

MASTER

Digitale besturing van de armbeweging voor random access op een compact disc

van der Meulen, Meine J.P.

Award date:
1988

[Link to publication](#)

Disclaimer

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

General rights

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

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

Technische Universiteit Eindhoven
Faculteit Elektrotechniek
Vakgroep Meten en Regelen

Digitale besturing van de armbeweging
voor random access op een
compact disc.

door M.J.P. van der Meulen

Verslag van afstudeeronderzoek,
verricht bij Philips, in de periode
oktober 1986 tot juni 1987
in opdracht van prof. ir. F.J. Kylstra,
en begeleid door ir. H.A.M Overath.
10 juli 1987.

De faculteit Elektrotechniek van de Technische Universiteit Eindhoven aanvaardt
geen aansprakelijkheid voor de inhoud van stage- en afstudeerverslagen.

Meulen, M.J.P. van der; **Digitale besturing van de armbeweging voor random access op een compact disc.** Afstudeerverslag, vakgroep ER, Technische Universiteit Eindhoven, juni 1987.

De groep 'digitale servo' van Philips werkt aan het digitaal implementeren van de regelingen in een compactdisc-speler. Binnen dit concept van digitaal regelen moet een procedure voor random access gerealiseerd worden. Zo'n procedure moet informatie op een compact disc zo snel mogelijk kunnen vinden en vrijgeven. Een verplaatsing in radiale en in tangentiële richting van de laserspot die de plaat aftast realiseert de zoekactie. De radiale beweging -spoor-springen genaamd- is het onderwerp van dit afstudeeronderzoek.

Voor het krijgen van inzicht in random access wordt de oude implementatie geanalyseerd en de problemen erin gelocaliseerd. Deze problemen betreffen vooral het uitrekenen van de spronglengte en de snelheidsregeling tijdens het spoor-springen. De analyse van deze twee problemen vormt de basis voor de random access procedure binnen het digitale concept. In de eerste plaats leidt ze tot een nauwkeurige spronglengteberekening en ten tweede tot een andere realisatie van de regelingen, waarbij het accent ligt op de controleerbaarheid ervan. Optimale controleerbaarheid wordt ook bewerkstelligd door het hanteren van een snelheidsprofiel, dat is afgeleid van een trapeziumvormig profiel.

Met de aldus gerealiseerde random access procedure is het mogelijk sprongen accuraat, dit door het nauwkeurig berekenen van de spronglengte, en gecontroleerd, door het gebruik van optimale regelingen, uit te voeren. De tijd nodig voor random access kan binnen een seconde voor de hele plaat en een halve seconde voor een derde van de plaat zijn, eisen voor de volgende generatie compactdiscspelers.

Meulen, M.J.P. van der; **A digital implementation of the radial control system for random access in compact disc.** Master's degree thesis, Measurement and Control Section ER, Eindhoven University of Technology, June 1987.

The Philips 'Digital Servo' group is working on digital implementation of control loops in a compact disc player. Within this digital concept a random access procedure has to be realised. Such a procedure is used for locating information on a compact disc. The search is performed by a tangential and radial movement of the laser-spot. The radial movement, called track-jumping, is the subject of this report.

A study of the old implementation of random access gives insight into the matter and also reveals problems concerning jump-length and controllability of the arm during track-jumping. The analysis of these problems forms the basis for the digital implementation of random access. A more accurate jump-length calculation and a better performance of the control loops (improved controllability and stability) have thus been attained. Optimal controllability is also achieved by using a trapezium-related speed-profile.

The characteristics of this new implementation are:

- high accuracy, achieved by a better calculation of the required jump-length, and an improved realisation of the jump.

- improved controllability, through the use of more optimal control loops.

Access-times can now be reduced to less than one second for a jump equivalent to 60 minutes of music and less than half a second for a 20 minute jump. These times are within the bounds for the next generation of compact disc players.

