14: Spectrum van de ruis aan de uitgang van de $\Sigma\Delta$ modulator

twee overbemonsteren (dit getal geldt als een soort vuistregel en wordt verkregen met (4) bij een keuze van $c = 1$). De beperking tot een eerste orde $\Sigma\Delta$ modulator doet niets af aan de algemene beschouwing. Wel is het zo dat bij hogere orde systemen instabiliteiten kunnen optreden waaraan speciale aandacht moet worden geschonken [2].

4.2 Decimerende filters

4.2.1 Filter beschrijving

Zoals in figuur 13 is aangegeven volgt er achter de $\Sigma\Delta$ modulator een decimerend filter. Dit filter heeft een tweeledig doel, n.l. ruisbandbreedte begrenzen en bemonsteringsfrequentie verlagen. In de literatuur [4] wordt een klasse van filters beschreven, de zogenaamde "integrate-and-dump filters", die zeer efficient te realiseren zijn. Het blijkt dat dit type filter gecombineerd met een 1-bit $\Sigma\Delta$ modulator een zeer compacte A/D omzetter oplevert. De systeemfunctie van een (eerste orde) integrate-and-dump filter wordt gegeven door:

$$H_1(z) = \frac{1}{N}\left(\frac{1 - z^{-N}}{1 - z^{-1}}\right)$$

(men noemt dit ook wel kamfilter of moving average filter). De orde die men associëert met dit type filters duidt op het "roll-off" karakter. Zoals uit figuur 14 is gebleken loopt het ruisspectrum van de $\Sigma\Delta$ op voor toenemende frequentie. Om toch voldoende onderdrukking te krijgen voor de ruis moeten we minstens een tweede orde integrate-and-dump filter nemen.

De systeemfunctie van een tweede orde integrate-and-dump filter wordt gegeven door:

$$H_2(Z) = \left(\frac{1}{N}\sum_{i=0}^{N-1} z^{-i}\right)^2 = \left(\frac{1}{N}\left(\frac{1 - z^{-N}}{1 - z^{-1}}\right)\right)^2$$

De frequentieresponsie staat afgebeeld in figuur 15.

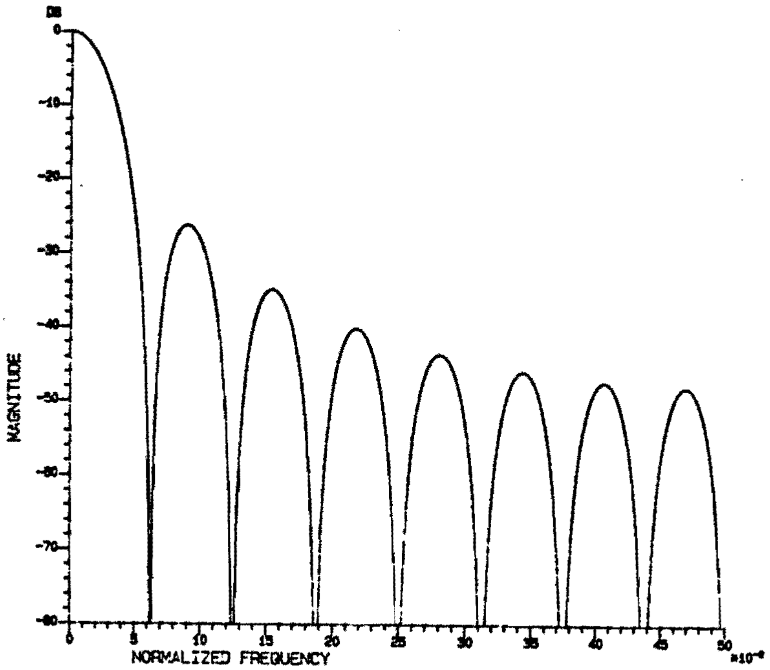
4.2.2 Realisatie

Wanneer we een tweede orde integrate-and-dump filter toepassen voor de ruisonderdrukking van de $\Sigma\Delta$ modulator, dan proberen we filteren en decimeren ineen te weven. Dit moet dan leiden tot een compact filter. We gaan daarbij uit van de impulsresponsie van een eerste orde integrate-and-dump filter die als volgt kan worden opgeschreven:

$$h_1[n] = \sum_{i=0}^{N-1} \delta[n - i] \qquad (6)$$

Hiermee kunnen we voor een tweede orde filter een impulsreponsie opschrijven, bestaande uit de convolutie van twee eerste orde impulsresponsies:

$$h_2[n] = h_1[n] * h_1[n] \qquad (7)$$



Figuur 15: Frequentieresponsie van het tweede orde integrate-and -dump filter (N = 16).

Wanneer we nu (6) invullen in (7) dan krijgen we:

$$h_2[n] = \sum_{i=0}^{N-1} \delta[n-i] * \sum_{j=0}^{N-1} \delta[n-j] \tag{8}$$

Uitwerken van de convulutie leidt tot:

$$h_2[n] = \sum_{k=0}^{N-1} \{ \sum_{i=0}^{N-1} \delta[k-i] \sum_{j=0}^{N-1} \delta[n-k-j] \} \tag{9}$$

$$h_2[n] = \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} \delta[n-k-j] \tag{10}$$

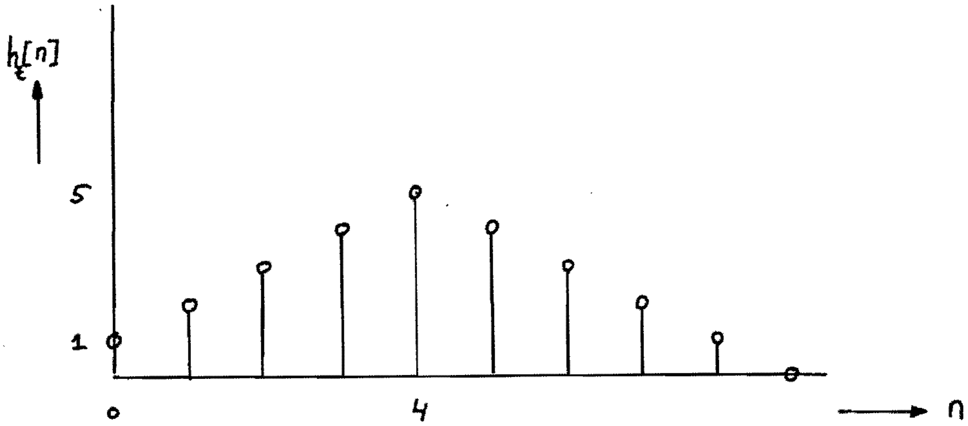
Bovenstaande vergelijking kan in de oorspronkelijke variabelen worden geschreven als:

$$h_2[n] = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \delta[n - i - j] \tag{11}$$

Beschouw $x[n]$ als het ingangssignaal en $y[n]$ als het uitgangssignaal van het filter dan geldt dat:

$$y[n] = \sum_{i=0}^{N-1} (i+1)x[n-i] + \sum_{i=0}^{N-1} (N-1-i)x[n-i-N] \tag{12}$$

De impulsresponsie heeft de vorm van een driehoek zoals uit bovenstaande vergelijking volgt. In figuur 16 is deze impulsresponsie geschetst. Het



Figuur 16: Impulsresponsie van een tweede orde integrate-and-dump filter (N = 5).

uiteindelijke doel is om het beschreven filter als decimerend filter uit te voeren, hetgeen betekent dat er nog een bemonsteringsfrequentie verlaging moet plaatsvinden. Dit kan impliciet met de filtering plaatvinden zoals we dat reeds bij het interpolerend filter zijn tegengekomen. Voor de theoretische beschouwing wordt verwezen naar [6]. Allereerst kiezen we de decimatiefactor M gelijk aan N, zodat we filteren en decimeren eenvoudig kunnen combineren. Globaal komt het er in dit geval op neer dat we N vertragingselementen nodig hebben voor de ingangsmonsters die met de coëfficiënten van het rechtse deel van vergelijking (12) vermenigvuldigd moeten worden. Een eerste implementatie van een decimerend tweede orde integrate-and-dump filter is gegeven in figuur 17. In deze figuur zien we linksboven een 1-bit schuifregister van lengte N die de bovenbedoelde ingangsmonsters vertraagd. Verder staat rechtsboven een een p-bit teller (met $p = \log_2 N$) die de coëfficiënten van de impulsresponsie maakt. Uitgaande van figuur 16 is het voorschrift dat de N vertraagde monsters achtereenvolgens vermenigvuldigd moeten worden met de coëfficiënten $h_t[2N-1-n]$ waarbij n loopt van 0 tot $N-1$. De deelprodukten $h_t[2N-1-n]x[n]$ worden bij elkaar opgeteld (geaccumuleerd). Als we in figuur 17 kijken dan zien we dat dit wordt gerealiseerd door de uitgang van het schuifregister met behulp van een (&) And-poort te vermenigvuldigen met de p-bit coëfficiënten uit de teller (een vermenigvuldiging van een 1-bit woord met een p-bit woord kan worden geimplementeerd met een And-poort). De uitgang van de And-poort gaat via een Or-poort naar de accumulator die de deelprodukten, zoals bovenvernoemd, opteld. De niet vertraagde monsters moeten achtereenvolgens worden vermenigvuldigd met de coëfficiënten $h_t[N-1-n]$ waarbij n ook loopt van 0 tot $N-1$. Dit gebeurt ook met behulp van een And-poort. De deelprodukten $h_t[N-1-n]x[n]$ worden ook via de Or-poort naar de accumulator geleidt waar ze bij de rest worden geteld. Het verschil met de vertraagde tak is dat de coëfficiënten van de niet vertraagde tak op de tijdstippen n het complement plus één zijn. In figuur 17 wordt deze extra één gemaakt door de waarde (0 of 1) van het ingangsmonster direct aan de opteller toe te voegen. De accumulator is gerealiseerd door een opteller en een geheugen (register) via een multiplexer terug te koppelen. De functie van de multiplexer is om de accumulator op nul te zetten nadat de deelprodukten zoals boven beschreven zijn geaccumuleerd. Het resultaat, de geaccumuleerde waarde, wordt dan doorgeschoven naar het resultaat register (onderaan in figuur 17). Dit resultaat register wordt steeds voorzien van een nieuwe waarde op de lage bemonsteringsfrequentie $f_{sl} = f_{sh}/N$.

Naast bovenbeschreven structuur mét vertragingselementen bestaat er ook een structuur zoals in figuur 18 wordt getoond die geen gebruik maakt van vertragingselementen maar van een dubbele accumulator. In deze structuur wordt de vertraging gerealiseerd door eerst met één accumulator het deel $h_t[2N - 1 - n]x[n]$ te berekenen. Na N accumulaties wordt het resultaat van deze accumulator doorgeschoven als startwaarde van de tweede accumulator. Vervolgens wordt in deze accumulator het resterende deel $h_t[N - 1 - n]x[n]$ berekend. Beide schakelingen zijn identiek, echter voor de implementatie zijn er enkele verschillen.

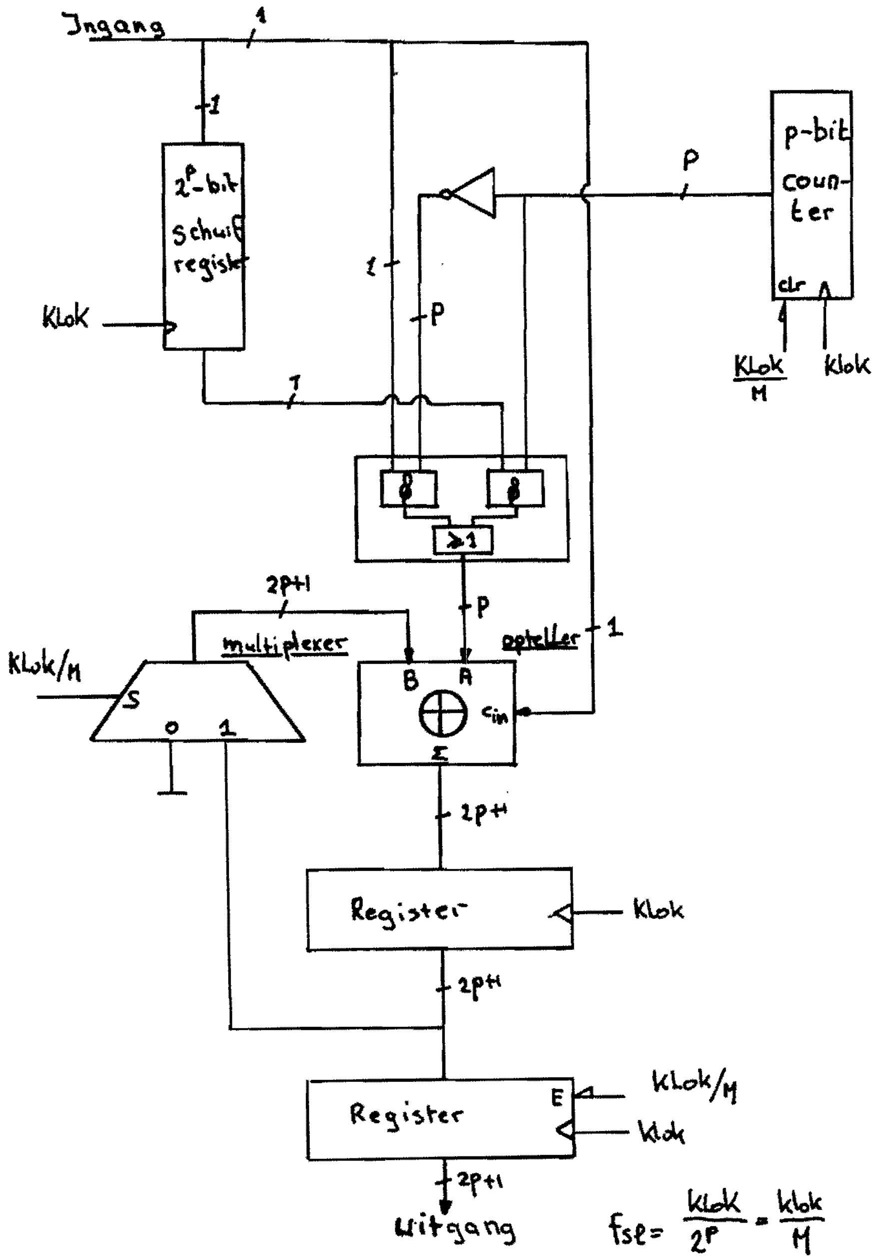
Het hangt van de decimatie factor af welke structuur het kleinste aantal componenten nodig heeft. Wanneer we als criterium het aantal elementaire componenten (C) nemen zoals flipflops, optellers etc. dan geldt voor de structuur met vertragingselementen:

$$C = 6\log_2(N) + 4 + N$$

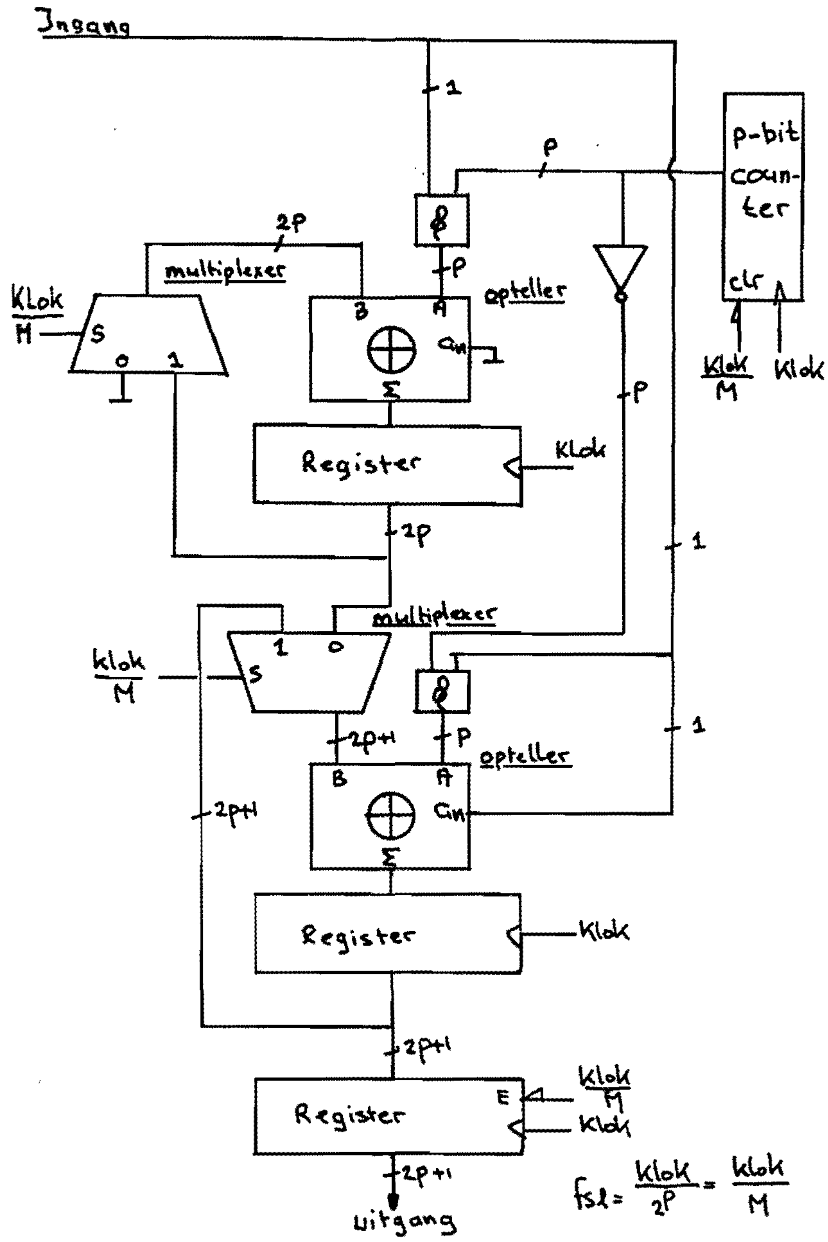
en voor de andere structuur:

$$C = 14\log_2(N) + 4$$

Wanneer $\log_2 N > 5$ dan is de laatste structuur met minder componenten te realiseren.



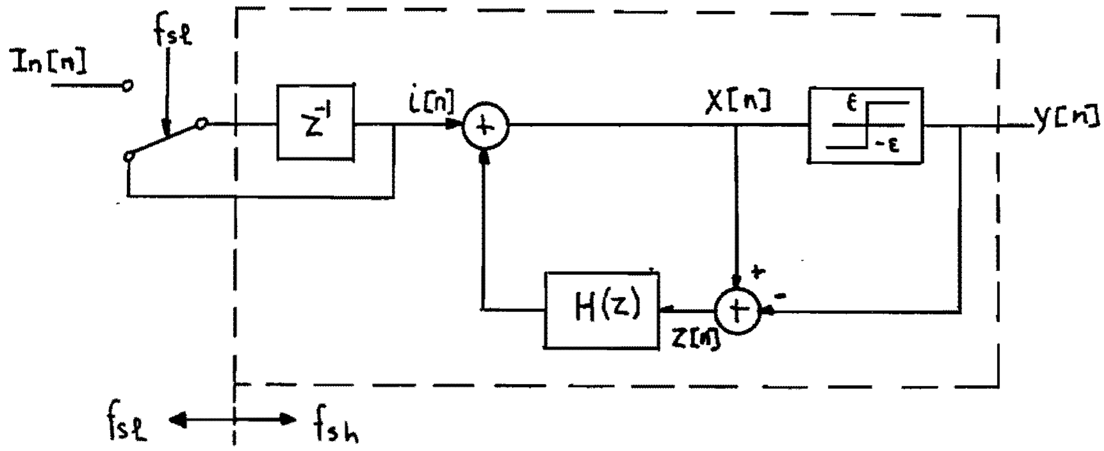
Figuur 17: Integrate-and-dump filter structuur met vertragingselementen.



Figuur 18: Integrate-and-dump filter structuur met dubbele accumulator.

4.3 De noiseshaper

Voor de noiseshaper kunnen we ongeveer dezelfde beschouwing hanteren als voor de $\Sigma\Delta$ modulator. Een noiseshaper kan bijv. worden toegepast als één bits D/A omzetter. In deze paragraaf wordt de noiseshaper beschreven zoals deze wordt toegepast in het volgende hoofdstuk. Er zijn echter nog vele varianten mogelijk die hier niet worden behandeld. Figuur 19 geeft het blokschema van de noiseshaper. Aan de ingang van de noiseshaper



Figuur 19: noiseshaper.

($In[n]$) worden de monsters aangeleverd met een bemonsteringsfrequentie f_{sl}. De noiseshaper bestaat verder uit een lusfilter $H(z)$ en een kwantisator. Beiden worden bedreven met een bemonsteringsfrequentie van f_{sh}. Het uitgangssignaal ($y[n]$), in onze beschouwing, is een één bits signaal. De werking van de kwantisator is als volgt: het meest beduidende bit wordt naar de uitgang doorgegeven, terwijl het overblijvende signaal (het ingangssignaal van de kwantisator waarvan het meest beduidende bit nul is) wordt doorgestuurd naar het lusfilter $H(z)$. In figuur 19 is dit gemodellerd d.m.v. de kwantisator (zoals toegepast bij de $\Sigma\Delta$ modulator) en het parallelle pad waar het uitgangssignaal van de kwantisator ($y[n]$) wordt afgetrokken van het signaal $x[n]$. Het doel van het lusfilter is nu om de ruis die wordt geintroduceerd door de kwantisator te filteren zodanig dat de ruisenergie in de band van $0 \ldots f_{sl}/2$ minimaal is.

In de toepassing van de hier beschreven noiseshaper wordt een eerste orde lusfilter gebruikt. Het is ook mogelijk hogere orde lusfilters te gebruiken die een beter onderdrukking van de kwantisatieruis geven. In dat geval wordt een stabiliteitsanalyse van de noiseshaper belangrijk en hiervoor zij verwezen naar [2].

Beschouwen we allereerst de kwantisator met het model zoals gebruikt in de $\Sigma\Delta$ modulator (zie vergelijking (2) pagina 19). Uit het blokdiagram van figuur 19 kunnen we afleiden dat het ingangssignaal van het lusfilter $z[n]$ geschreven kan worden als

$$z[n] = x[n](1-c) - r[n]$$

Voor het uitgangssignaal $Y(z) = Z\{y[n]\}$ van de noiseshaper geldt dan:

$$Y(z) = \frac{cI(z)}{1 - H(z)(1-c)} + \frac{(1-H(z))R(Z)}{1-H(z)(1-c)} \tag{13}$$

De bemonstering van het ingangssignaal $In[n]$ op een bemonsteringsfrequentie f_{sl} is in figuur 19 aangegeven door de keuzeschakelaar aan de ingang gevolgd door een vertragingselement. Gedurende één periode van de bemonsteringsfrequentie f_{sh}, met een herhalingsfrequentie f_{sl}, wordt het vertragingselement met het ingangssignaal $In[n]$ verbonden. Gedurende de andere perioden wordt het vertragingselement met zichzelf verbonden en geeft een constant signaal $i[n]$. Dit heeft als gevolg dat er reeds een $\sin x/x$ distorsie (sample and hold met een eerste nulpunt op f_{sl}) over het signaal $i[n]$ zit. Het lusfilter, de kwantisator, $i[n]$ en het uitgangssignaal $y[n]$ worden bemonsterd met een bemonsteringsfrequentie f_{sh}. Ook hier nemen we aan dat de verhouding tussen f_{sh} en f_{sl} gelijk is aan de factor M.