Inhoudsopgave

1	Inleiding.	5
2	Analyse van random access.	7
2.1	Signalen voor radiële regeling in CM100.	7
2.2	Realisatie random access in CM100.	8
2.3	Modellering spoorpringen.	11
2.3.1	Snelheidsregeling met terugkoppeling.	11
2.3.2	Het lineaire model.	12
2.3.3	Bandbreedte van de snelheidsregeling.	12
2.3.4	Controle model voor spoorpringen.	13
2.3.5	Versnel- en vertraagpulsen.	13
2.4	Opmerkingen over huidige regeling.	14
2.5	Herziene probleemstelling.	15
3	Theoretische achtergrond spoorpringen.	16
3.1	Bepaling spronglengte.	16
3.2	Beschikbare regelsignalen.	16
3.3	Profiel van een sprong.	17
3.4	De positierегeling.	18
3.4.1	Dimensionering van de positierегeling.	19
3.5	De snelheidsregeling.	21
3.6	Overgangen tussen de verschillende regelingen.	21
3.7	Voorsturing omwentelingssnelheid van plaat.	22
4	Hardwareimplementatie.	24
4.1	Hardwareomgeving.	24
4.2	De buslogica.	25
4.3	Vierkwadrantenteller.	25
4.4	Snelheidsmeting.	26
4.5	Positieteller.	27
5	Softwareimplementatie.	28
5.1	Het hoofdprogramma.	28
5.2	De playerroutine.	29
5.3	De calcroutine.	29
5.3.1	De calculroutine.	30
5.4	De jumproutine.	31
5.5	De interruptroutine.	31

6	Test van sporspringprocedure.	34
6.1	De testopstelling.	34
6.2	Radiele regeling.	35
6.2.1	Meting overdracht radiele regeling.	35
6.2.2	Evaluatie berekende regeling.	35
6.2.3	Evaluatie geoptimaliseerde volgregeling.	36
6.2.4	Test van berekende regeling tijdens sporspringen.	36
6.2.5	Test van geoptimaliseerde regeling voor sporspringen.	37
6.2.6	Evalueren geoptimaliseerde sprongregeling.	38
6.3	De spronglengte en sprongtijd.	38
6.3.1	Sprongnauwkeurigheid bij sprong kleiner dan N_{ref}	38
6.3.2	Sprong bij een lengte groter dan N_{ref}	39
6.3.3	Relatie spronglengte en sprongtijd.	39
7	Conclusies en aanbevelingen.	41
A	Verband snelheid arm en [DAC2,DAC1,DAC0].	
B	Overdracht stroom door arm naar snelheid ten opzichte van compact disc.	
C	Verband tussen tijd en aantal sporen op een compact disc.	
D	Schema's van de hardware.	
E	Timing van de hardware.	
F	Programmering PAL.	
G	Flowdiagrammen.	
H	programma's voor PCB5010.	
I	Resultaten en metingen.	

Hoofdstuk 1

Inleiding.

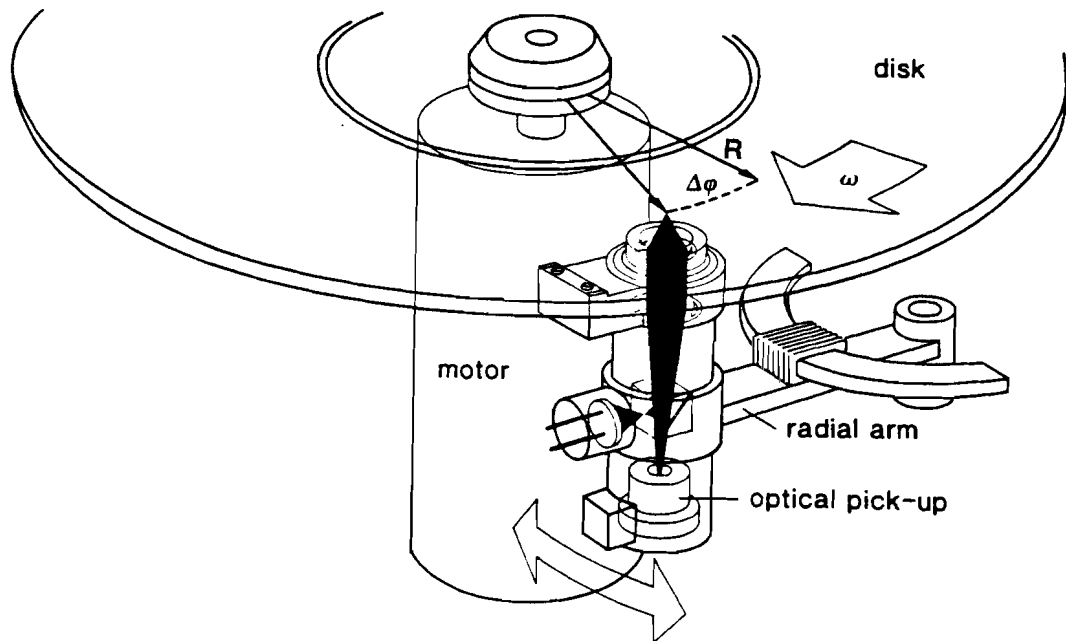
Een combinatie van technieken van verschillende vakgebieden, zoals lasertechnologie, materiaalkunde, optica, en regeltechniek, heeft in de jaren zeventig geleid tot de uitvinding van een nieuw medium voor opslag van gegevens: de compact disc. De zeer grote opslagcapaciteit in verhouding tot de omvang, de hanteerbaarheid, de reproduceerbaarheid, de relatieve onkwetsbaarheid etc. hebben geleid tot toepassingen in een aantal gebieden. De populairste hiervan is het opslaan van muziek. De voordelen ten opzichte van oude systemen als langspeelplaat en cassette zijn groot: de compact disc is klein, vertoont geen vervorming als tikken en ruis, en is gebruikersvriendelijk. Daarnaast is ze bruikbaar voor opslag van gegevens, zoals een encyclopedie, programmatuur, of boeken. In dit geval valt de grote opslagcapaciteit het meeste op: de telefoonnummers van Nederland passen gemakkelijk op één compact disc.

Op een compact disc staat informatie gecodeerd in een spiraalvormig spoor van putjes. Dit spoor heeft een spoed tussen de 1,5 en 1,7 μm en moet met een constante snelheid tussen 1,2 en 1,4 m/s door een laserspot worden uitgelezen.

Om de informatie van een compact disc te lezen is er een samenspel van elektronica, mechanica en optica in de speler aanwezig. De optica behelst een laser met een focuserings-servo en fotodiodes voor detectie van informatie en regelsignalen van de compact disc. Errorcorrectie- en -detectie-elektronica verwerkt de informatie. De regelsignalen worden gebruikt om de laser op de goede plaats op de disc te positioneren en te focuseren. Hiervoor zijn, zoals in figuur 1.1 is te zien, drie regelingen aanwezig. De motorregeling zorgt voor de juiste omwentelingssnelheid van de disc, de focusregeling voor focusering van de laser op de plaat en de radiale regeling voor positionering van de laser op het spoor op de compact disc.

In een compactdisc-speler is een procedure aanwezig voor random access. Deze procedure wordt gebruikt om de laser van de ene plaats op disc te bewegen naar een andere, bijvoorbeeld wanneer de luisteraar een ander nummer wil horen. Deze beweging bestaat uit een radiale en een tangentiële verplaatsing over de plaat. De radiale beweging heet spoorpringen.

De regelingen zijn tot nu toe hoofdzakelijk analoog gerealiseerd met behulp van actieve filters. In de strijd om goedkope compactdisc-speler speelt integratie een grote rol. Nu is digitale elektronica eenvoudiger te integreren dan analoge, denk bijvoorbeeld aan condensatoren en weerstanden. Het is dus een logische stap om de regelingen digitaal te gaan implementeren. De groep 'digitale servo' van Philips werkt voor dit doel aan een speciale servoprocessor. Voor de ontwikkeling is een speciale testopstelling aanwezig met een signaalprocessor van het type PCB5010, die het uitvoeren van de regelingen voor zijn rekening neemt, en een microprocessor van het type 8051, die de nodige communicatie met de buitenwereld verzorgt.



Figuur 1.1: Schematische weergave van het loopwerk van een compact disc-speler.

Het spoorpringen is het onderwerp van het afstudeeronderzoek. Een procedure in de servoprocessor moet de radiale actuator met een gecontroleerde snelheid een gegeven aantal sporen over laten steken ten behoeve van random access. Deze procedure moet gerealiseerd worden met de testopstelling. De opdracht kent een aantal fases:

- inleerfase:
 - het compact disc systeem, met het accent op de regelingen
 - de bestaande realisatie van spoorpringen
 - de testopstelling
- theoretische studie spoorpringen
- ontwikkelen nieuwe spoorpringprocedure
- simulatie van de procedure
- realisatie van de procedure
- testen van de procedure.