Om nu een eis te kunnen formuleren voor het karakter van het lusfilter $H(z)$ voeren we eerst de substitutie $z = e^{j\theta}$ in. Voor het gewenste frequentiegebied van het uitgangssignaal $y[n]$ nemen we aan dat $\theta_1 \leq \frac{\pi}{M}$. Wanneer nu ervoor wordt gezorgd dat $H(e^{j\theta}) \approx 1$ is voor $\theta \leq \theta_1$ dan kunnen we voor (13) schrijven:

$$Y(e^{j\theta}) = I(e^{j\theta}) + \frac{(1-H(e^{j\theta}))R(e^{j\theta})}{c}$$

We zien dat het signaal praktisch onvervormd door de noiseshaper heen gaat, terwijl de kwantisatieruis een zodanige filtering ondergaat dat in het gewenste frequentiegebied $\theta \leq \theta_1$ weinig ruisenergie overblijft. Een klasse van loopfilters die aan de gestelde eis kunnen voldoen zijn te schrijven als

$$H(z) = \frac{(z-a)^N - (z-b)^N}{(z-a)^N}$$

waarvoor geldt dat $-1 < a \leq b \leq 1$ en N de orde van het filter aangeeft. Met de beperking tot een eerste orde noiseshaper ($N = 1$) en de keuze $a = 0$ en $b = 1$ krijgen we een eenvoudige vertraging als loopfilter. De systeemfunctie wordt gegeven door:

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

Voor deze noiseshaper gelden dezelfde beschouwingen als bij de $\Sigma\Delta$ modulator t.a.v. het ruisgedrag, het spectrum van de kwantisatieruis (zie figuur 14) en de verwachte signaal ruisverhouding in het gewenste frequentiegebied. Met een eenvoudig analoog filter achter deze noiseshaper kan men de ongewenste kwantisatieruis weg filteren. Vergelijken we het lusfilter van de eerste orde $\Sigma\Delta$ modulator met het lusfilter van bovenstaande noiseshaper dan blijkt te gelden:

$$H(z) = \frac{G(z)}{1 + G(z)}$$

Een D/A converter op basis van een noiseshaper met een 16 *bits* resolutie is reeds gerealiseerd voor HiFi audio toepassingen [5].

5 Het digitale CD-servosysteem

In dit hoofdstuk wordt een digitale implementatie van de focusservo behandeld. Allereerst worden de eisen geformuleerd die we aan de regeling stellen. Daarna wordt een implementatie gegeven.

5.1 Eisen

De eisen die we aan de digitale focusregeling moeten stellen volgen voor een deel uit de specificatie van de plaat, en voor een ander deel uit de te verwachten invloeden van buiten (schokken). Er bestaat een document [7] waarin men alle gegevens t.a.v het compact disc systeem (plaat plus speler) kan vinden. Hieronder volgt een opsomming van de specificatie punten waaraan de focusregeling moet voldoen.

- de spot moet in focus zijn in een gebied van $\approx 1\,\mu m$
- plaatspecificatie :
  - maximale vertikale afwijking $\pm 1\,mm$
  - maximale vertikale acceleratie $(10g\, tot\, 500\,Hz)$ $(g = 10\,m/s^2)$
- bandbreedte focusregeling: $800\,Hz$. De reduktie curve wordt bepaald door het loopfilter.
- fasemarge van de regeling ≈ 50 graden
- voor de bemonsteringsfrequentie f_{sl} van de digitale regelaar wordt $33\,kHz$ gekozen.

De bandbreedte en de reduktiecurve volgen direct uit de plaatspecifikatie. In eerste instantie lijkt de bemonsteringsfrequentie f_{sl} nogal hoog voor een te realiseren bandbreedte van slechts $800\,Hz$. Om de bemonsteringsfrequentie te kunnen bepalen moeten we een rekensommetje maken, waarin alle extra fasedraaiingen (bijv. sample en hold en rekendelay) tot uitting komen. Stel dat we als A/D omzetter een $\Sigma\Delta$ modulator met (decimerend) integrate- and- dump filter (afgekort: I & D) kiezen zoals in het vorige hoofdstuk is behandeld. De bemonsteringsfrequentie aan de uitgang van het decimerend filter f_{sl} kiezen we gelijk aan bovengenoemde $33\,kHz$, en de decimatie factor $M = 64$. De extra fasedraaiing die aan het regelcicuit wordt toegevoegd kunnen we opgebouwd denken uit:

delay t.g.v I & D filter	8.7°
rekendelay (bijv. $1/5 f_{sl}$)	1.7°
hold delay bij $800 Hz$	4.4°
dit geeft een totaal van	14.8°

Vergelijken we dit met een analoge regeling dan moet er er ongeveer 15° meer "opgehaald" worden om een vergelijkbare fasemarge te halen. Kiezen we i.p.v. een $\Sigma\Delta$ modulator een gewone A/D omzetter met een conversietijd van bijv. $5\,\mu s$. En verder een eerste orde prefilter om vouwvervorming tegen te gaan met bijv. $20\,dB$ demping op f_{sl} dan ziet het plaatje er als volgt uit:

delay t.g.v A/D en prefilter	15°
rekendelay (bijv. $1/5 f_{sl}$)	1.7°
hold delay bij $800 Hz$	4.4°
dit geeft een totaal van	21.1°

We zien hier duidelijk een voordeel van de $\Sigma\Delta$ modulator t.o.v. de gewone A/D omzetter. In de volgende paragraaf wordt er iets verder ingegaan op de keuze van het 1-bits A/D systeem met $\Sigma\Delta$ modulator.

5.2 Foutsignaal opwekking

Voor de foutsignaal opwekking staan er vier detector signalen ter beschikking die na lineaire combinatie het gewenste foutsignaal opleveren (zie vergelijking 1 pagina 7). Om deze signalen te digitaliseren maken we gebruik van de eerder behandelde $\Sigma\Delta$ modulator gevolgd door een tweede orde integrate-and-dump filter. In het hoofdstuk over het CD-systeem is er al gesproken over de optische afsnijfrequentie van ongeveer $1,4\,Mhz$ wanneer de plaatsnelheid $1,25\,m/s$ bedraagt. Dit is belangrijk i.v.m. de keuze van de bemonsteringsfrequentie f_{sh} van de $\Sigma\Delta$ modulator. Immers we willen geen ongewenste vouwvervorming in de signaalband hebben, en liefst ook geen filter toepassen. In dit geval, met een signaal bandbreedte van

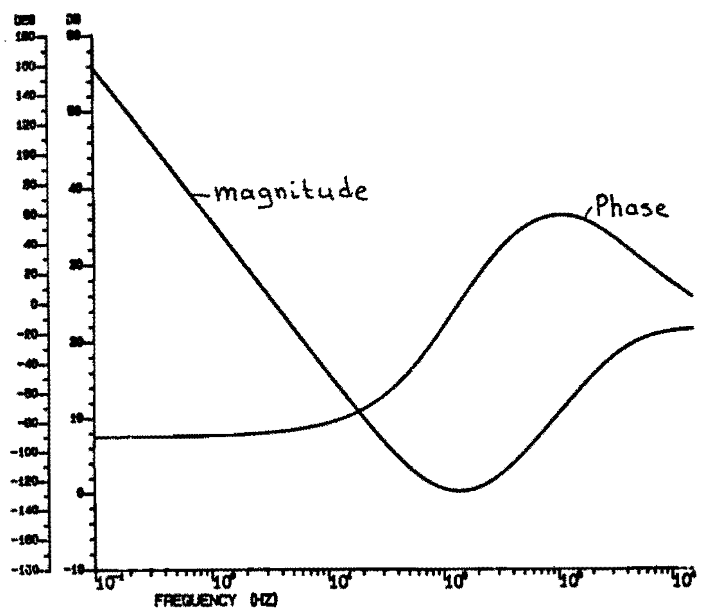
$33/2\,kHz$, kunnen we f_{sh} gelijk nemen aan $64*33\,kHz = 2,112\,Mhz$, zonder dat we vouwvervorming introduceren. Daarmee is tevens de decimatiefactor M vastgelegd op 64 (voor het decimerend filter geldt dan ook $N = 64$). Door middel van een berekening of eenvoudig opzoeken in een tabel (zie bijv. [4]) vinden we dat de signaal ruisverhouding die wordt verkregen met het I & D filter $69\,dB$ bedraagt in een bandbreedte tot $4\,kHz$. Voor de ruisbandbreedte hoeven we in dit geval niet de nyquist bandbreedte $f_{sl}/2$ te nemen, immers alleen de ruis die in de regellus (met een bandbreedte van $800\,Hz$) wordt geinjecteerd is van belang. De reden dat $4\,kHz$ wordt gekozen is om niet te veel buiten de band ruis in de lus te introduceren, immers de focusspoel gedraagt zich als een luidspreker.

5.3 De digitale regelaar

De analoge focusregelaar in de compact disc spelers is uitgevoerd als PID regelaar. De focus actuator is een tweede orde systeem: de stroom in de spoel zorgt voor een versnelling van de actuator, terwijl de focusregeling een positie- of plaatsregeling is. De "D-actie" van de regelaar zorgt voor de stabiliteit en de "I-actie" regelt de statische fout naar nul. De PID regelaar, uitgevoerd als digitale regelaar is te beschrijven met de systeemfunctie:

$$F(z) = K_p + \frac{K_i z}{z-1} + \frac{K_d(z-1)}{z-c} \tag{14}$$

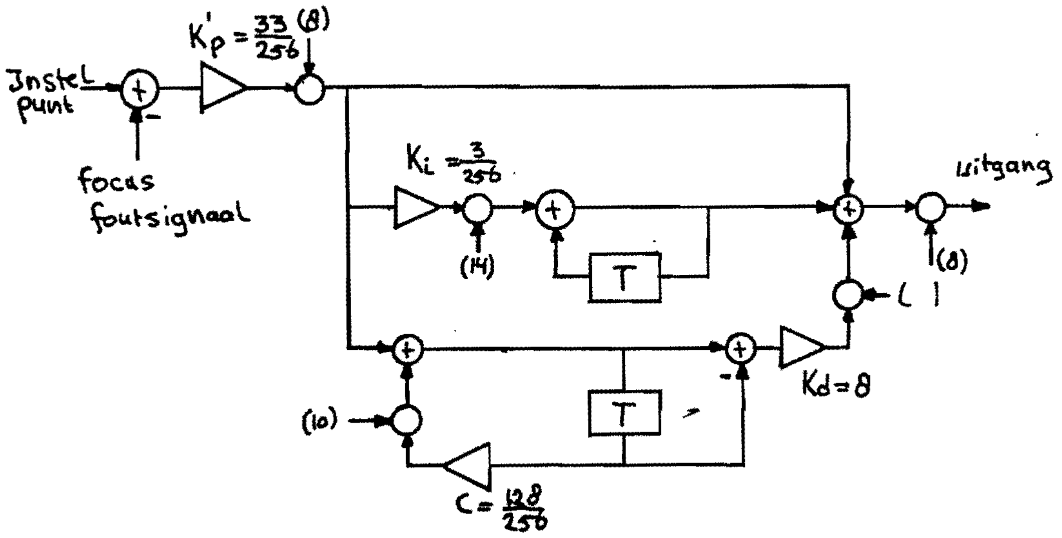
Wanneer deze regelaar als *parallelle PID sectie* wordt uitgevoerd dan is een van de voordelen van deze uitvoering dat de P, I en D actie onafhankelijk beinvloedt kunnen worden door de keuze van K_p, K_i en K_d. De differentiërende actie wordt vanaf een bepaalde frequentie gestopt ($c \neq 0$) om geen ongewenste ruis op te halen voor hoge frequenties. Verder blijkt dat deze filterstructuur vrij ongevoelig is voor kwantisatie van de coefficienten. Figuur 20 geeft de frequentieresponsie van het PID filter weer. De ligging van de kantelpunten wordt bepaald door de eisen zoals geformuleerd in paragraaf 5.1. De filterstructuur van het focus PID filter staat afgebeeld in figuur 21. Een verschil met formule (14) is dat de proportionele versterking als algemene versterkingsfactor vóór de sectie is geschakeld. Dit heeft als voordeel dat de bandbreedte van de regeling ingesteld kan worden met coëfficiënt K_p' zonder dat de kantelpunten worden beinvloed.



Figuur 20: Frequentieresponsie van het PID filter in de focus regeling.

5.4 De totale focusregeling

Om de gehele regeling in het $\mathcal{Z}$ domein te kunnen doorrekenen moet de actuator van het $\mathcal{L}$ domein naar het $\mathcal{Z}$ domein worden getransformeerd. Dit wordt gedaan met behulp van de impuls invariante transformatie zoals die in de literatuur te vinden is [6]. Deze transformatie heeft de eigenschap dat de impulsresponsie behouden blijft, maar dat er wel vouwvervorming kan optreden. De verhouding tussen bemonsteringsfrequentie en de dynamische eigenschappen van de actuator is zó groot, dat eventuele effecten t.g.v. vouwvervorming verwaarloosbaar zijn. Reden echter om toch déze transformatie te kiezen is dat het sample and hold effect van de D/A functie in de simulaties kan worden gemodelleerd. Stel dat men de veel toegepaste bilineaire transformatie wil gebruiken. We gaan uit van een tijdcontinu model van focusactuator en postfilter waarvan de Laplacegetransformeerde wordt gegeven door $H(s)$. Het ingangssignaal noemen we $x(t)$. Het ingangssignaal $x(t)$ sturen we eerst door een nulde orde houdschakeling waarvan de



Figuur 21: Filterstructuur van het PID filter in de focus regeling.

Laplacegetransformeerde wordt gegeven door:

$$\frac{1 - e^{-sT}}{s}$$

De Laplacegetransformeerde van het totale systeem $G(s)$ wordt gegeven door

$$G(s) = \frac{1 - e^{-sT}}{s} H(s)$$

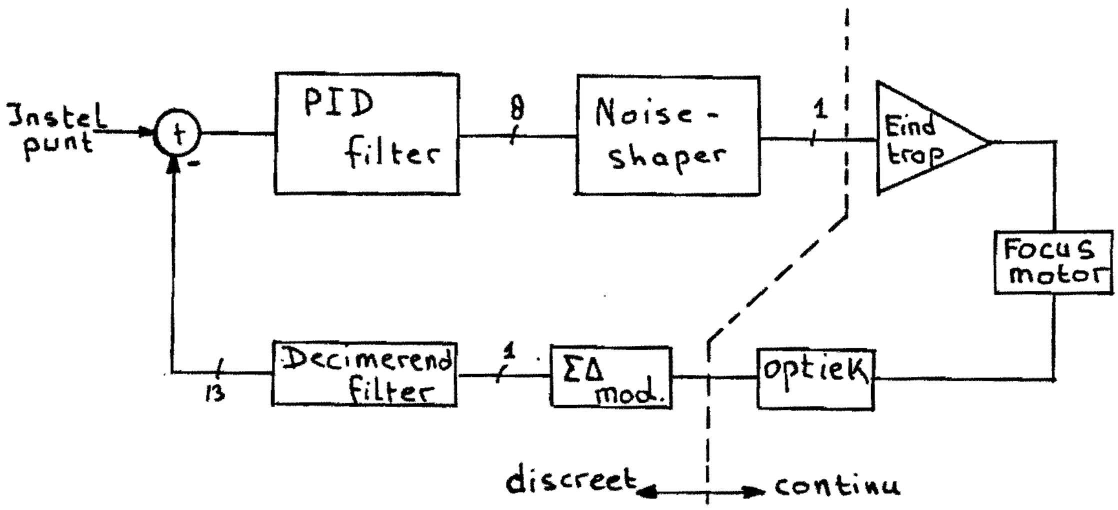
Het blijkt nu dat de Z getransformeerde van $g[n]$ niet kan worden bepaald met behulp van de bilineaire transformatie. De exponentiële term kan met de bilineaire transformatie niet worden getransformeerd. Met de impulsinvariante transformatie kan men voor de transformatie opschrijven:

$$Z\{g[n]\} = 1 - z^{-1} Z\{g'[n]\}$$

waarbij:

$$G'(s) = \frac{1}{s} H(s)$$

Om nu de regellus te kunnen sluiten missen we nog een component en dat is de D/A convertor. De D/A convertor is uitgevoerd als een eerste orde noiseshaper, waarbij als woordbreedte aan de uitgang 1-bit genomen is. De woordbreedte aan de uitgang van het PID filter zoals zoals aangegeven in figuur 21 bij $f_{st} = 33\,kHz$ is $8\,bits$ en wordt via de noiseshaper omgezet in een 1-bits signaal op $2.12\,MHz$. Dit gaat via het principe zoals behandeld in paragraaf 4.3, en men kan met een eenvoudig analoog laagdoorlaatfilter achter de noiseshaper de vereiste SNR bereiken. Het blokschema van de totale regeling staat afgebeeld in figuur 22. Het verkregen signaal aan de uitgang van het postfilter kan direct worden toegevoerd aan een vermogensversterker die de actuator aan kan sturen. Met deze realisatie is



Figuur 22: Complete digitale focus regeling.

aangetoond dat het zeer goed mogelijk is om A/D en D/A conversie voor dit soort regelsystemen tot een minimum te beperken. Reden om deze beperking na te streven is dat, wanneer men denkt aan realisatie van conventionele A/D en D/A functies op een volledig digitale chip, men daar toch voorzorgsmaatregelen moet treffen t.a.v. de layout in verband met overspraak. Met de hier getoonde realisatie, waarbij gebruik is gemaakt van $\Sigma\Delta$ modulatoren en noiseshapers, is dit probleem aanzienlijk kleiner.

Referenties

[1] Groen R.W.C., "On the calculation of quantization noise in digital filters caused by magnitude truncation" Nat. Lab. Report no 6188

[2] Stikvoort E.F. "Some Remarks on the Stability and Performance of the Noise Shaper or Sigma-Delta Modulator" *IEEE trans. on comm.* vol. COM 36 pp. 1157-1162. Oct. 1988.

[3] Candy J.C. et al. "The Structure of Quantization Noise from Sigma-Delta Modulation" *IEEE trans. on comm.* vol. COM 29, pp. 1316-1323. Sept. 1981.

[4] Candy J.C. "Decimation for Sigma-Delta Modulation" *IEEE trans. on comm.* vol. COM 34, pp. 72-76. Jan. 1986.

[5] Naus P.J. et al "A CMOS stereo 16-bit D/A converter for digital audio" *IEEE J. Solid State Circuits* vol. SC 22, pp. 390-395. June 1987.

[6] van den Enden A.W.M, VerHoeckx N.A.M. "Digitale Signaalbewerking" *Delta Press Amerongen.*

[7] Compact Disc Redbook.

DIGITALE SIGNAALBEWERKING

deel II.10

Signaalbehandeling bij de DCC

door

A.W.M. van den Enden

Philips Nat. Lab. Eindhoven

SIMPLE FILTER STRUCTURES FOR DCC

1. Introduction
2. The DCC system
3. First generation DCC
4. Alternatives
5. Conclusions

BITSTREAM D/A CONVERSION

1. Oversampling
2. Noise shaping
3. Bitstream conversion

THE DIGITAL LOUDSPEAKER SYSTEM

Digital Compact Cassette

- Extension of well known compact cassette system
- Improved digital sound quality
- Uses highly efficient Precision Adaptive Subband Coding
- Digital multitrack recording
- Error correction techniques
- New thin film head design
- Autoreverse in all players
- New features

PHILIPS

E.D|FOTO

DIGITAL
dcc
COMPACT CASSETTE

I18-03-0.IMA

Design philosophy

- Market research indicates compatibility is key issue
- System should be attractive for the software industry
- Use video tape that is produced in mass volume, so minimum wavelength limited to about 1 micron
- Use compact cassette mechanism; this means the tape speed is 4.76 cm/sec.
- For azimuth tolerance minimum number of tracks

Result: 8 tracks, bitrate 96 kbits/sec/track, minimum wavelength 0.99 micron

PHILIPS

E.D|FOTO

DIGITAL dcc COMPACT CASSETTE

I18-04-0.IMA

Performance

- 2 audio channels,
  sampling frequency = 32, 44.1 or 48 kHz
- Emphasis (15/50 microsec.) optional
- Dynamic range better than 105 dB
- Signal to noise ratio up to 92 dB
- THD + noise less than 0.0025%
- Wow and flutter crystal controlled
- Recording time up to 2 times 45 minutes,
  with future option of 2 times 60 minutes

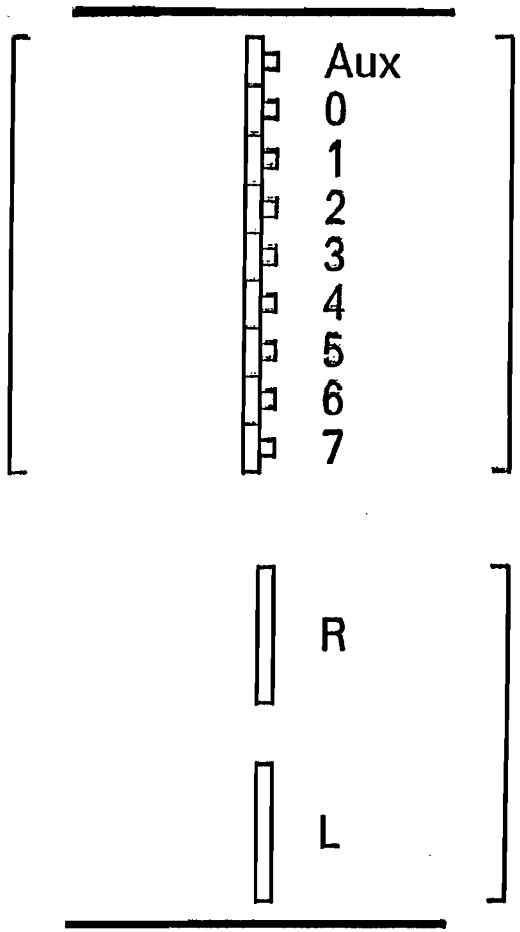
DCC cassette

- Compatible with compact cassette, same basic dimensions
- However, with a slimmer, smoother look
- Only one side has holes for tape drive hubs
- Other side available for text and/or graphics
- Slider protects tape and locks reels during transport

DCC head design

Record

9 IRH heads
Digital record
185 microns



Playback

9 MRH heads
Digital playback
70 microns

2 MRH heads
Analog playback
600 microns

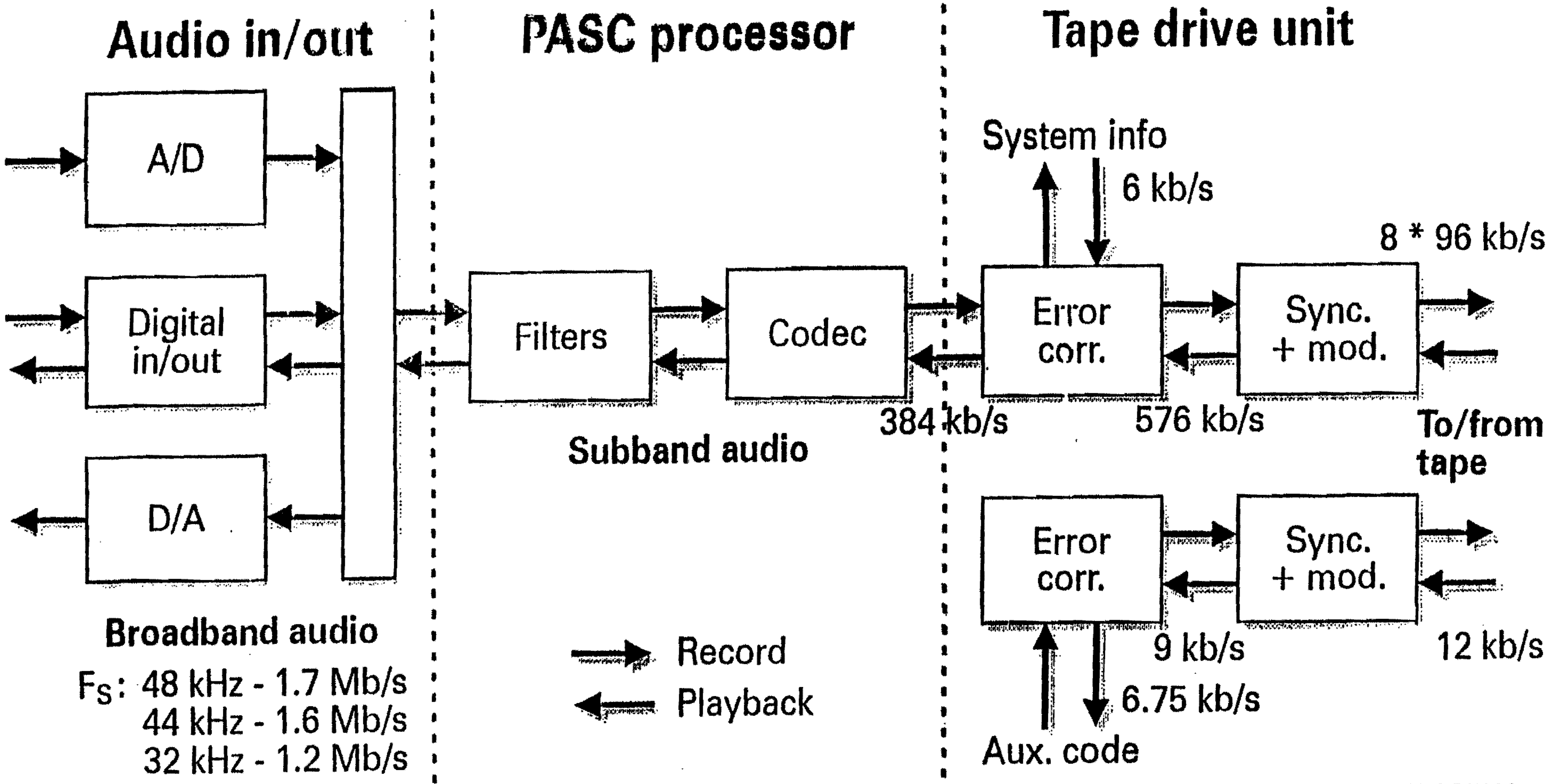
PHILIPS

E.D|FOTO

DIGITAL dcc COMPACT CASSETTE

I18-07-0.IMA

Block diagram



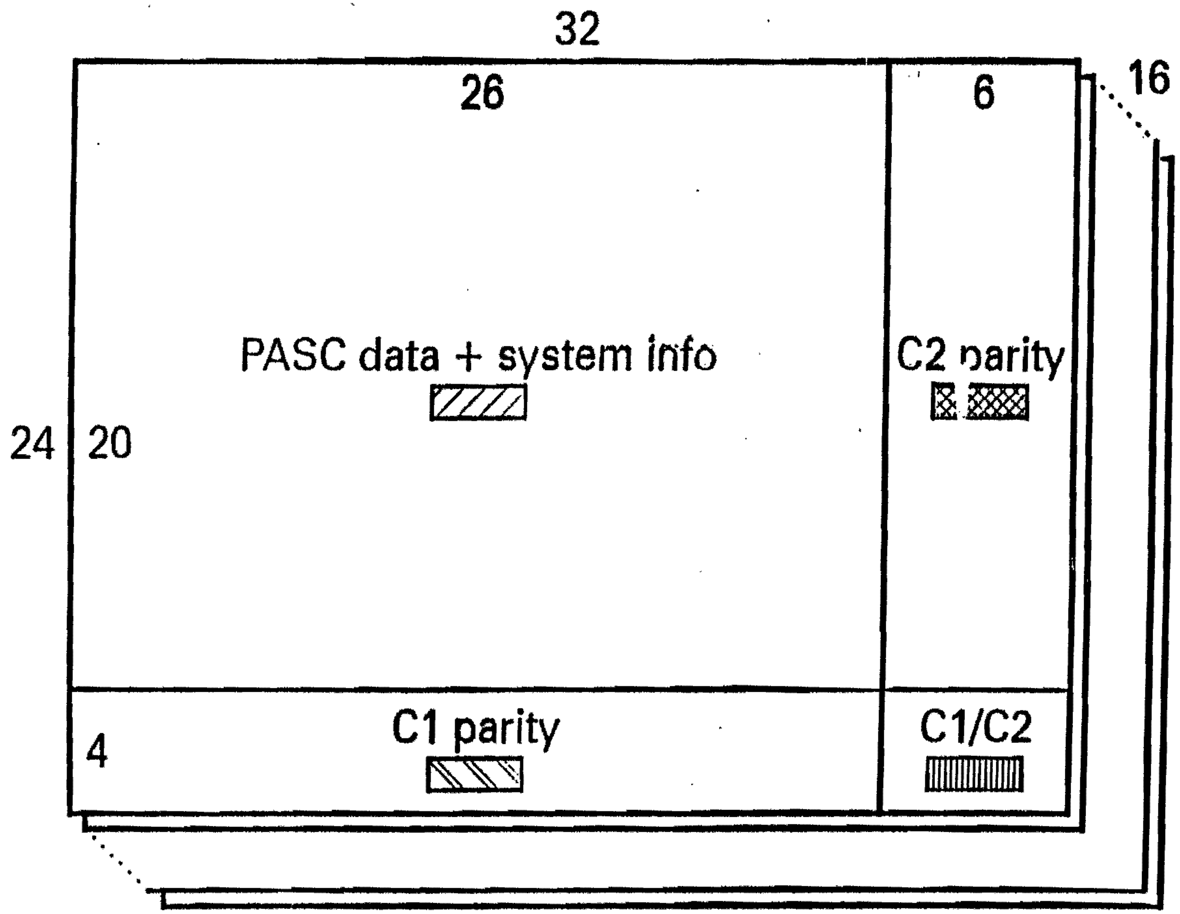
PHILIPS

E.D|FOTO

DIGITAL dcc COMPACT CASSETTE

I18-08-0.IMA

Error correction product code matrixes



Bytes

Data	520
C1	104
C2	120
C1/C2	24
	768
	* 16
	12288

PHILIPS

DIGITAL
COMPACT CASSETTE

Conclusions

- With DCC a new era in audio reproduction has started in which the attention is focussed on what the human ear really hears

- Being compatible with Compact Cassette DCC offers the best opportunity available for consumers and industry to enter into the field of digital recording

- DCC is expected to become a mass market product offering new opportunities to the software industry for music distribution

PHILIPS

Financieel dagblad

ONDERNEMINGEN BEDRIJFSTAKKEN

Zaterdag 2 en maan

Vakpers prefereert Philips' DCC boven Sony's MiniDisc

De audiovakpers roemt de geluidskwaliteit van de nieuwe digitale cassette (DCC) van Philips, maar is vooralsnog teleurgesteld in de geluidskwaliteit en uitrusting van de eveneens nieuwe MiniDisc van Sony. Dat kan worden geconcludeerd uit de eerste recensies van beide geluidsformaten, die sinds enige tijd verkrijgbaar zijn.

In de markt is met spanning gewacht op de eerste recensies, omdat Philips en Sony in een felle concurrentiestrijd zijn gewikkeld om de gunst van de consument die zijn bestaande cassettes moet gaan vervangen door een nieuwe geluidsdrager. Philips heeft daarbij gekozen voor de Digital Compact Cassette (DCC), in essentie een digitale opvolger van de bestaande geluidscassette. Sony heeft gekozen voor een klein formaat cd waarop kan worden opgenomen. Philips heeft in eerste instantie alleen een recorder voor in de huiskamer uitgebracht, terwijl Sony juist met portable spelers de consument tracht te veroveren. De Sony-spelers (waarvan er een ook kan opnemen) kosten tussen de ƒ 1100 en ƒ 1400, terwijl de Philips DCC-recorder in de winkel ligt voor rond de ƒ 1500 tot ƒ 1600.

Met name de Britse vakpers laat zich zeer kritisch uit over de MiniDisc. De geluidskwaliteit haalt bij lange na niet het niveau van de compact disc en wordt door het blad Hifi Choice omschreven als 'vlak' met een 'onnatuurlijk hoog'. 'In een vergelijking met een portable cd-speler wordt duidelijk dat het nieuwe medium bepaald niet waarmaakt waarop is gehoopt'. Sony heeft weinig reden om zelfgenoegzaam te zijn en het is moeilijk niet te denken dat de MiniDisc Sony's laatste grote misser zal blijken te zijn, net zoals Betamax en de El-cassette.'

De geluidskwaliteit komt in de regel niet verder dan die van de analoge geluidscassette, waarvan de MiniDisc nu juist de opvolger moet zijn, zo stelt Hifi Choice. 'De geluidskwaliteit van Philips' DCC is erg goed en, naar mijn mening, superieur aan wat Sony heeft te bieden.' Ook het Duitse vaktijdschrift Audio is teleurgesteld: 'De eerste geteste MD-speler komt wat klank en meetresultaten betreft niet verder dan een standaard cassettedeck in de DM 500 prijsklasse.'

Ook het stroomverbruik van de MD-spelers is nog een bron van ergernis. De zware oplaadbare batterijen gaan niet langer mee dan 75 minuten of één volledig afgespeelde MiniDisc, zo schrijft Audiophile. Wil men opnemen, dan wordt de opnametijd zelfs verkort tot 60 minuten. Ook raakt de speler soms de kluts kwijt bij het aangeven van titels van discs en nummers en denkt de speler soms nog met een vorige disc van doen te hebben.

Bij al deze kritiek moet worden aangemerkt dat Sony nooit heeft beweerd dat de geluidskwaliteit van de MiniDisc gelijk is aan die van de cd, maar dat hij deze 'benadert'. Bovendien betreft het de eerste modellen. Binnen een jaar zullen er goedkopere versies op de markt zijn die meer kunnen, veel lichter zijn, minder stroom verbruiken en geen introductiefoutjes meer hebben.

Maar dat neemt niet weg dat met de eerste produkten van beide fabrikanten, die vooral door 'audiofielen' worden gekocht, wel de trends worden gezet, en dat consumenten, verwend door cd-geluidskwaliteit, van een nieuw en nog duur medium minimaal dezelfde geluidskwaliteit zullen verwachten.

In de vakpers wordt opgemerkt dat de indruk heerst dat Sony de spelers te snel op de markt heeft gebracht om met de kerst iets in de winkel te hebben.

De DCC-recorder die Philips op de markt heeft gebracht, wordt alom geprezen. Luisterpanels van de Britse The Sunday Times waren in een directe vergelijking van DCC, MD, cd, analoge cassettes en gewone lp's erg enthousiast over de geluidskwaliteit van het Philips-medium. Het Nederlandse tijdschrift Home Studio omschrijft DCC als 'een perfecte opvolger van het analoge cassettedeck'. 'De testresultaten, zowel, op het gehoor als meettechnisch-liegen er niet om', zo schrijft het blad. 'Er is dan ook alle reden om aan te nemen dat DCC een even grote toekomst tegemoet gaat als zijn illustere voorganger.'

Conclusie

Onze ervaringen met de DCC-900 recorder hebben duidelijk gemaakt dat Philips met DCC een perfecte opvolger van het analoge cassettedeck heeft ontwikkeld. Want hoewel ons testapparaat een exemplaar is uit de 0-serie, bleek het in alle opzichten voortreffelijk te functioneren en liegen de testresultaten - zowel op het gehoor als meettechnisch - er niet om.

Er is dan ook alle reden toe om aan te nemen dat DCC een even 'grote' toekomst tegemoet gaat als zijn illustere voorganger. Want nog dit jaar kan iedere hifi- en muziekliefhebber digitaal opnemen op - tenminste - het kwaliteitsniveau van Compact Disc en DAT, terwijl het bovendien mogelijk is om op hetzelfde apparaat analoge cassettes met uitstekende kwaliteit weer te geven.

Ons rest alleen onze lezers nog enkele maanden geduld te vragen en Philips (en uiteraard Matsushita) te complimenteren met de bijna ongelofelijke prestaties van 'Digital Compact Cassette.

Fabrikant: Philips Consumer Electronics B.V. Eindhoven.
Prijs: N.N.B.

Eindhovens Dagblad: 12-6-1993

Kattegejank op mini disc

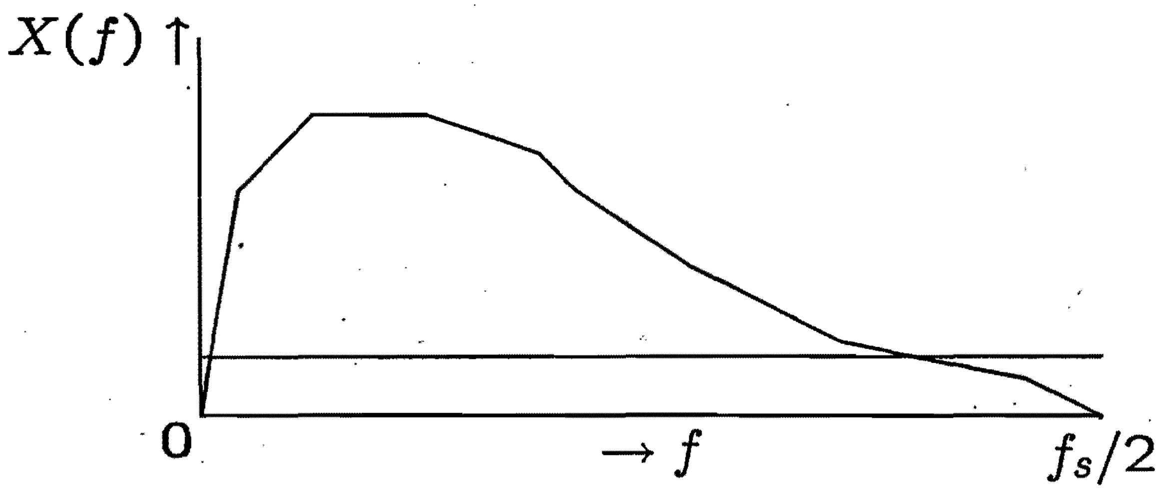
De introductie van de digitale compact cassette (DCC) – de opvolger van de analoge geluidscassette – verloopt volgens Philips driemaal zo snel als de succesvolle introductie van de compact disc tien jaar geleden.

Terwijl het Philips-systeem door de vakpers wordt geprezen krijgt de mini disc (MD) – het concurrerende systeem van Sony – de ene klap na de andere. Opvallend is dat het Japanse Matsushita – bondgenoot van Philips bij de ontwikkeling en introductie van de DCC – daarbij een aardig woordje meespreekt. Matsushita liet onlangs de vijfde generatie horen van een MD-kopie, waarbij telkens een kopie was gemaakt van een kopie. Het resultaat was (voor Sony) ontluisterend: kattegejank, aldus een toehoorder.

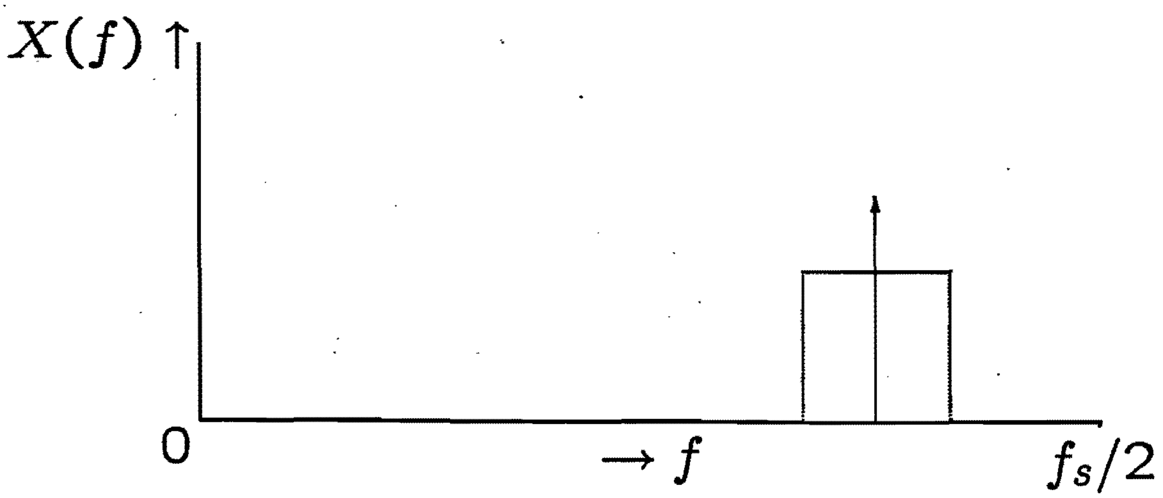
Insiders in de branche raken er steeds meer van overtuigd dat Sony zich bij de ontwikkeling van de mini disc teveel heeft laten opjagen. Als gevolg daarvan zou het Japanse concern een produkt met de nodige tekortkomingen op de markt hebben gebracht.

INTRODUCTION

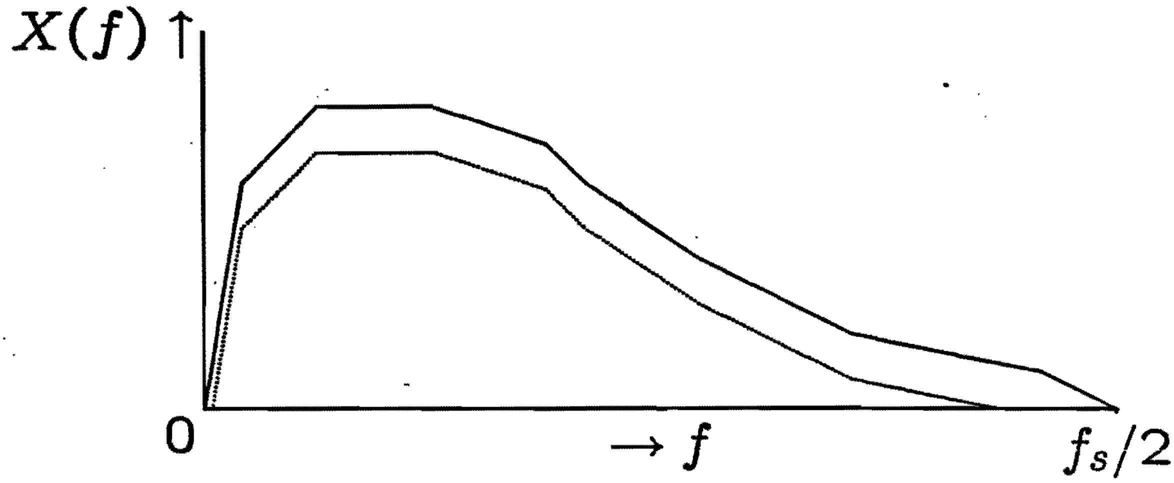
Signal spectrum with PCM noise (Compact Disc)



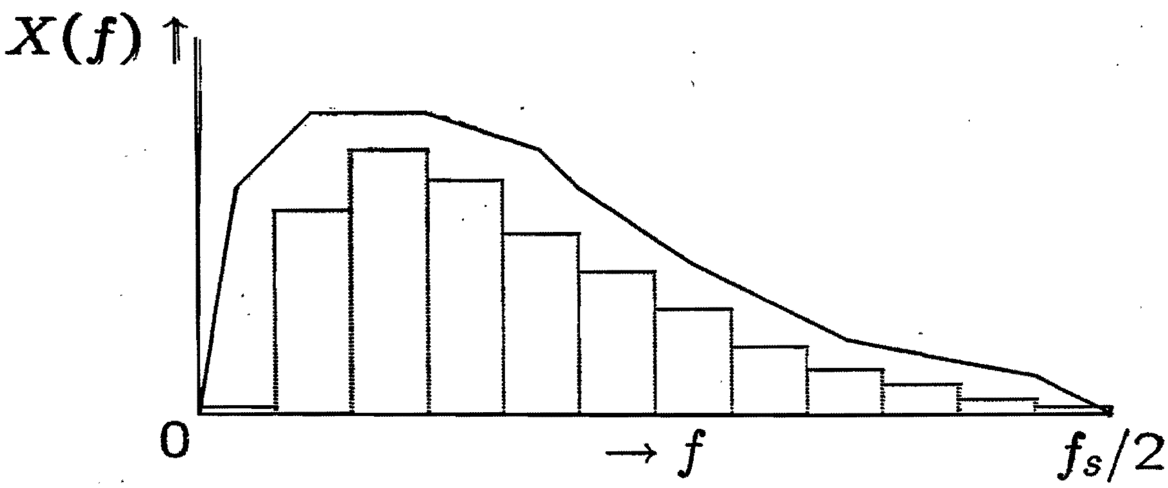
The masking principle

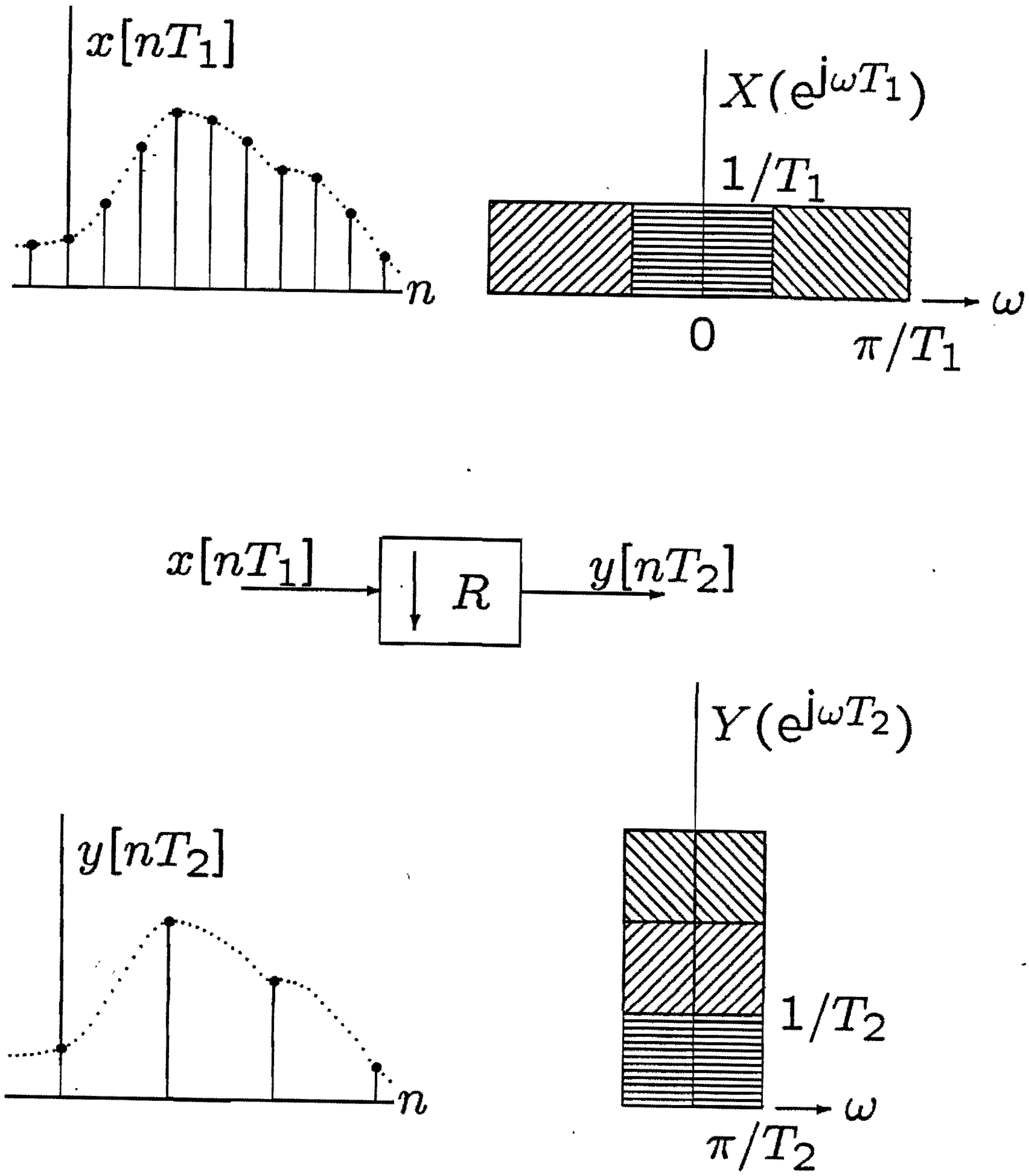


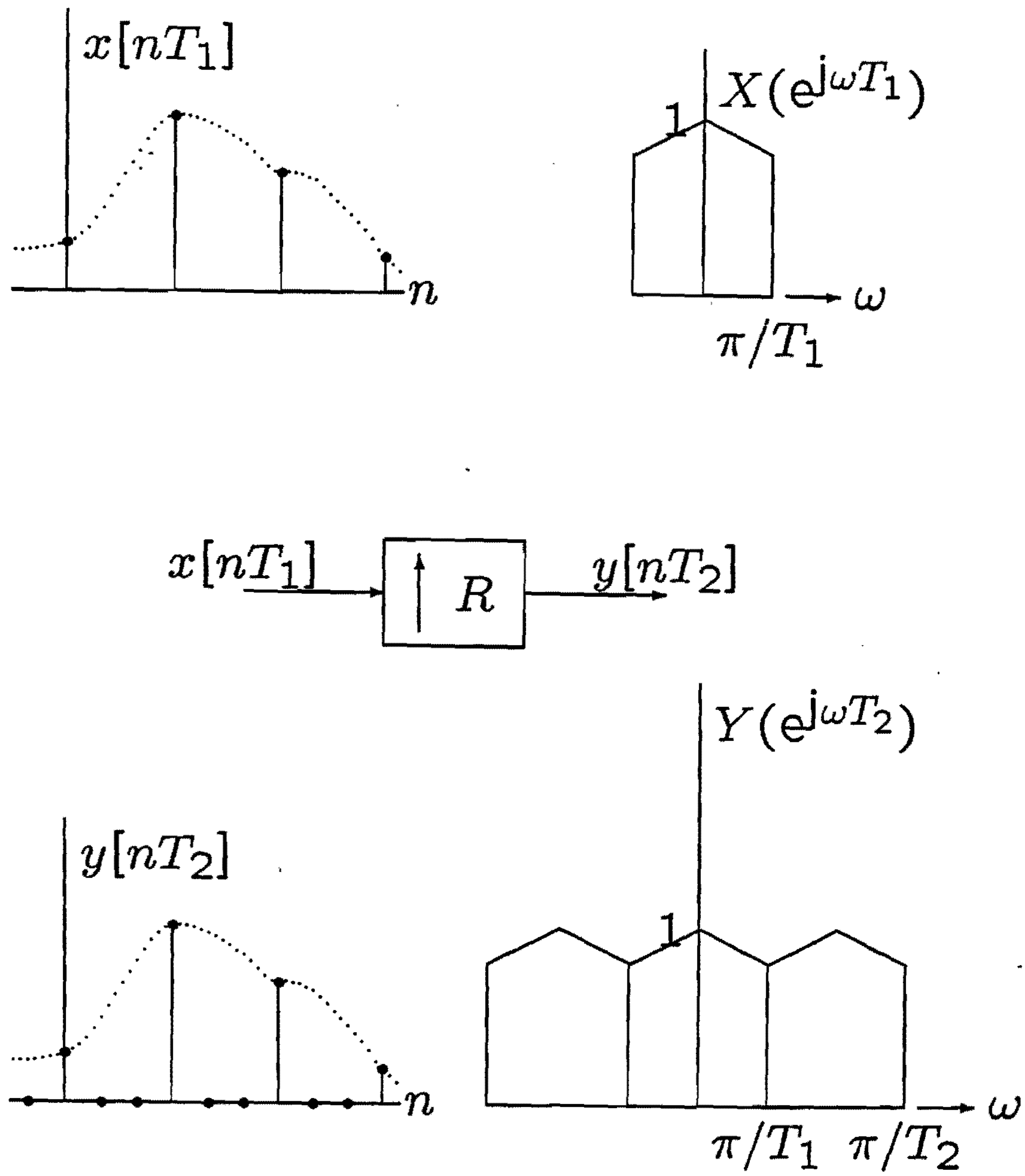
Signal spectrum with maximum noise



Signal spectrum with subband noise

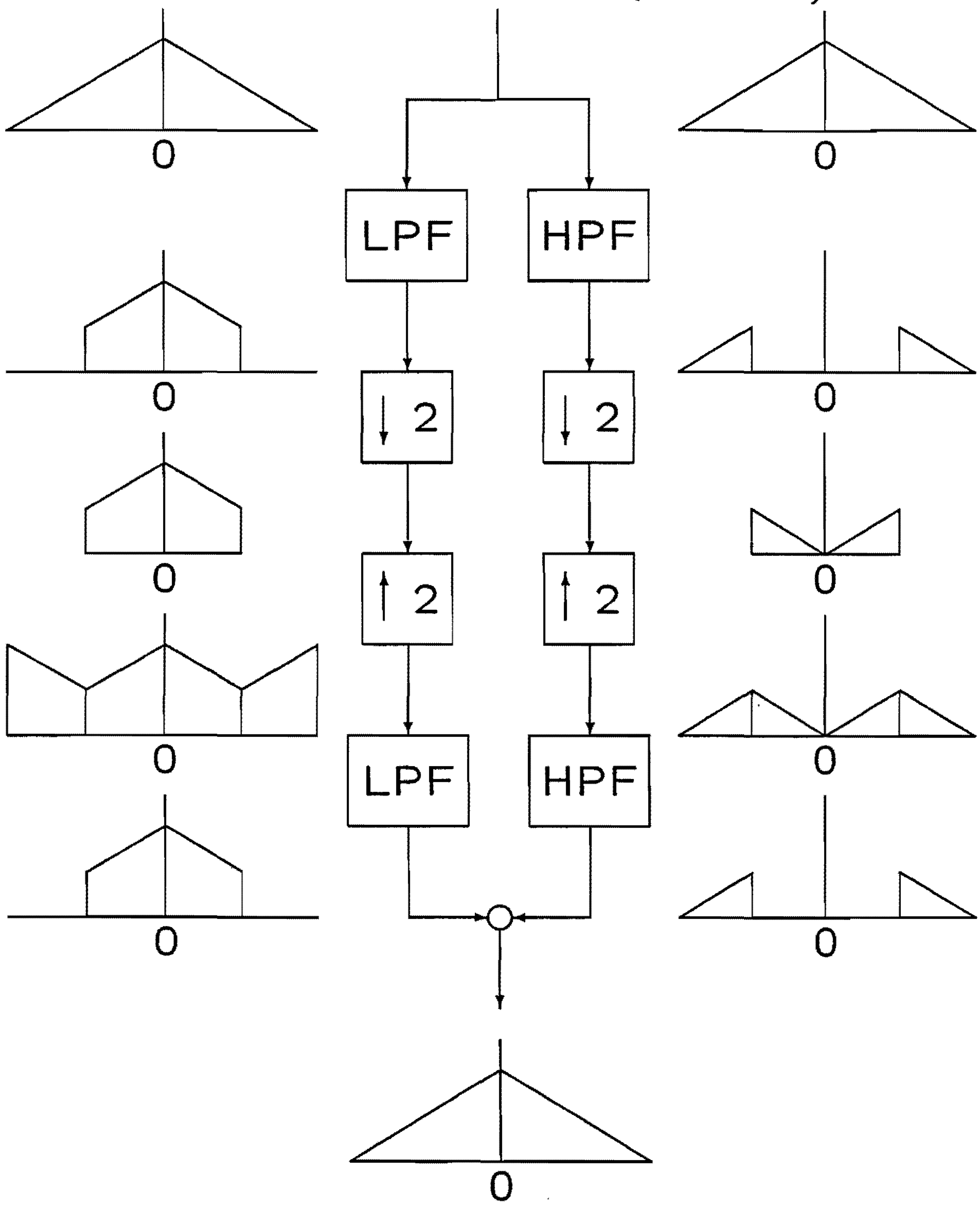



$x[nT_1]$
n
$X(e^{j\omega T_1})$
$1/T_1$
0
π/T_1
ω
$x[nT_1]$
$\downarrow R$
$y[nT_2]$
$Y(e^{j\omega T_2})$
$y[nT_2]$
$1/T_2$
n
π/T_2
ω



$x[nT_1]$
n
$X(e^{j\omega T_1})$
1
ω
π/T_1
$x[nT_1]$
R
$y[nT_2]$
$Y(e^{j\omega T_2})$
$y[nT_2]$
1
n
ω
π/T_1
π/T_2

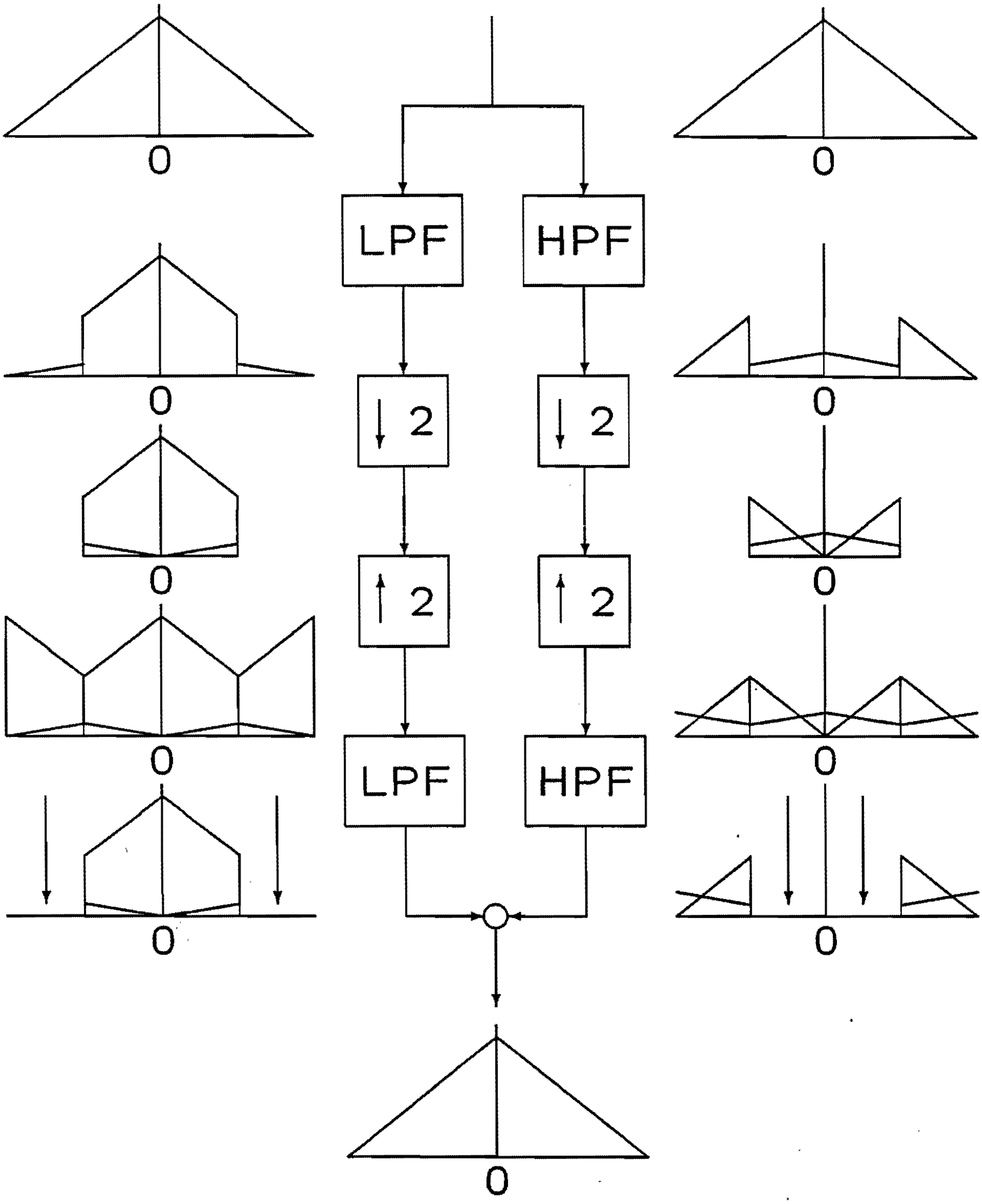
Subband coding (2 bands)



Low Pass Filter: $h_L[n]$ (length 64)

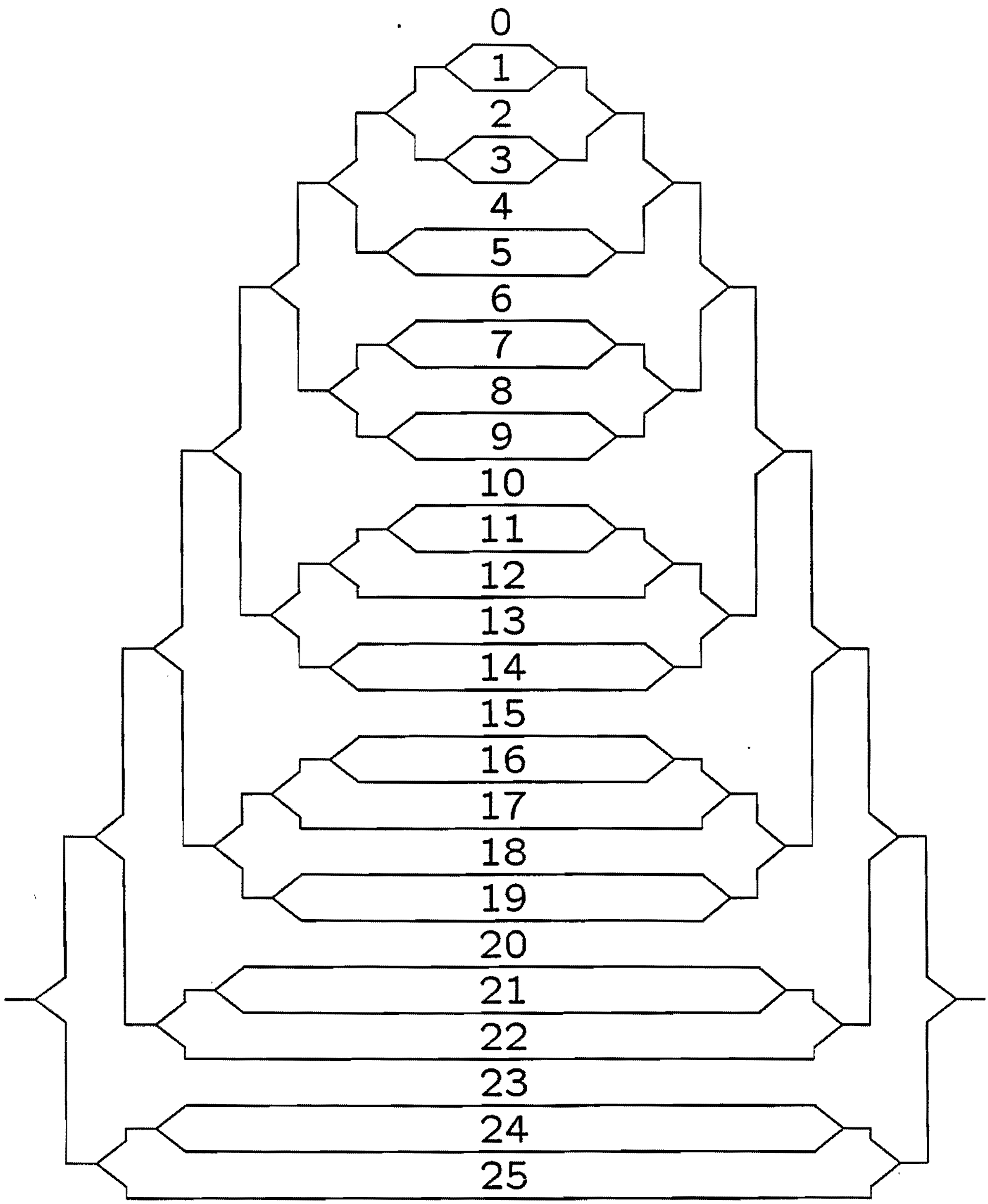
High Pass Filter: $h_H[n] = (-1)^n h_L[n]$

Problem



Aliasing/phase distortion eliminated

Amplitude distortion (almost) eliminated

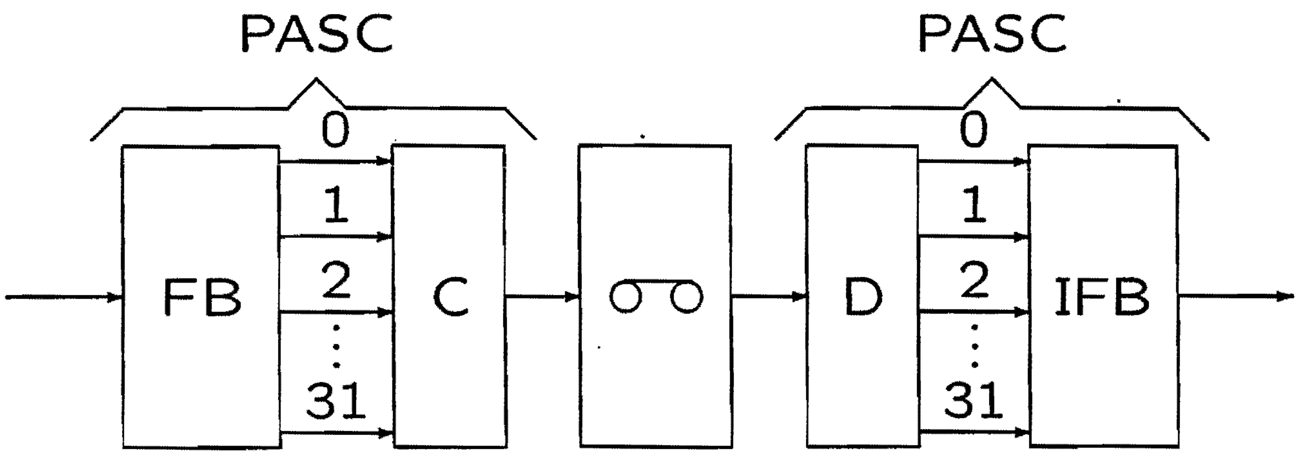


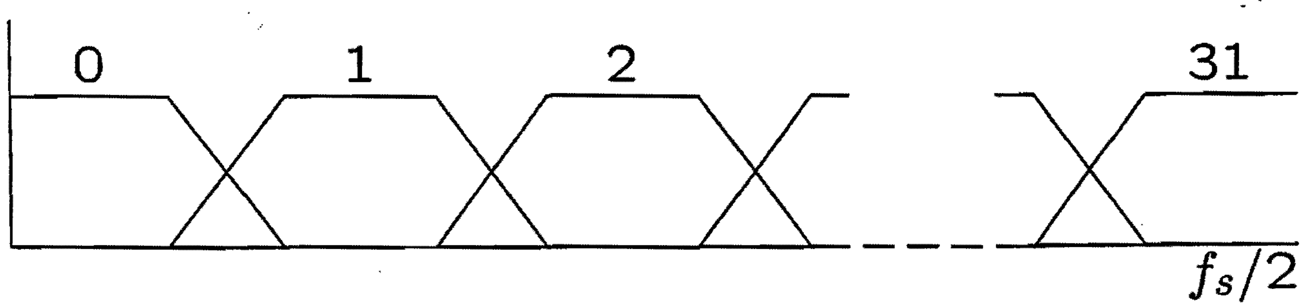
First proposal:

$116.F_s = 5.10^6$ multiplications/second

16320 samples storage (400 kBit)

PASC: Precision Adaptive Subband Coding
(I)FB: (Inverse) Filter Bank
C: Coding
D: Decoding

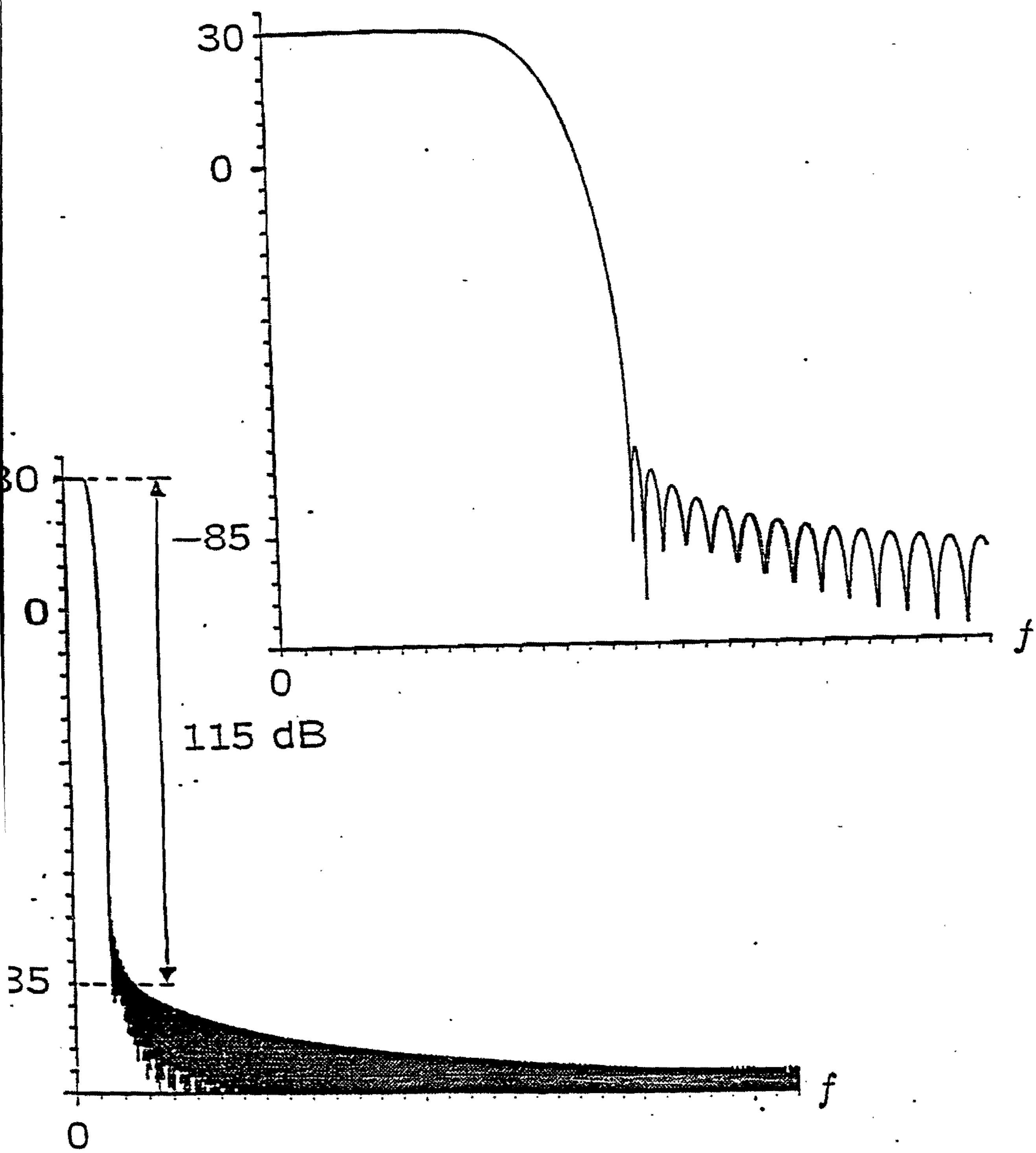




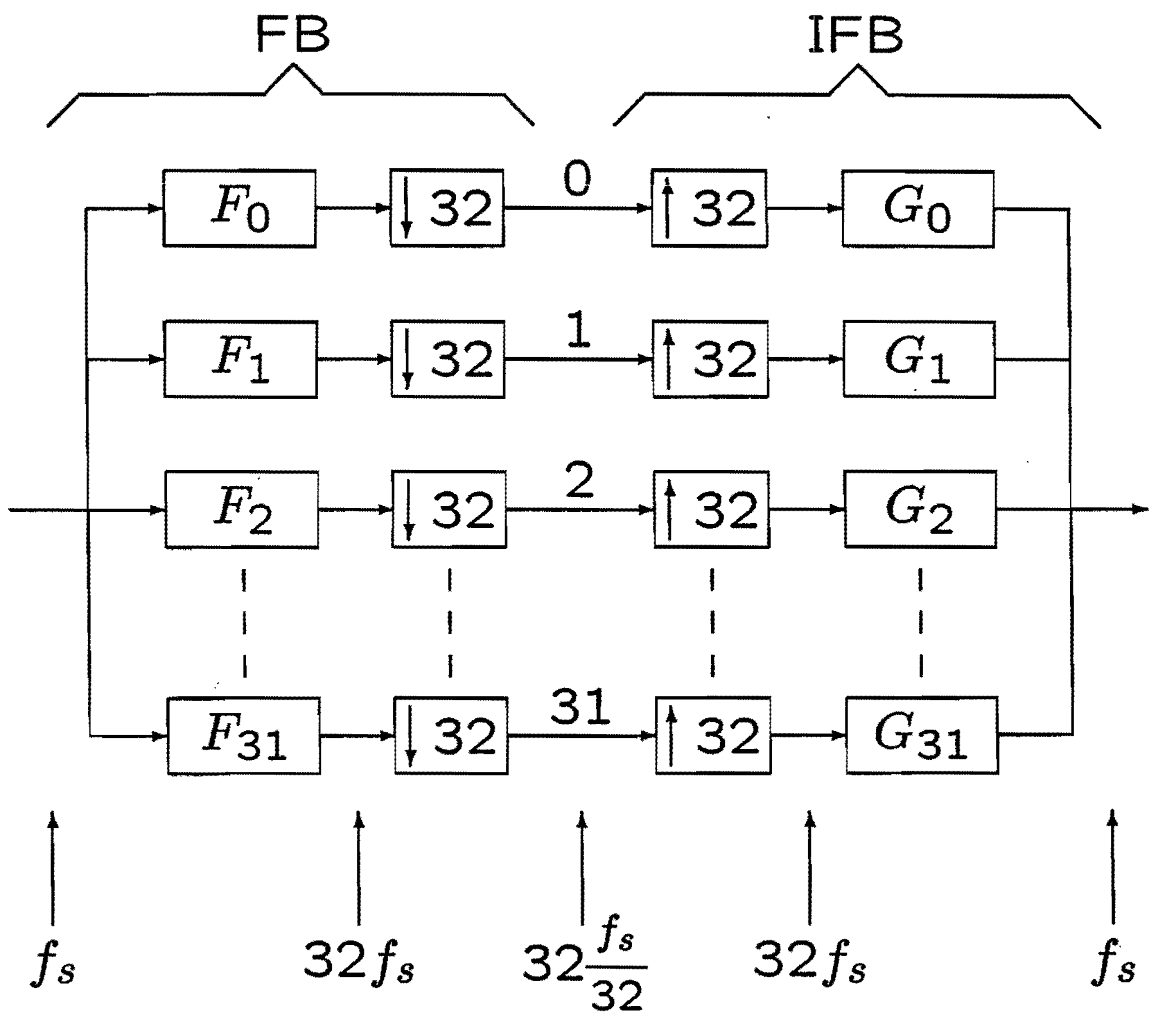
Attenuation: larger than 115 dB

Filterlength: 512

Frequency response of the lowpass filter

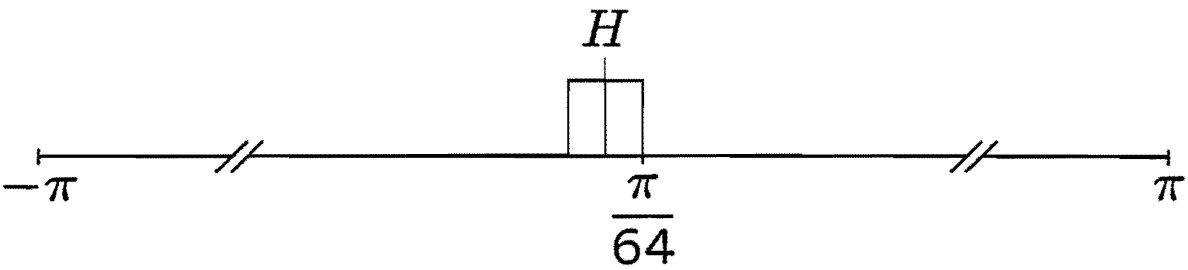


Ignore the coding and the decoding



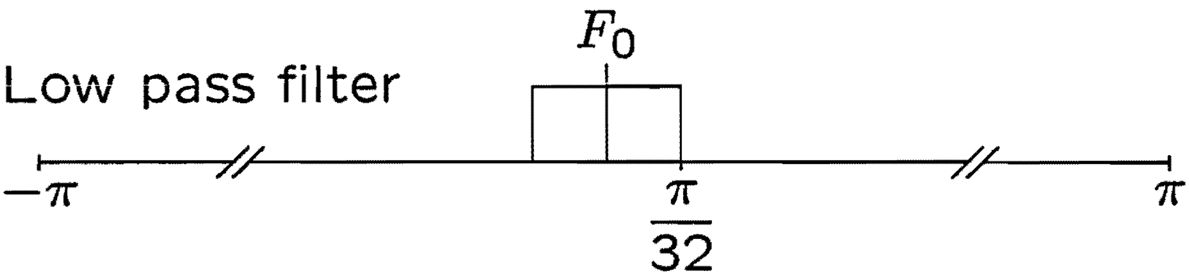
Smaller bandwith: lower sampling frequency

The prototype filter H (half the bandwidth)

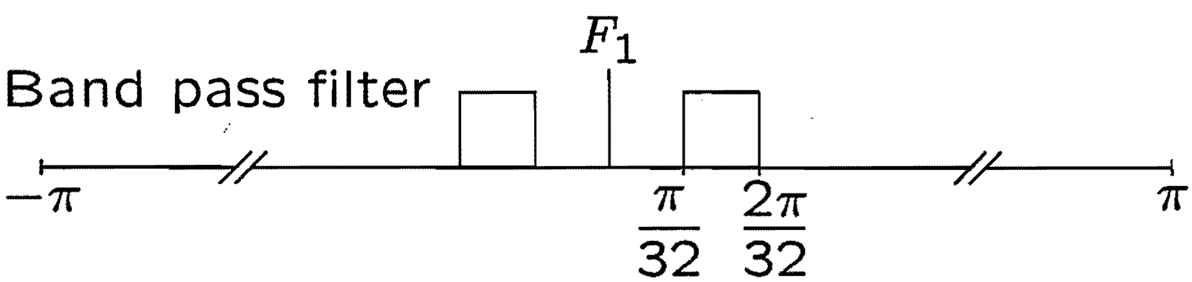


mpulse response $h[n]$

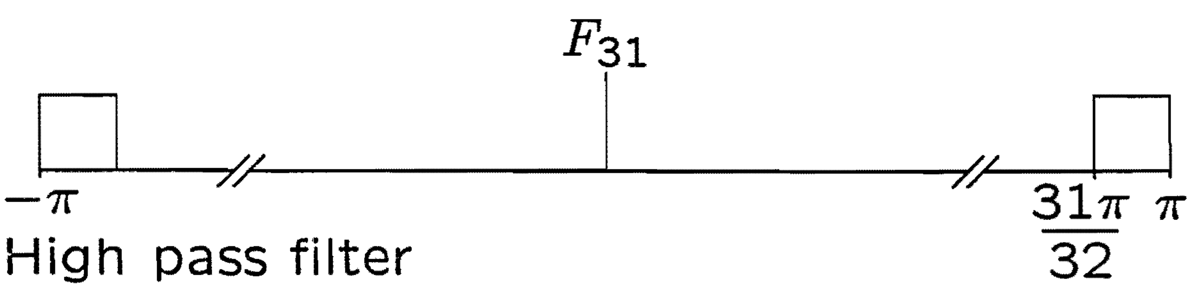
Modulate $h[n]$: $f_0[n] = h[n]\cos(n\pi/64 + \phi_0)$



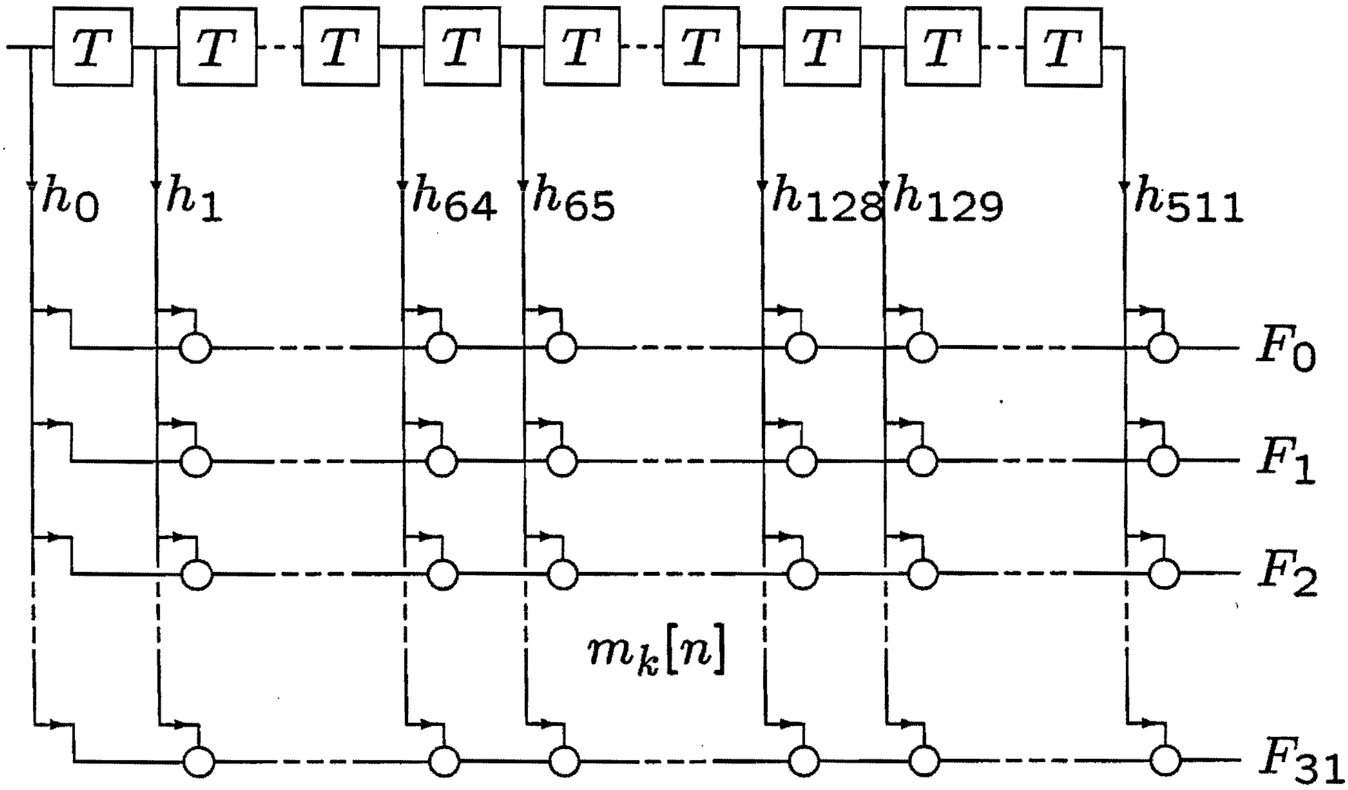
Modulate $h[n]$: $f_1[n] = h[n]\cos(n3\pi/64 + \phi_1)$



Modulate $h[n]$: $f_{31}[n] = h[n]\cos(n63\pi/64 + \phi_{31})$



First generation DCC



k: 0, 1, ..., 31

n: 0, 1, ..., 511

$$m_k[n] = \cos(n(2k+1)\pi/64 + \phi_k)$$

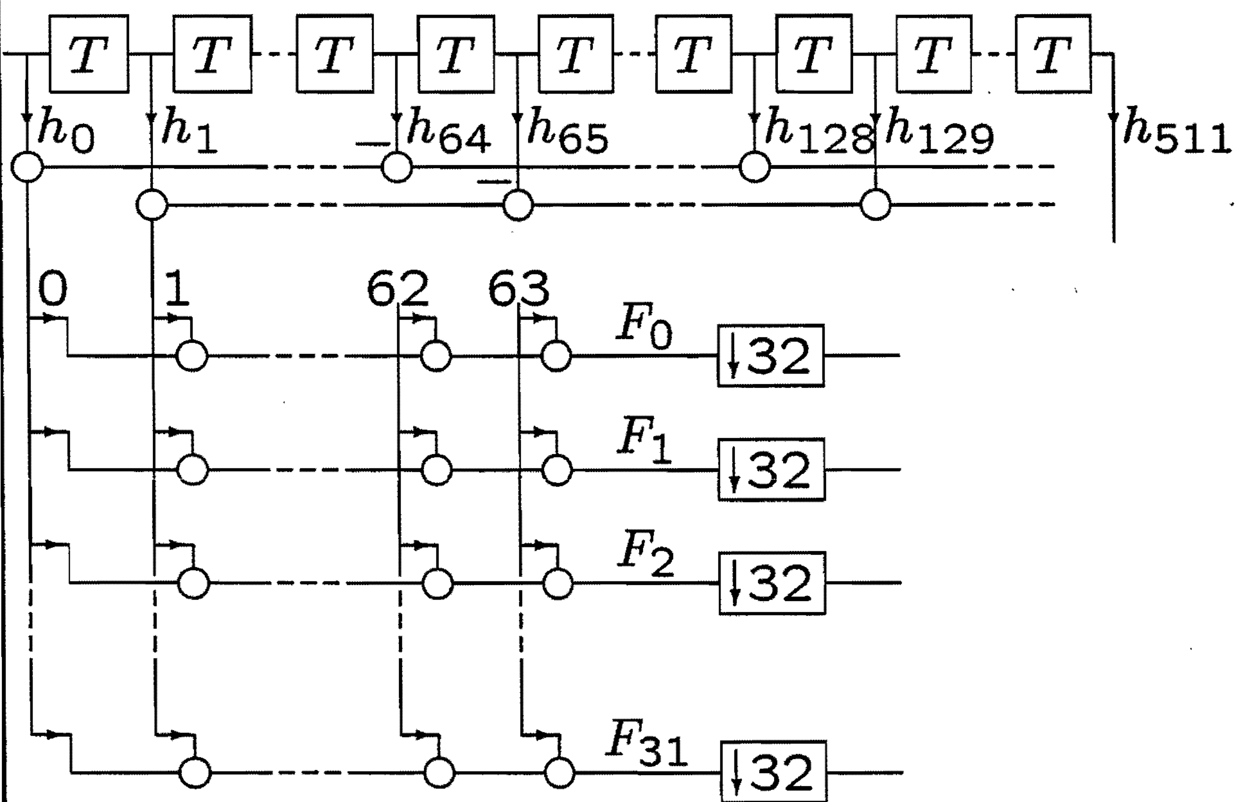
Decreasing the complexity

$$m_k[n] = \cos(n(2k+1)\pi/64 + \phi_k)$$

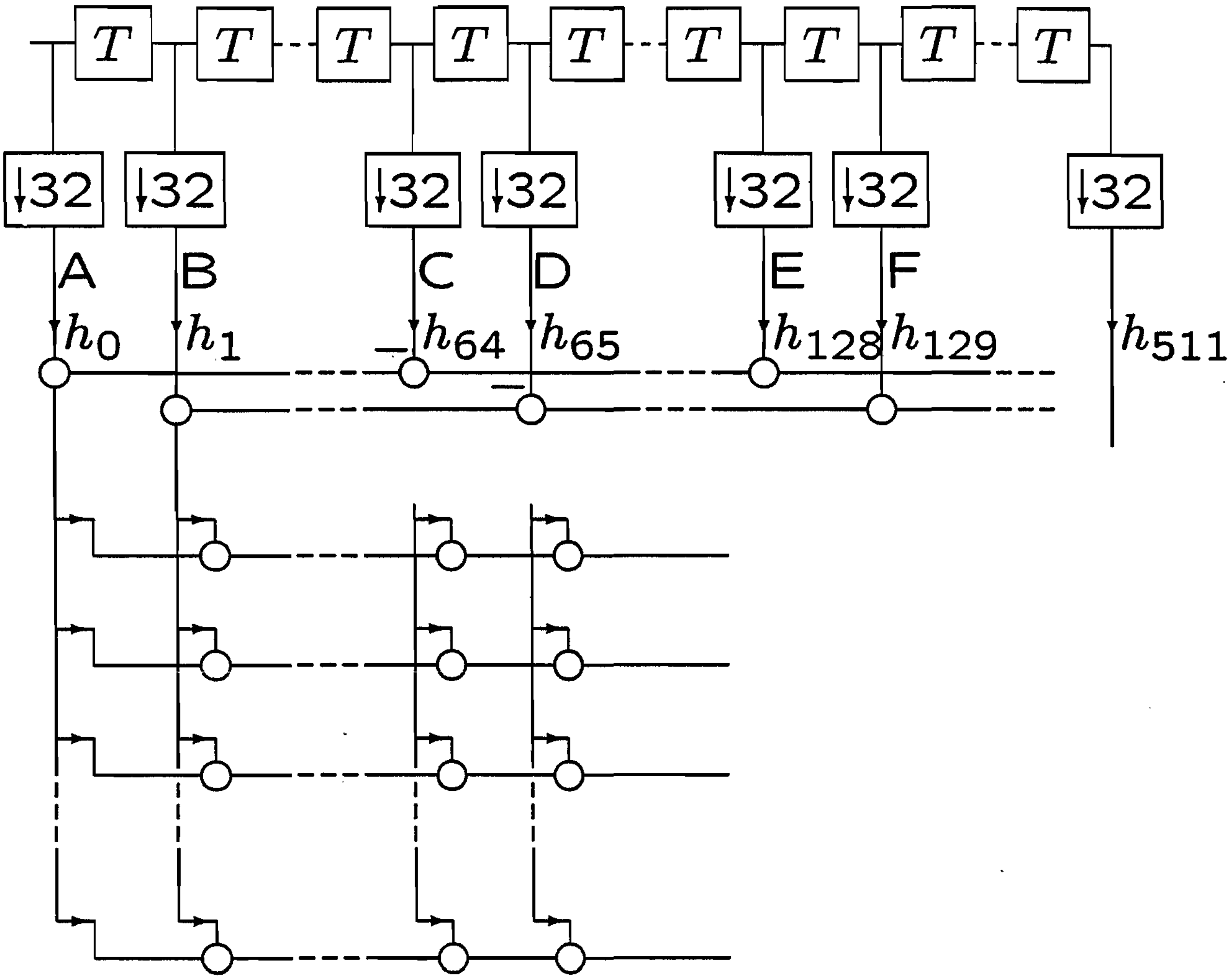
$$m_k[n+64i] = \cos((n+64i)(2k+1)\pi/64 + \phi_k)$$

$$= \cos(n(2k+1)\pi/64 + (2k+1)\pi i + \phi_k)$$

$$= (-1)^i m_k[n]$$

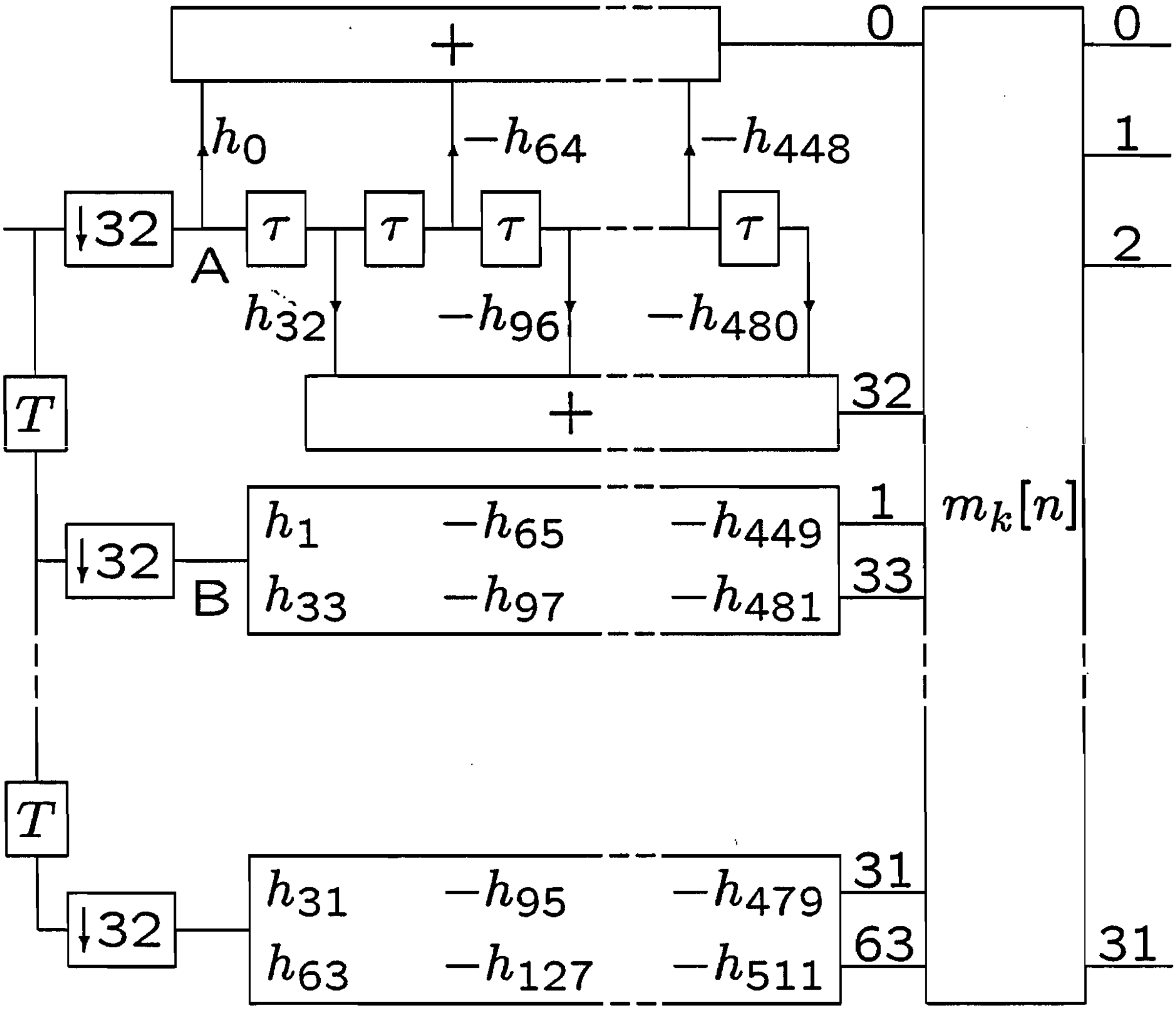


Interchange multiplication and decimation



$$x_A[n] = x[32n]$$
$$x_B[n] = x[32n-1]$$
$$x_C[n] = x[32n-64] = x_A[n-2]$$
$$x_D[n] = x[32n-65] = x_B[n-2]$$
$$x_E[n] = x[32n-128] = x_A[n-4]$$
$$x_F[n] = x[32n-129] = x_B[n-4]$$

Resulting structure



$\tau = 32T$ (one downsampled clock interval)

$m_k[n]$: 64*32 = 2048 multiplications

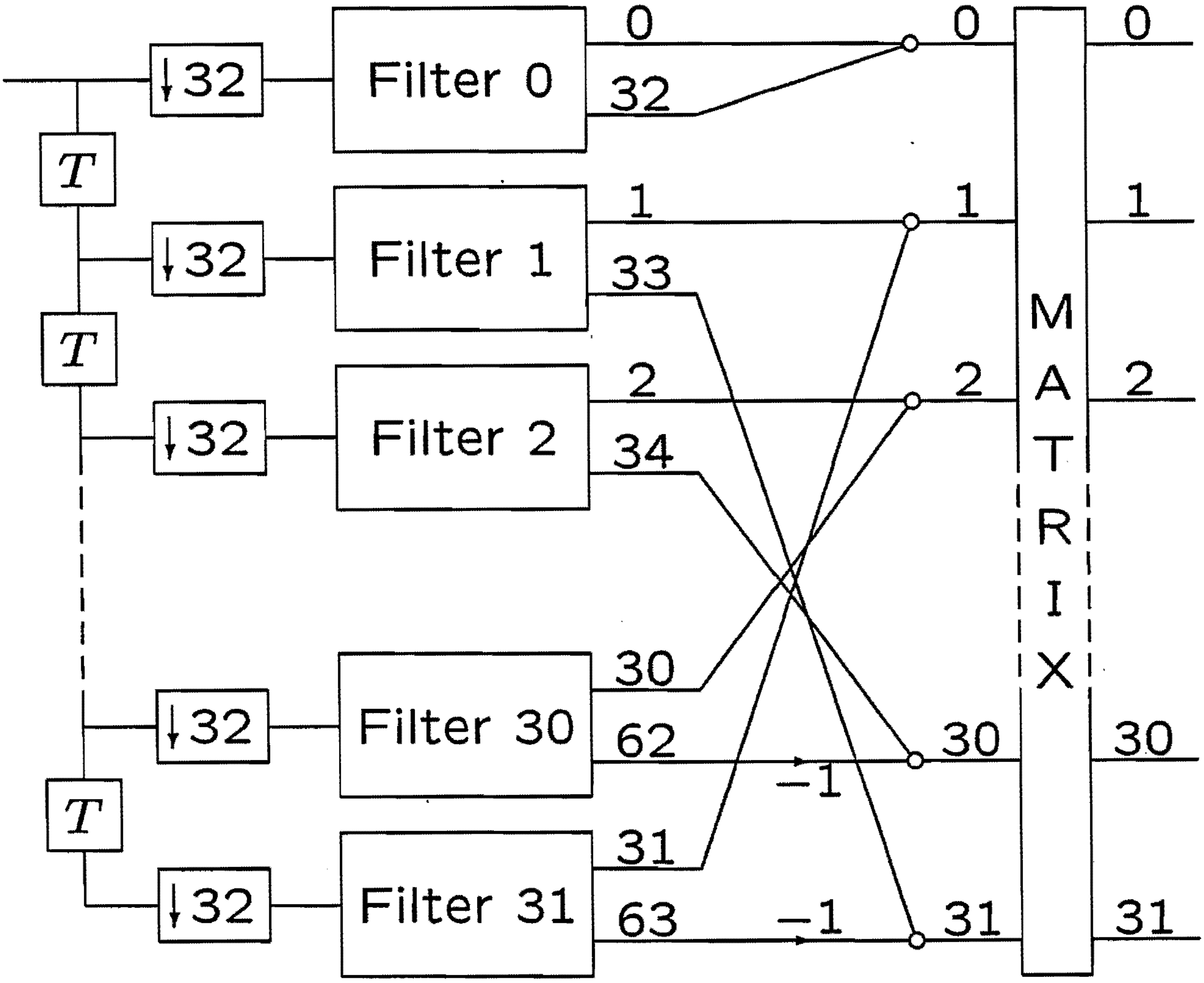
Filters: 512 multiplications

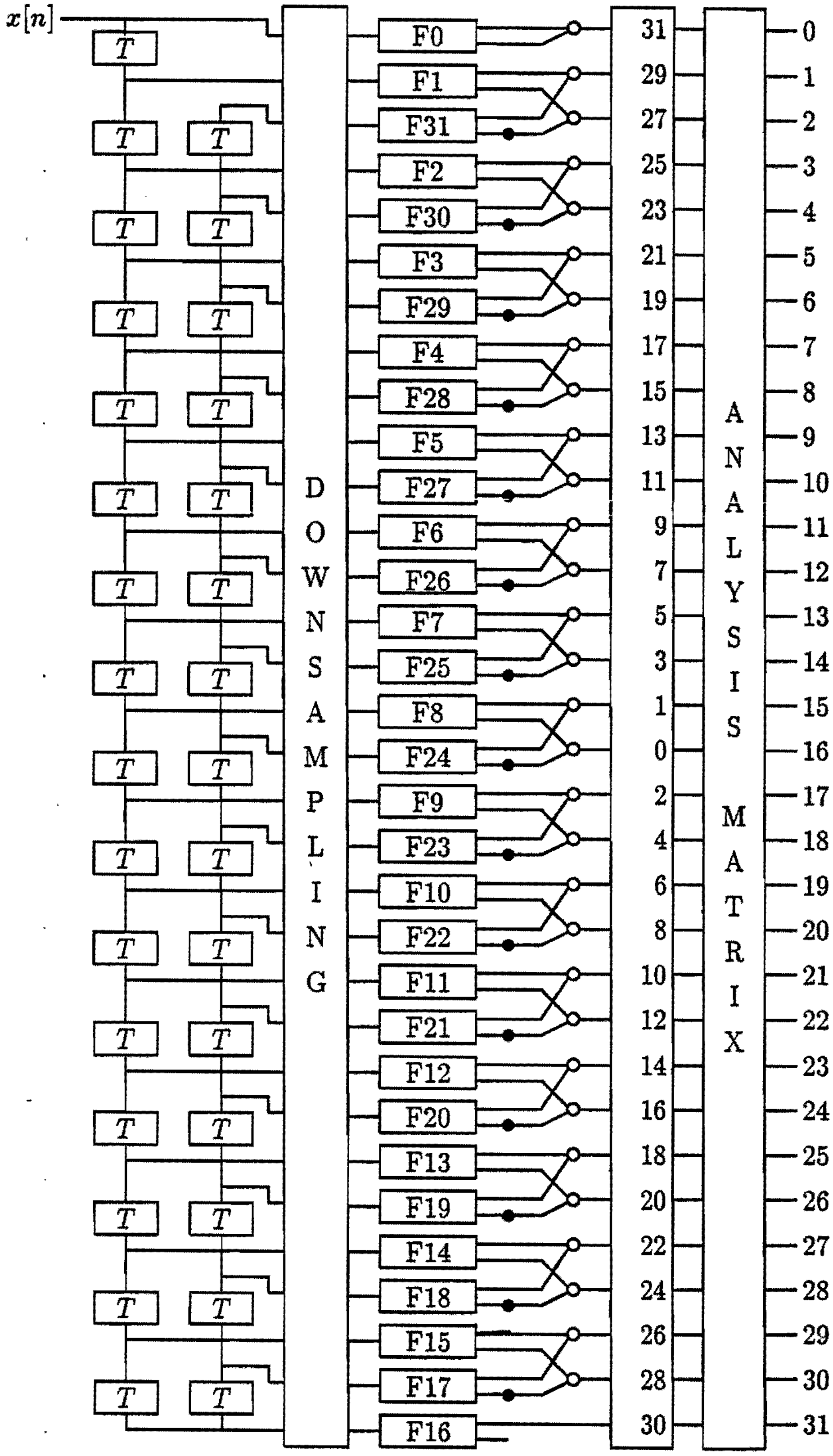
Alternatives

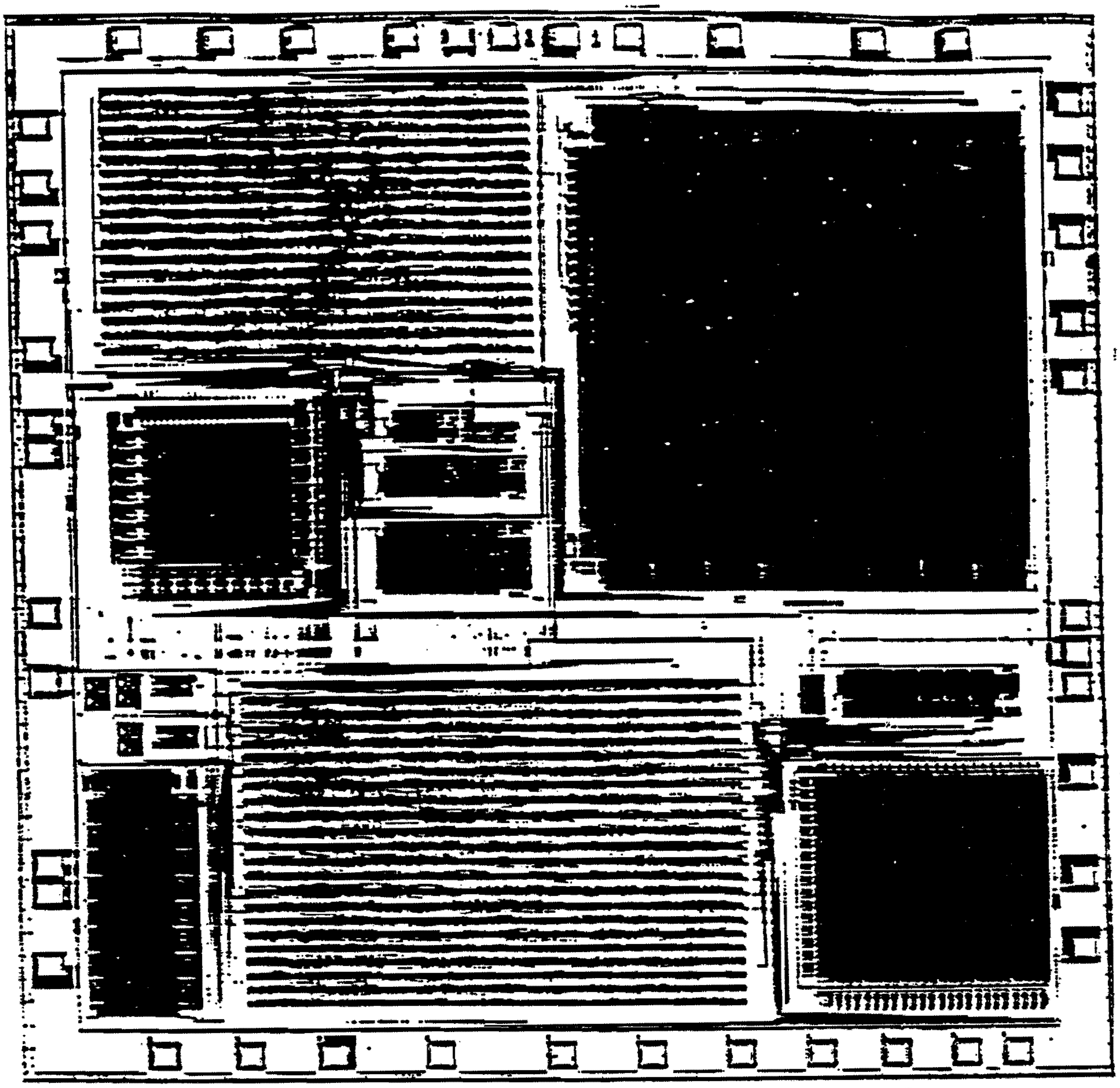
We have: $m_k[n] = \cos(n(2k+1)\pi/64 + (-1)^k\pi/4)$

It can be proved that:

$$
\begin{array}{lll}
m_k[n] & = m_k[32-n] & \text{for } 0 \le n \le 15 \\
m_k[33+n] & = -m_k[63-n] & \text{for } 0 \le n \le 14 \\
m_k[48] & = 0 &
\end{array}
$$




x[n]
T
DOWNSAMPLING
F0
F1
F31
F2
F30
F3
F29
F4
F28
F5
F27
F6
F26
F7
F25
F8
F24
F9
F23
F10
F22
F11
F21
F12
F20
F13
F19
F14
F18
F15
F17
F16
31 29 27 25 23 21 19 17 15 13 11 9 7 5 3 1 0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
ANALYSIS MATRIX
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

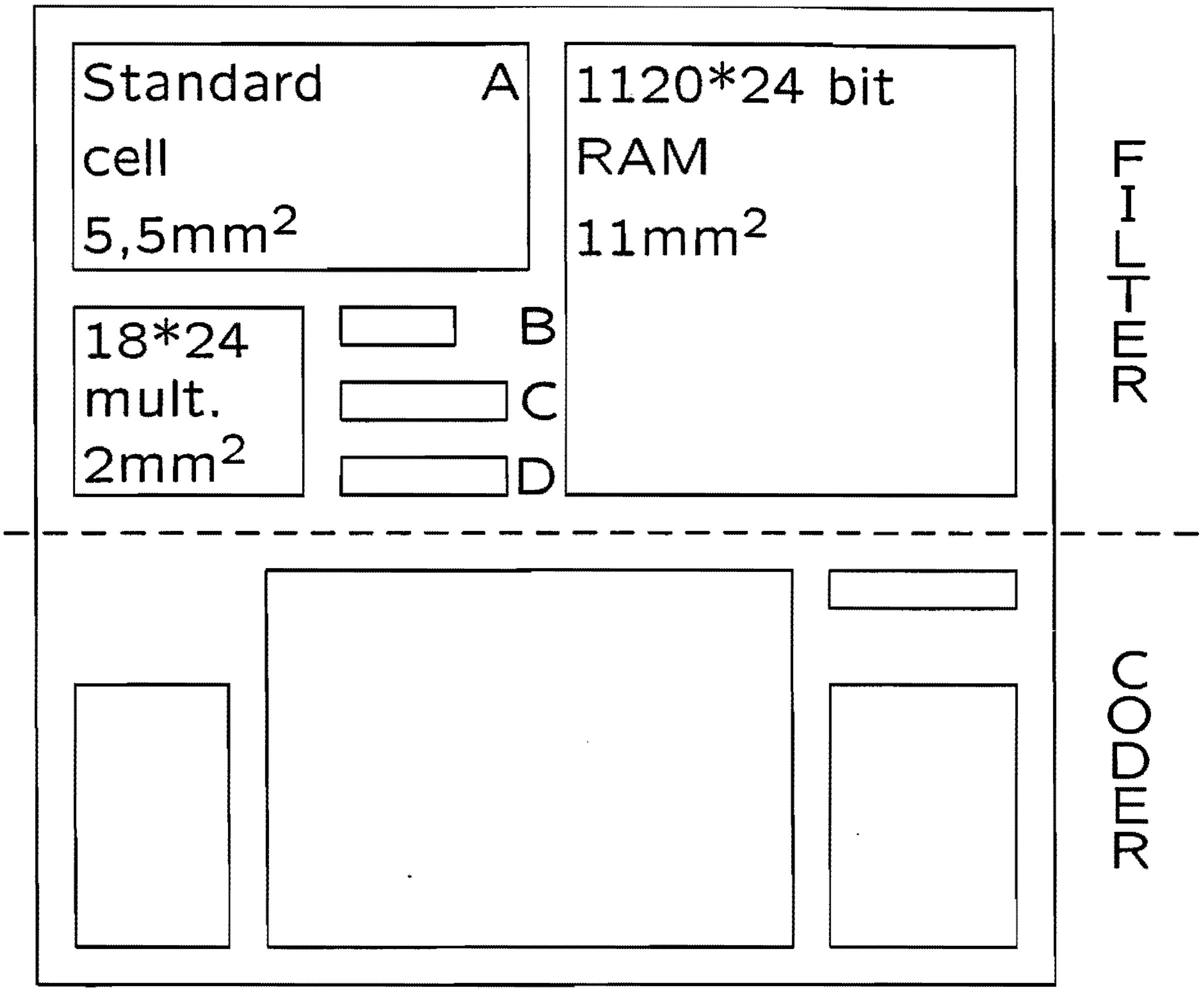


PASC–generation 3 (Southampton)

Stereo recording or reproduction

Filter bank	Generation 1/2	Generation 3
Area	25 mm^2	25 mm^2
Dissipation	230 mW	117 mW

Stereo analysis or synthesis



A: 30 bit accumulator

input/output, ...

B: 128x7 ROM (adress)

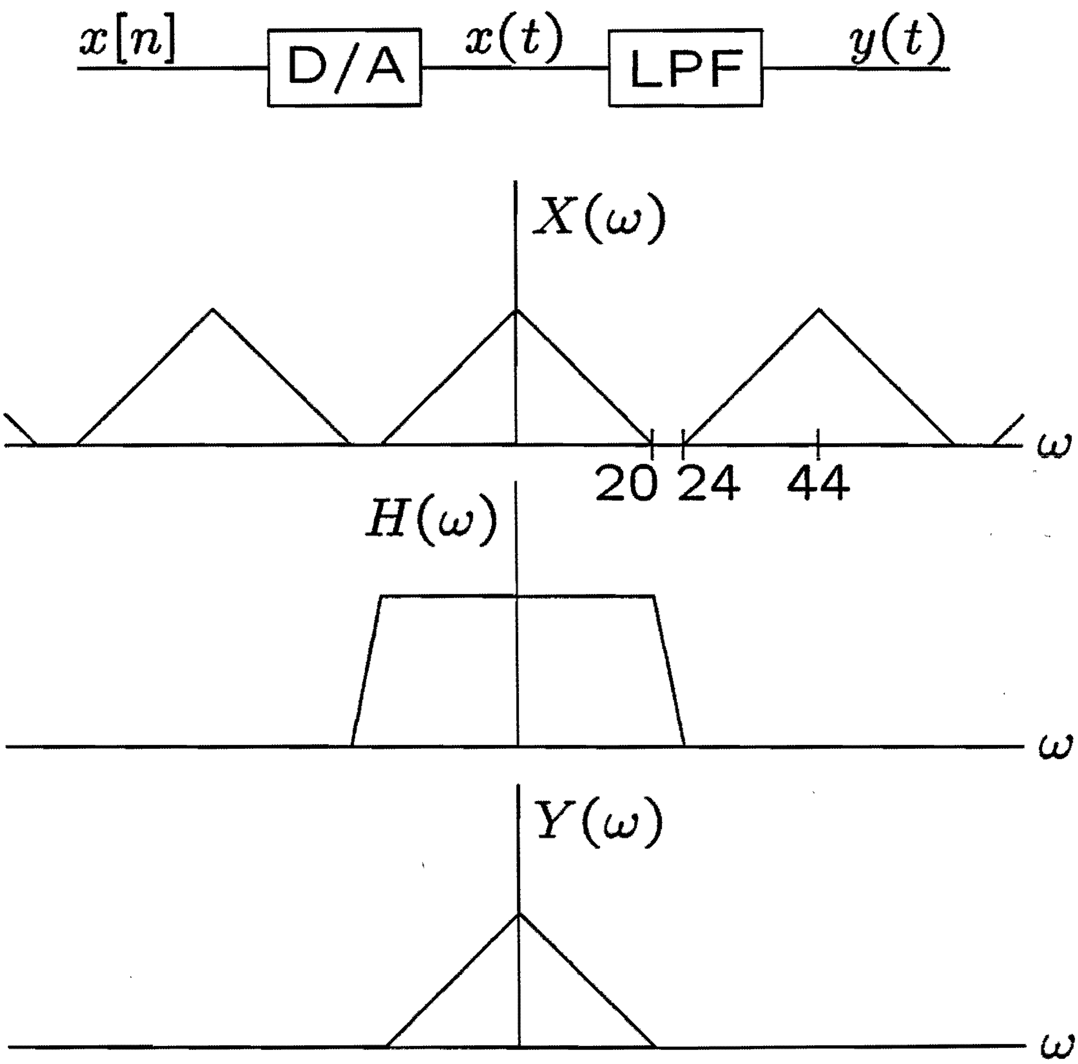
C: 210x18 ROM (micro controller)

D: 704x18 ROM (coefficients)

Effective ROM area: $1mm^2$

OVERSAMPLING FOR DIGITAL AUDIO

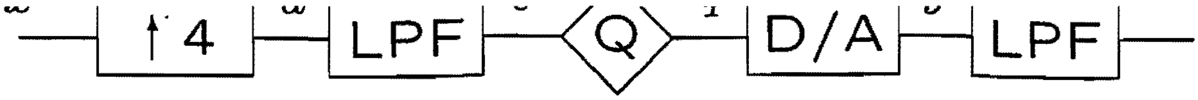
Conventional D/A conversion

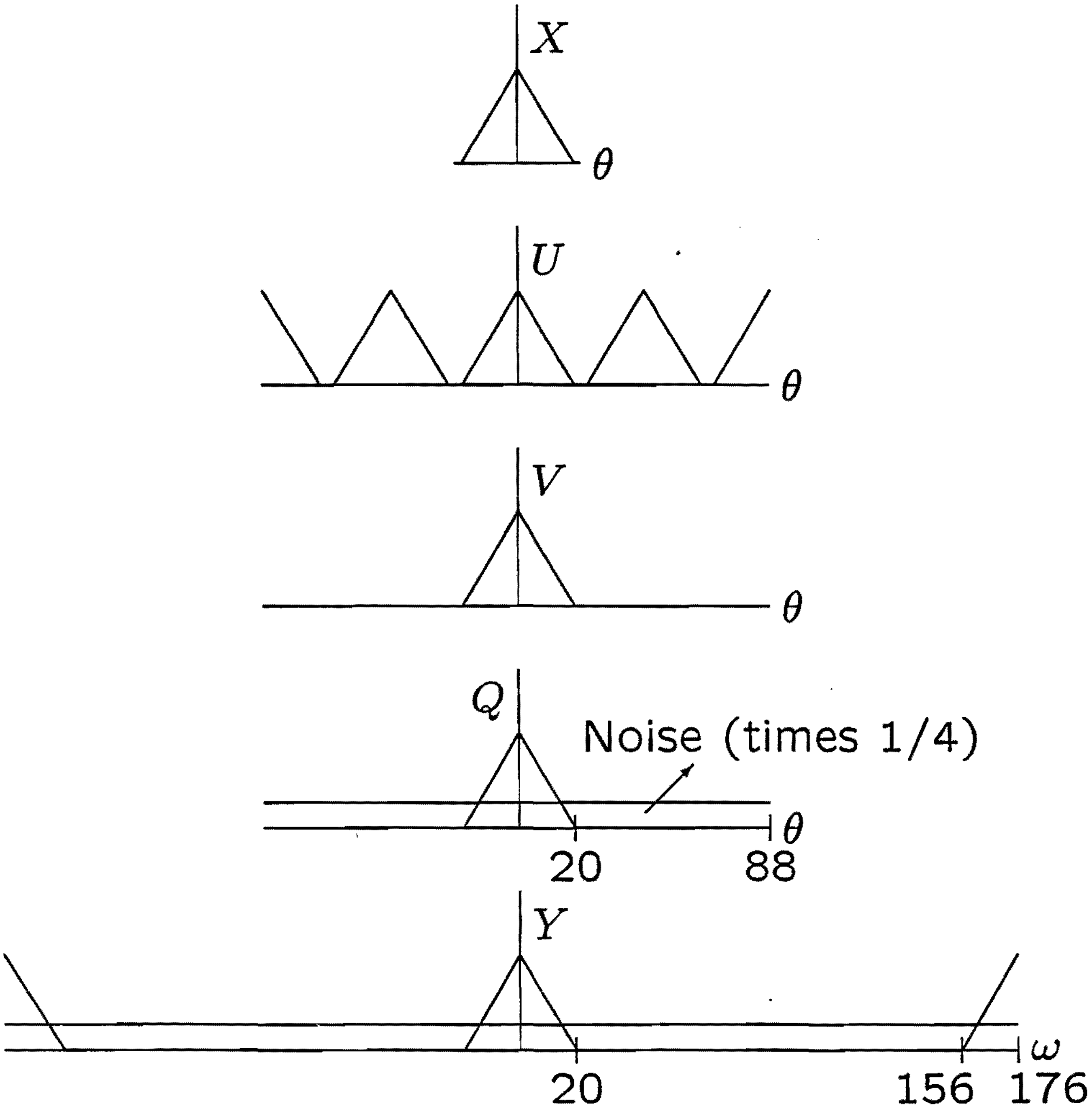


Analog post filter:

Pass band	0 kHz to 20 kHz
Transition band	20 kHz to 24 kHz
Stop band	from 24 kHz

Oversampling: digital versus analog processing



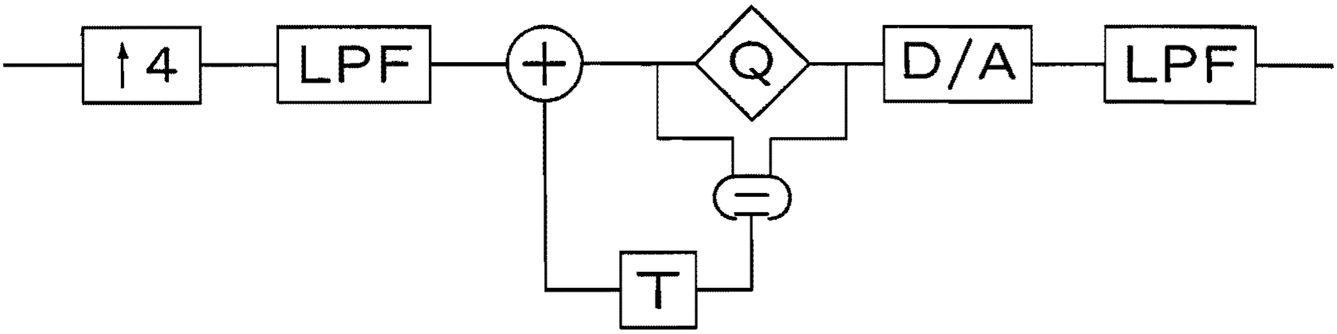


Analog post filter:

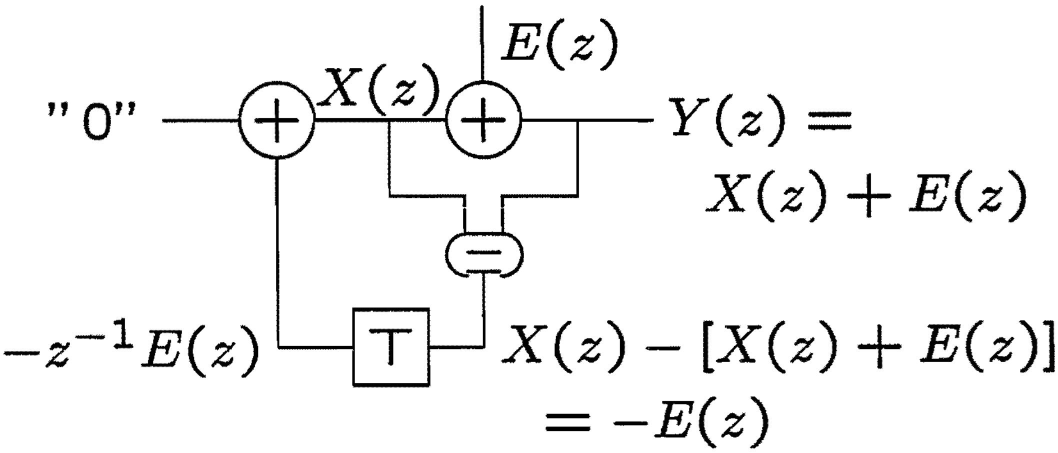
Pass band	0 kHz to 20 kHz
Transition band	20 kHz to 156 kHz
Stop band	from 156 kHz

Noise shaping:

the feedback of the quantization error



Model of the noise shaper



From this model we have:

$$X(z) = -z^{-1}E(z)$$

$$Y(z) = X(z) + E(z) = (1 - z^{-1})E(z)$$

$$Y(z)/E(z) = H(z) = 1 - z^{-1}$$

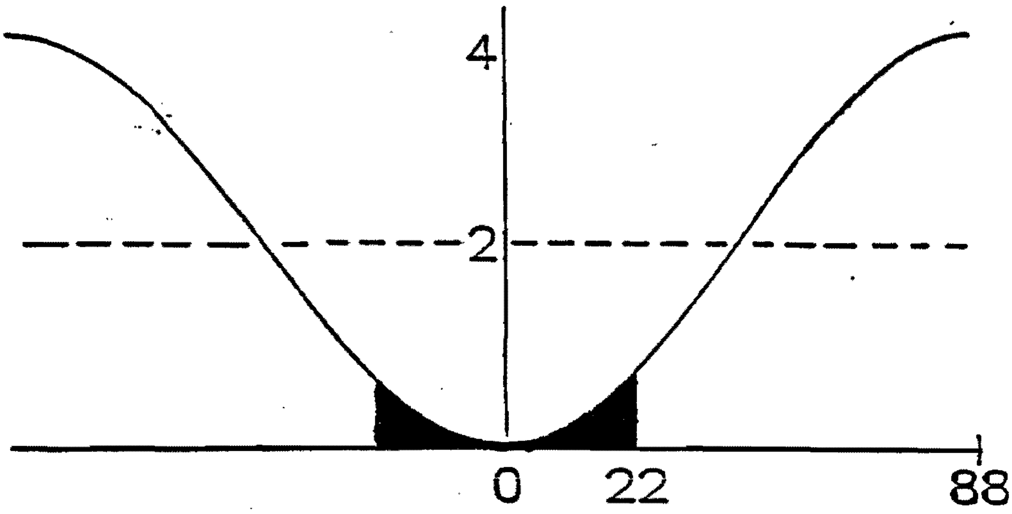
$$A(e^{j\theta}) = |1 - e^{-j\theta}| = 2.|\sin(\theta/2)|$$

The noise power of the error source is $q^2/12$

The noise spectrum is shaped by $2.|\sin(\theta/2)|$

The noise spectrum is shaped by $2.|\sin(\theta/2)|$

or power shaped by $4.\sin^2(\theta/2)$



The output noise power is

$$2q^2/12$$

The output noise power in the audio band is

$$0.05q^2/12$$

This means a gain of more than 2 bits in

signal to quantization distortion

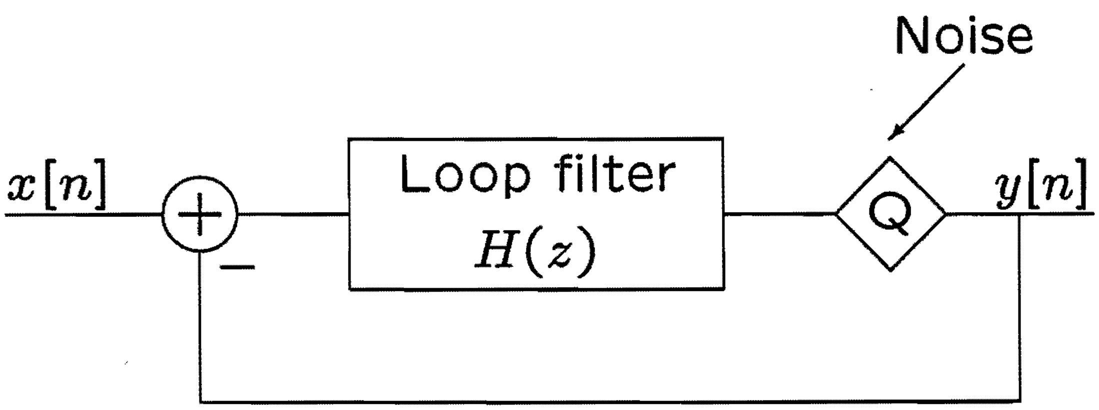
For instance: 14 bit signal/16 bit quality

Gain in S/N ratio due to:

1. Oversampling (noise outside the band)
2. Digital filtering (noise colouring)

Bit stream conversion uses:

1. Digital interpolation to $256.f_s$ ($\approx$ 11 Mhz)
2. Quantization to 1 bit
3. Noise shaping

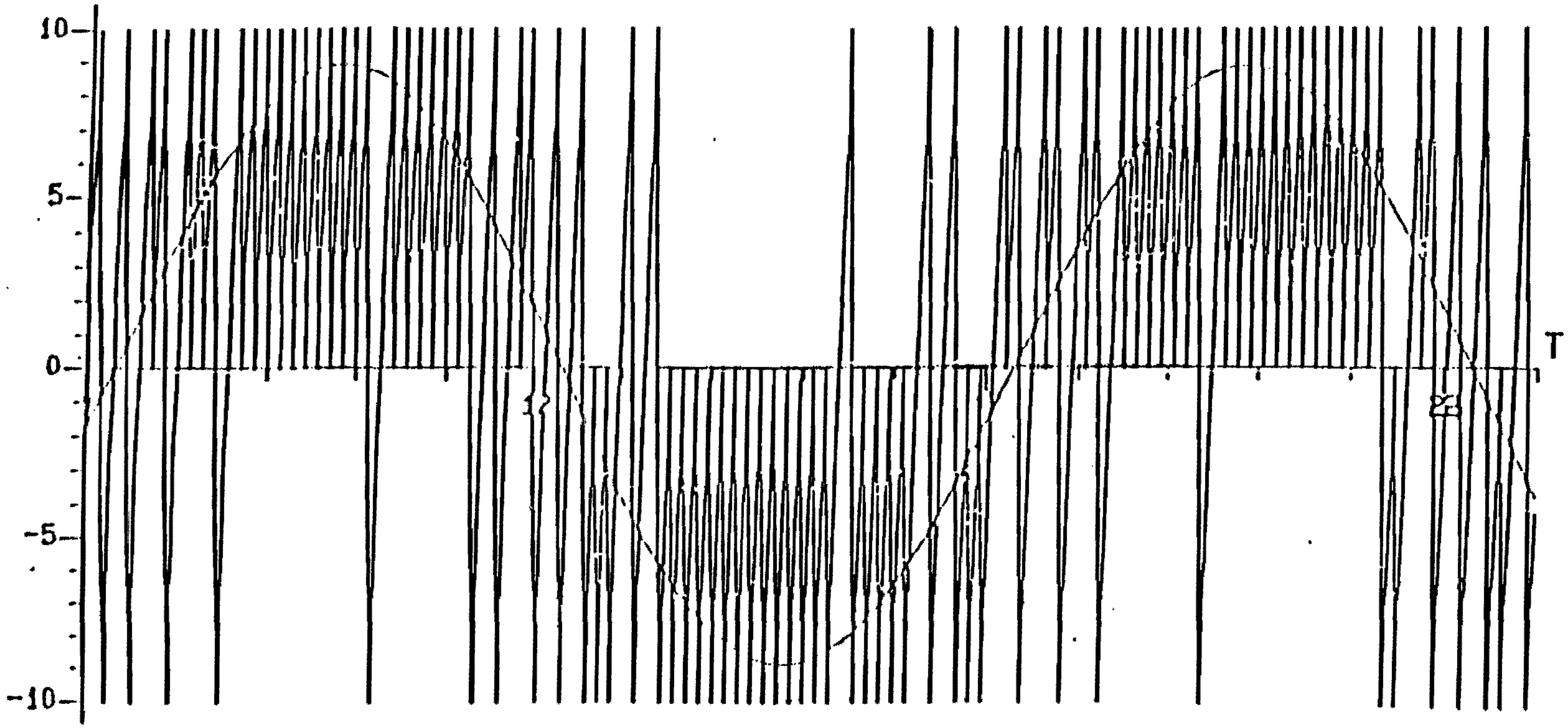


From $x[n]$ to $y[n]$: $H_x = H/(1+H)$

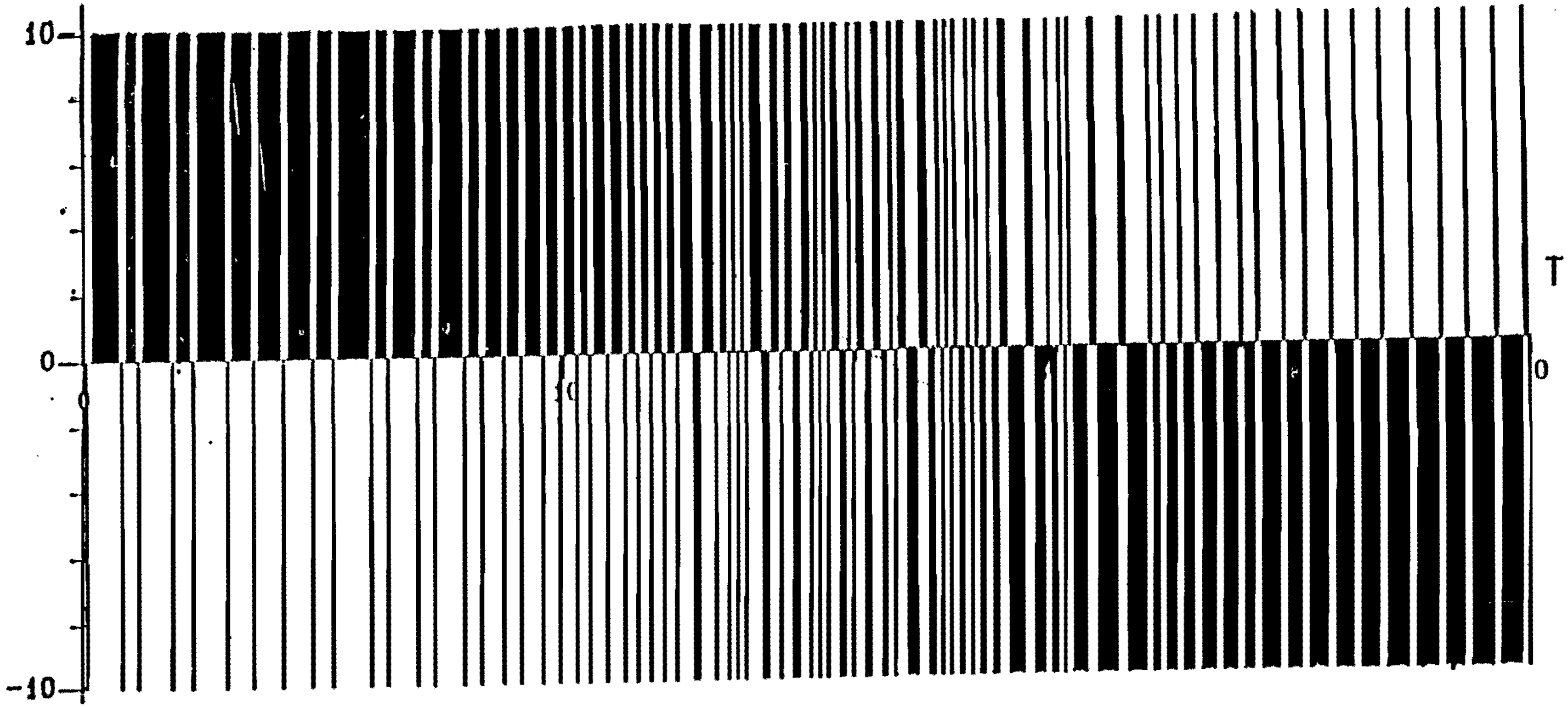
From noise to $y[n]$: $H_n = 1/(1+H)$

In the audio band we choose $H >> 1$; therefore:

$H_x \approx 1$ and $H_n \approx 0$



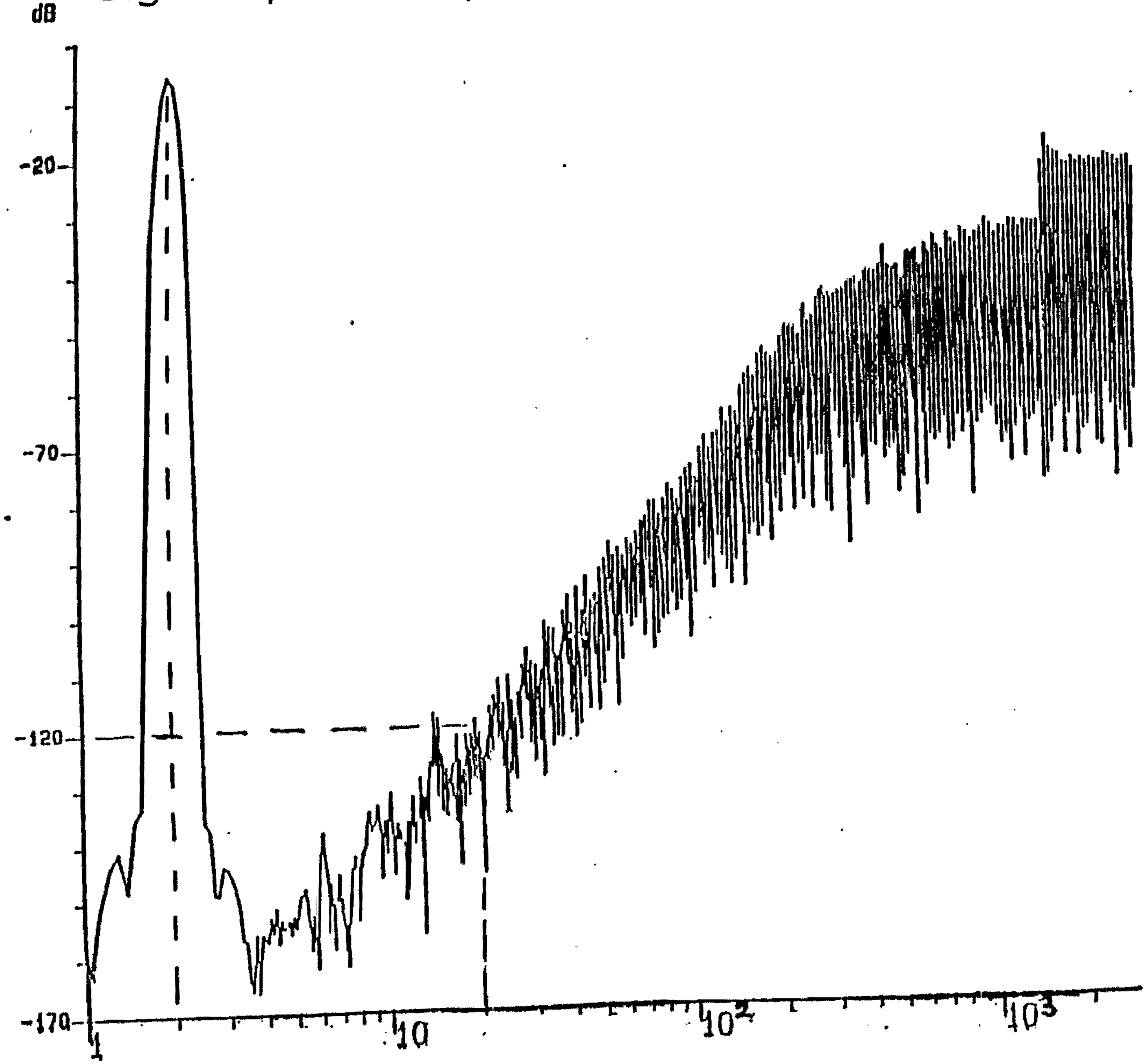
Simulation of pulse train from the quantizer
as a result of a sine wave input signal
(Reduced sample rate for illustration)



Practical example of pulse train (bitstream)
due to a sine wave input (simulated)

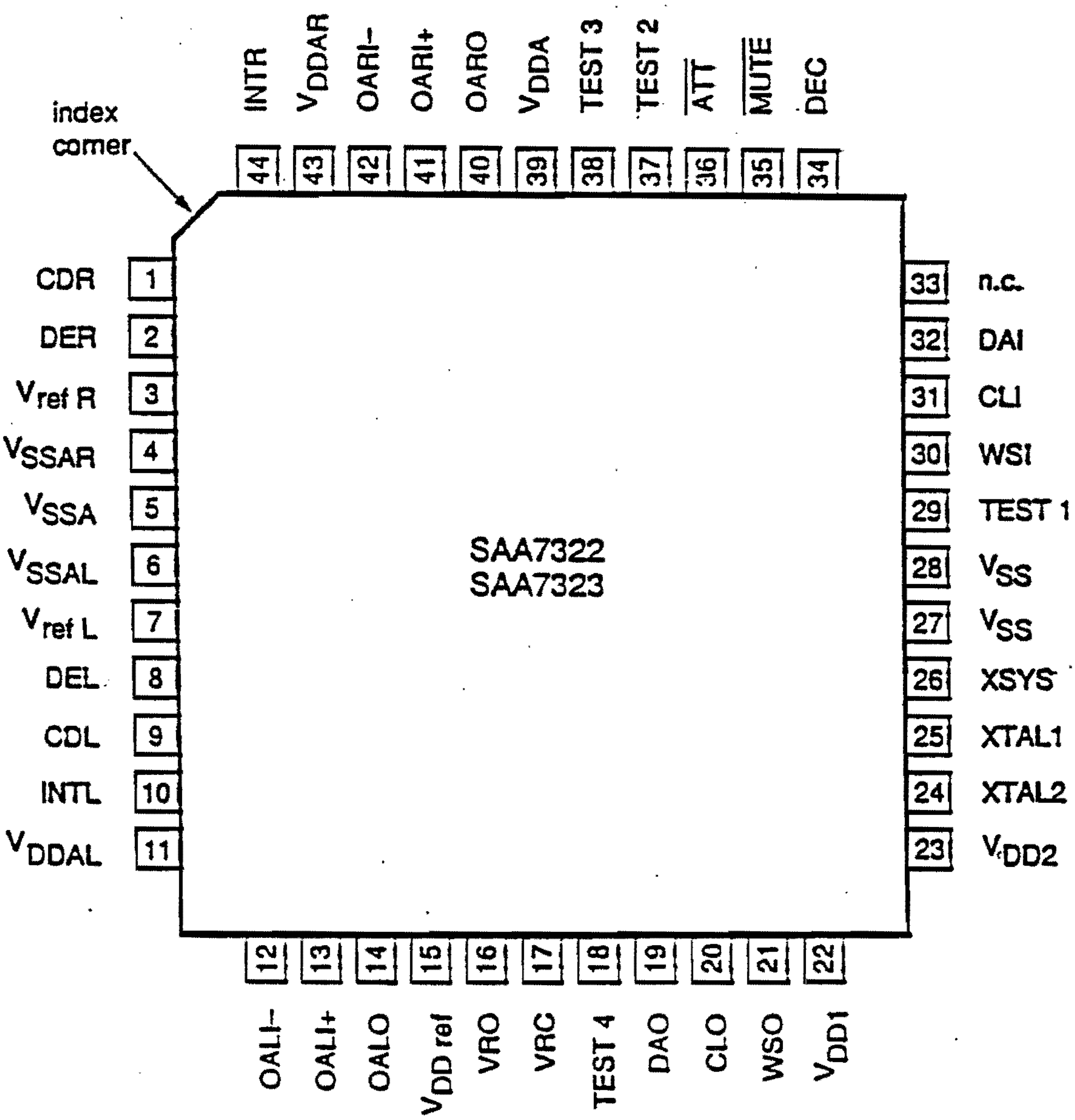
40

Signal + noise spectrum



49

BITSTREAM CONVERSION



SAA7322/7323

I²S SERIAL INPUT

↓ 16 bit; 44.1 kHz

4* — 4* OVERSAMPLING
(FIR filter)

↓ 16 bit; 176.4 kHz
attenuation/mute

128* — 32* OVERSAMPLING
(linear interpolator)
DITHER SIGNAL ADDED

256* — 2* OVERSAMPLING
(sample and hold)

↓ 17 bit; 11.2 MHz

1–BIT PDM SIGNAL
(2nd order noise shaper)

↓ 1 bit; 11.2 MHz

BITSTREAM CONVERSION DAC
(switched capacitor network)

↓ de–emphasis

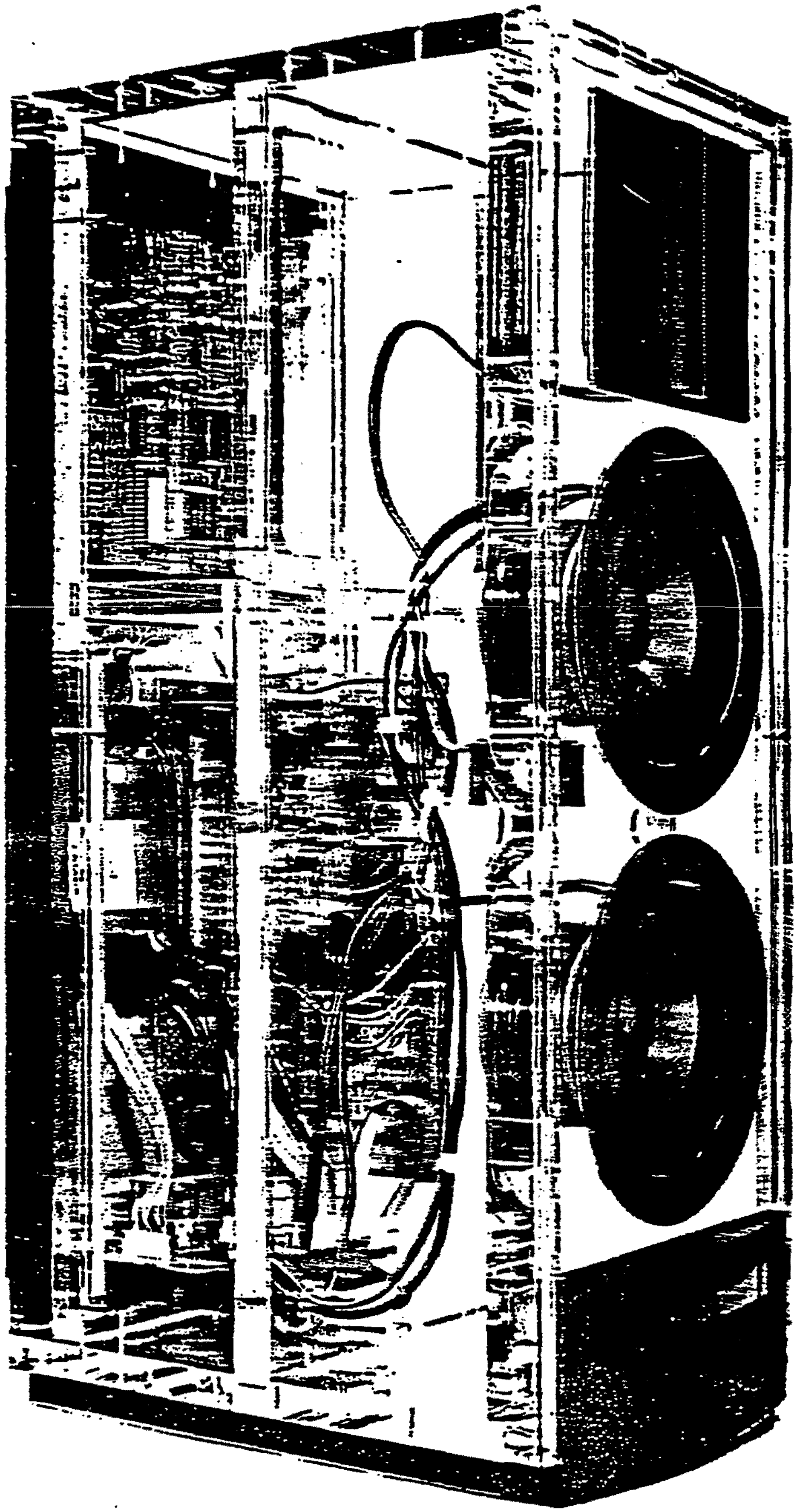
3rd ORDER LOW–PASS FILTER

↓ ANALOG OUTPUT

The Digital Speaker System (DSS930):

* the digital signal is supplied to the speaker system
* digital correction of amplitude, phase and cross–over
* D/A conversion inside the box
* separate woofer and tweeter amplifiers
* digital DSS bus (IEC958 compatible) to a Digital System Controller (DSC950)
* 12 speakers can be grouped into 3 systems (room A, B and C) and they can be controlled separately
* a two way connection between the speaker and and the controller are performed using frequency multiplexing
* audio interface: I^2S

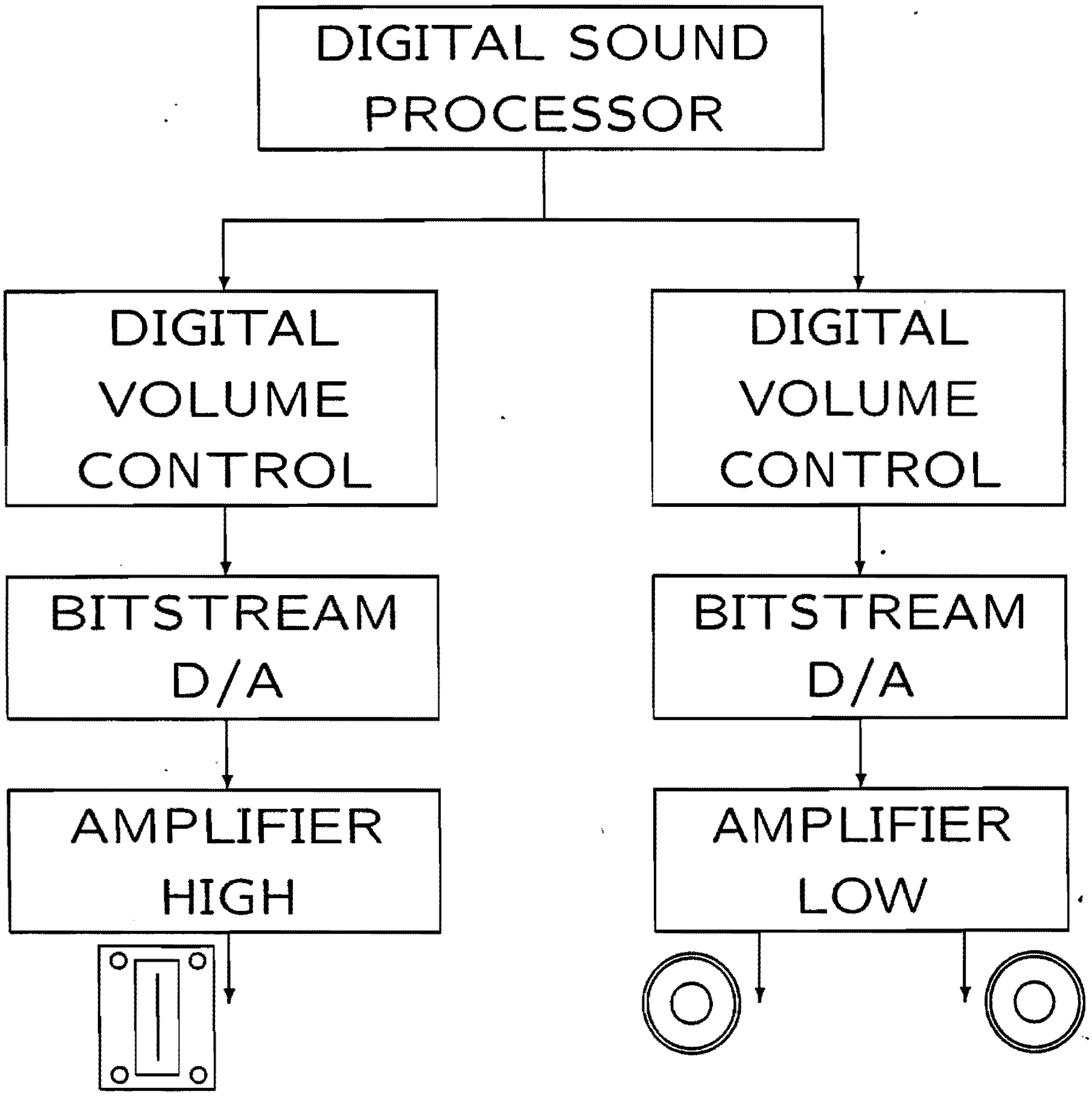
52



hilips Research

PHILIPS

DSS 930 Blockdiagram

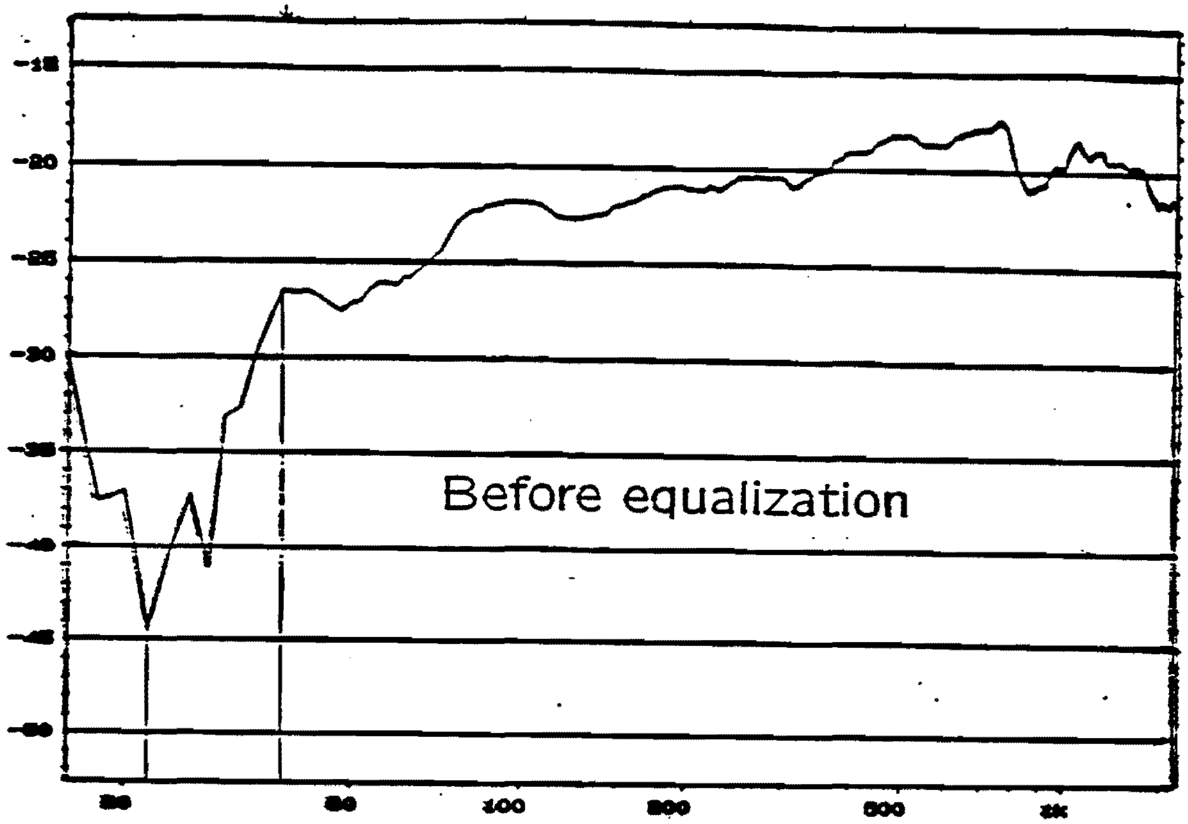


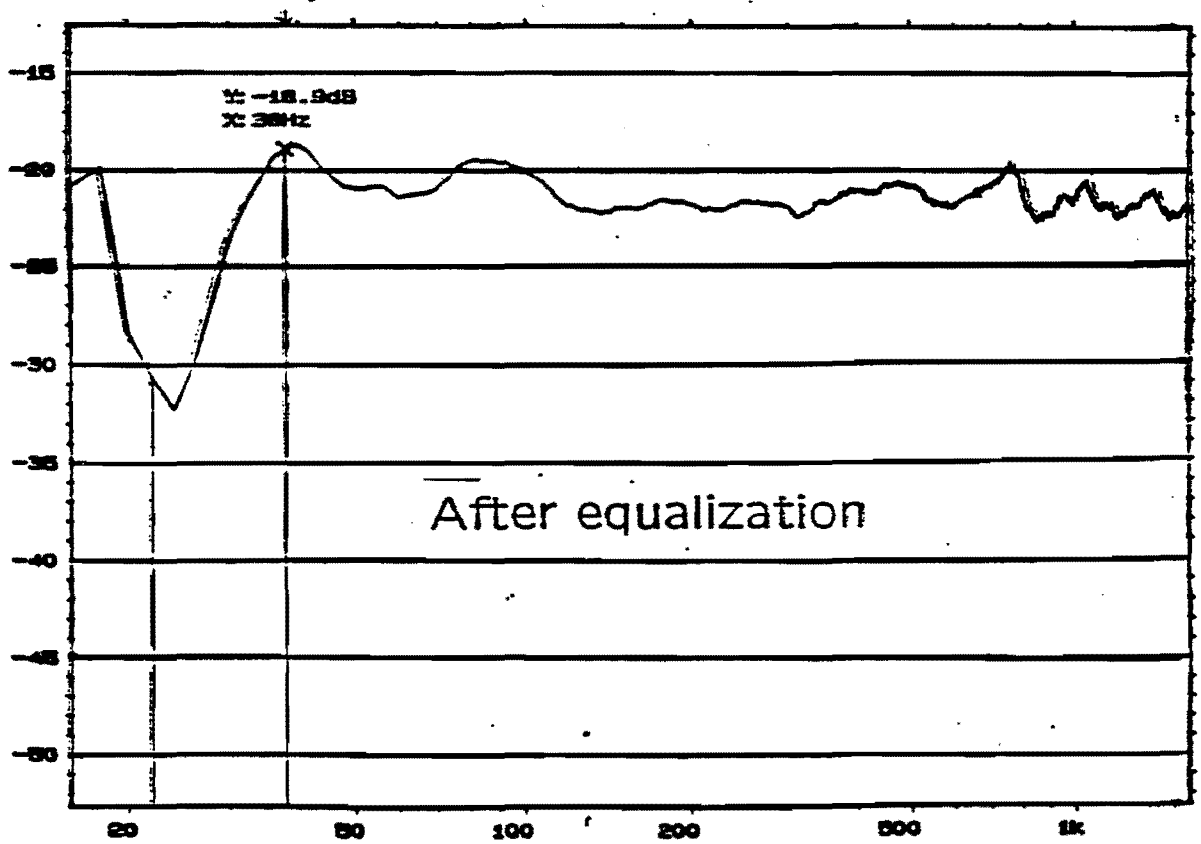
POWER SUPPLY
MICRO PROCESSOR CONTROL
DIGITAL LINK

The Digital Sound Processor of the DSS930:

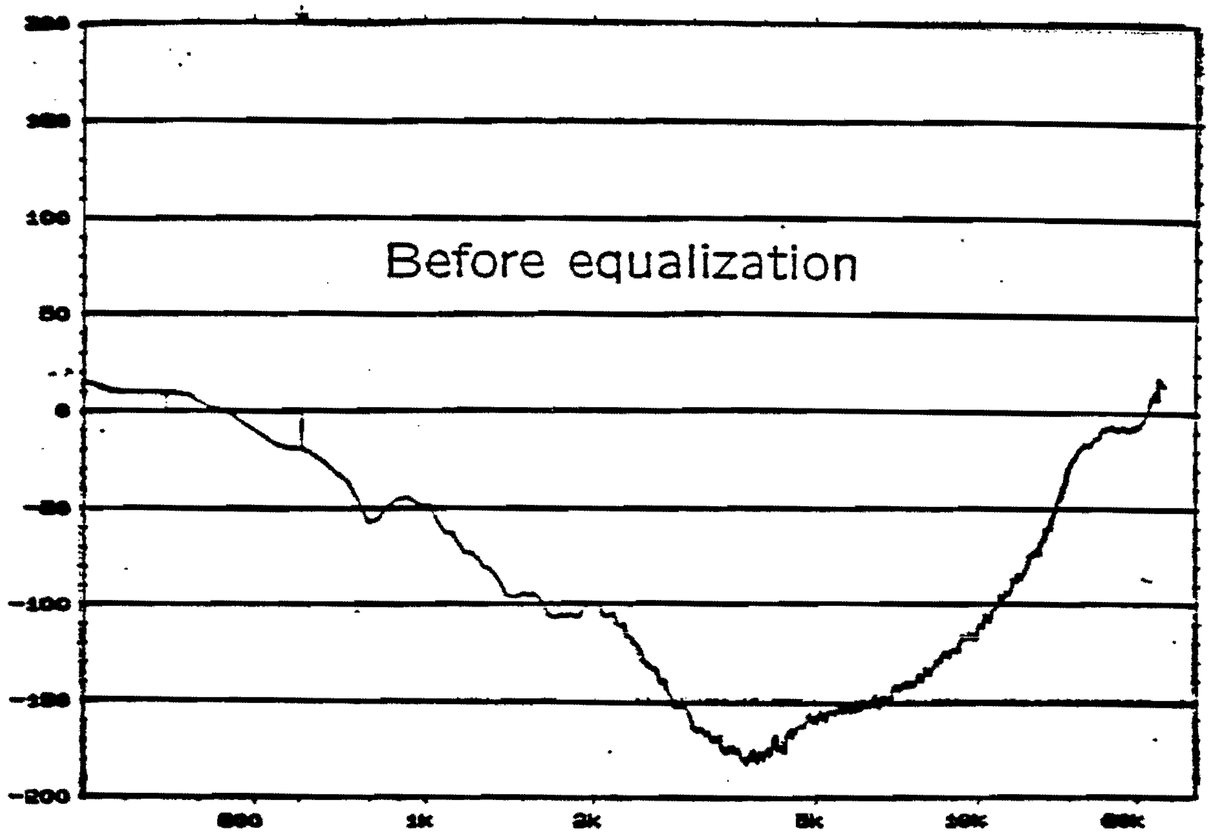
* linear amplitude (1.5 dB for 50 Hz to 20 kHz)
* linear phase (20 degrees for 350 Hz to 19 kHz)
* high sonic performance
* equalization is adapted to the human ear
* the digital cross–over filter is a 85 tap FIR filter cascaded by five second order IIR filters
* for the tweeter a 21 tap FIR filter is cascaded by three second order IIR filters
* delay compensation between woofer/tweeter
* digital input 32, 44.1 and 48 kHz
* correct frequency is automatically chosen
* digital volume control
* two Bitstream D/A convertors
* microprocessor controlled

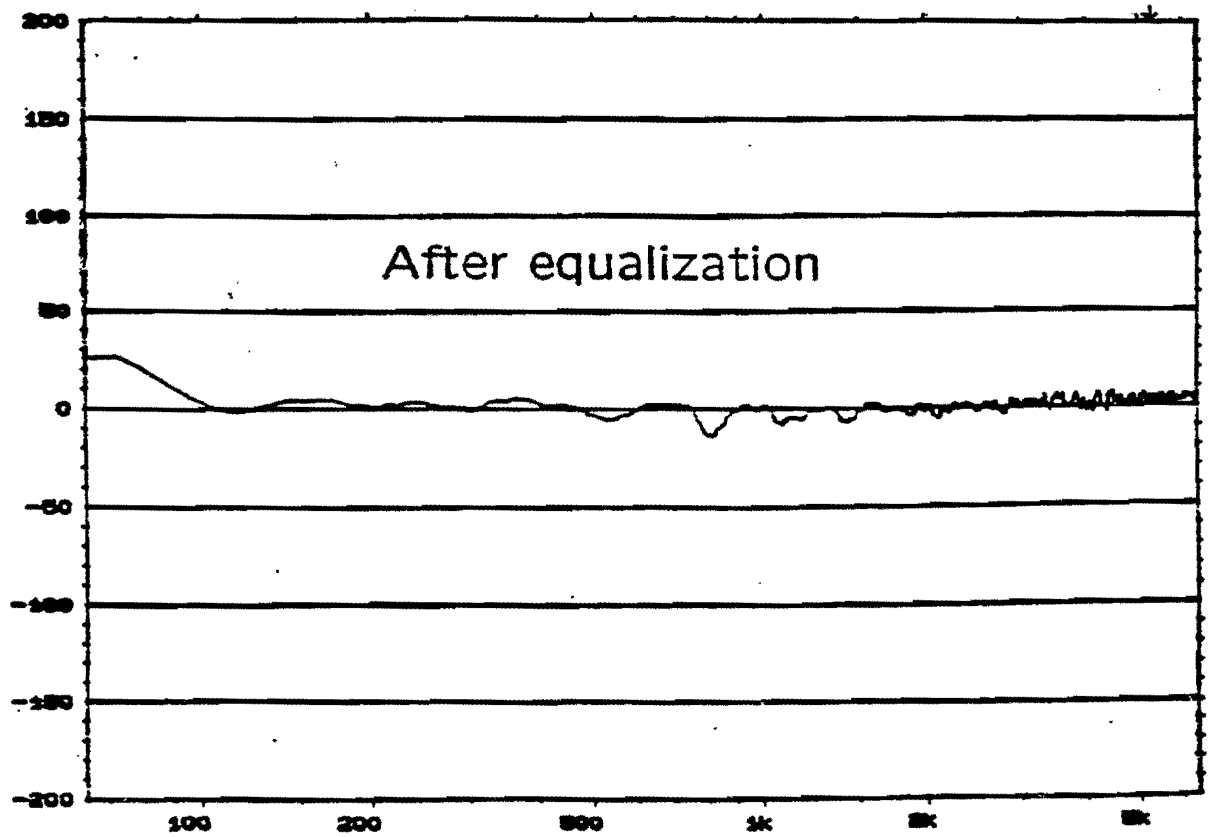
The amplitude response



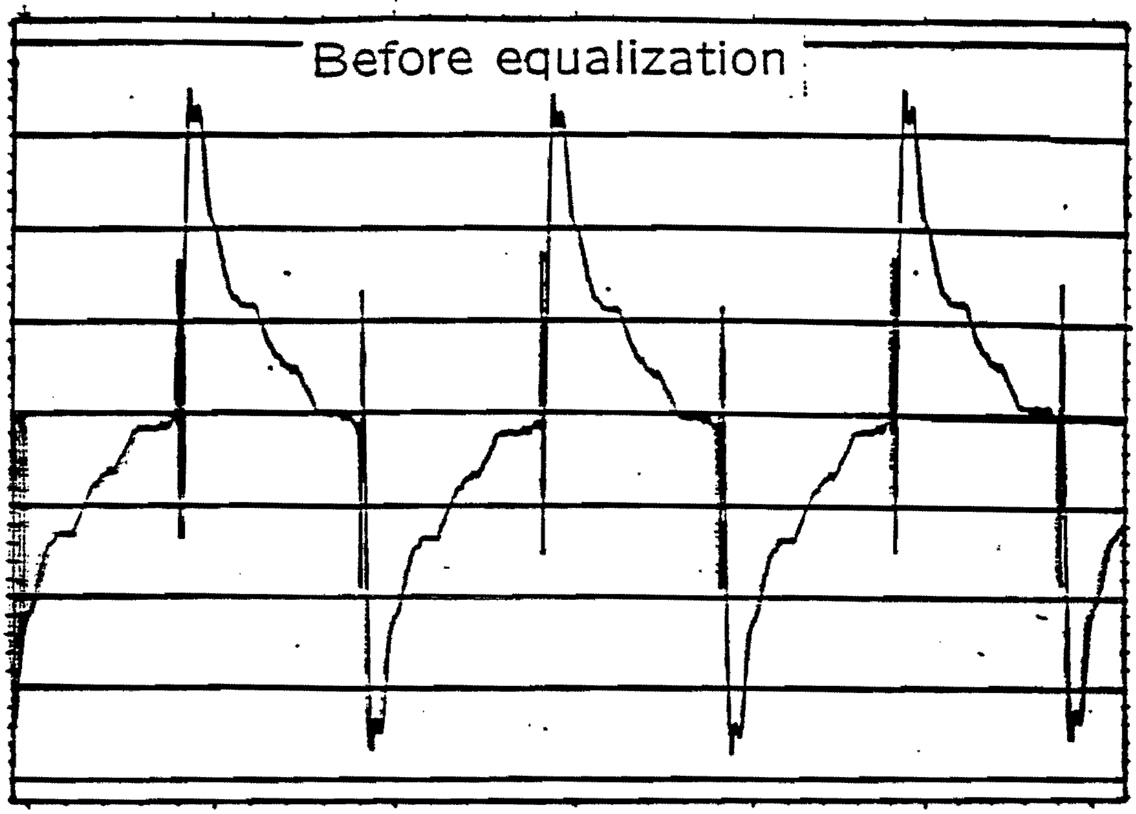


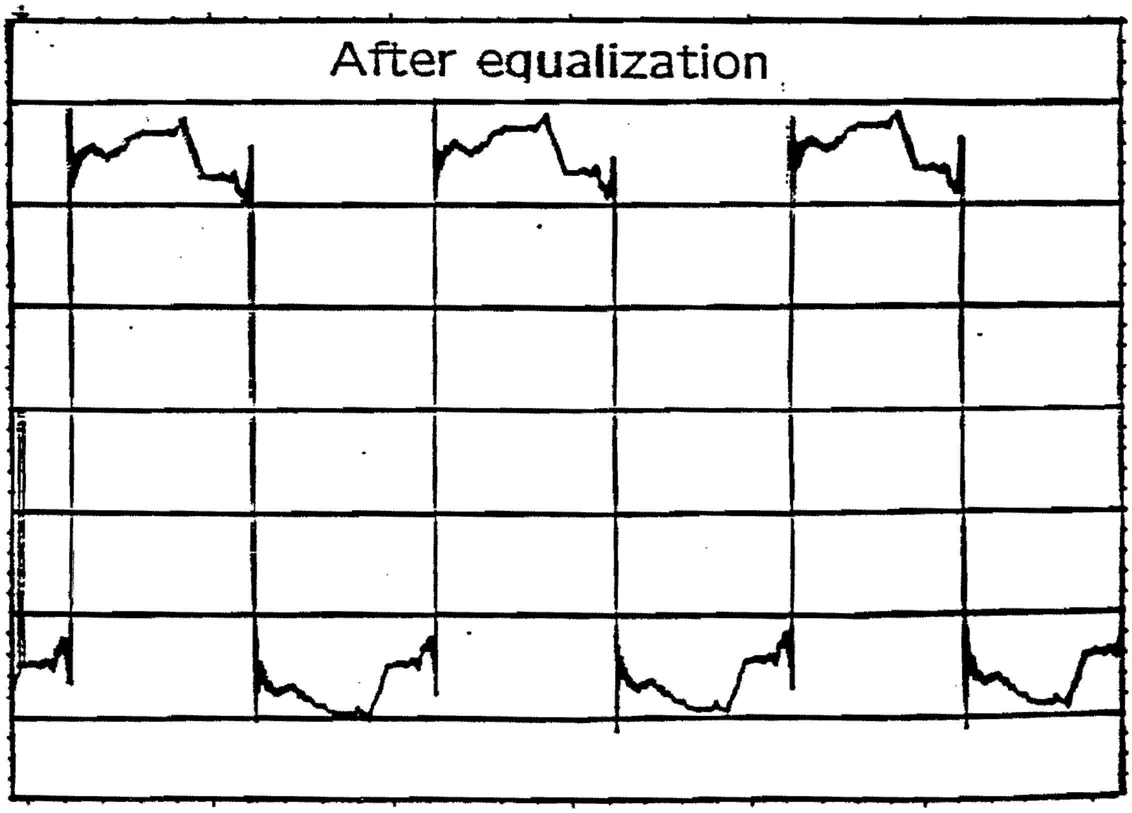
The phase response





The response to a 100 Hz square wave





The response to a 1 kHz square wave