Dit verslag beschrijft de ontwikkeling van een procedure voor het spoorpringen in een compactdisc-speler. Procedures zoals die in voorgaande generaties spelers bestonden vormen hiervoor de basis. Deze procedures worden zodanig veranderd dat ze passen in het concept 'digitale servo' en daarnaast ook voldoen aan nieuwe specificaties, die zwaarder zijn dan die voor spelers van de oude generatie ten aanzien van spoorpringen.

Hoofdstuk 2

Analyse van random access.

Er moet een nieuwe procedure komen voor random access in een cd-speler. Een analyse van de huidige realisatie is daarvoor een logische eerste stap. Dit hoofdstuk maakt die analyse, maar gaat wel uit van een redelijke voorkennis op het gebied van compactdisc-spelers. Voor dit afstudeerwerk zijn voor dit doel een aantal werken doorgenomen. De grondslagen van optische opslag van informatie zijn beschreven in [5]. Meer toegespitst op compact disc zijn de rapporten [2] en [4] waarin vrij nauwkeurig is te lezen hoe de regelingen functioneren, waarbij de tweede zich vooral richt op het uitrekenen van regelaars om te komen tot stootongevoelige compactdisc-spelers. Het verslag [1] gaat in op de random access procedure in een speler. In [6] is de uiteindelijke realisatie van een cd-rom-speler te zien.

De informatie is op de plaat gegroepeerd in zogenaamde frames van $1/75$ seconde die elk voorzien zijn van een tijdcode in minuten, seconden, en frames. De random access procedure rekent uit waar de gewenste tijdcode zich op de plaat bevindt ten opzichte van de plaats waar de laserspot zich bevindt, en positioneert de laserspot boven de informatie. Een radiale en een tangente verplaatsing van de laserspot realiseren deze positionering. In deze analyse ligt de nadruk op de radiale beweging. Deze is het gevolg van rotatie van de arm en wordt spoorspringen genoemd.

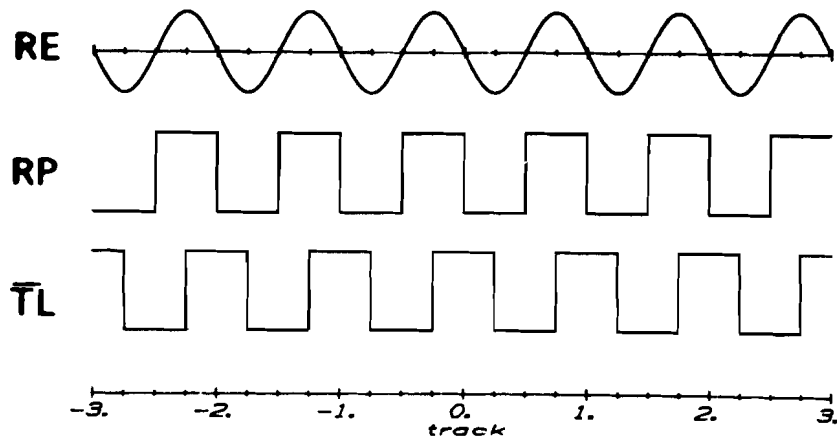
Voor het vormen van modellen en inzichten in random access is gemeten aan een CM100 speler. Deze speler heeft het voordeel dat hij niet in vergaande mate is geïntegreerd, zodat de meeste signalen beschikbaar zijn voor metingen. Zo wordt de basis voor de digitale implementatie gelegd.

2.1 Signalen voor radiale regeling in CM100.

In de CM100 speler zijn een aantal signalen die gebruikt worden voor radiale regeling en spoorspringen in het bijzonder. Het gaat om:

- Het RE-sigitaal (Radial Error zie figuur 2.1). Het RE-sigitaal signaal is een sinus als functie van de positie en heeft een periode gelijk aan de spoed, met nulpunten op en midden tussen de sporen. Omdat in de buurt van het spoor het RE-sigitaal evenredig is met de positieafwijking is het geschikt als regelsigitaal voor het maken van een positie-regeling op een spoor.
- Het \overline{TL} -sigitaal (Track Loss zie figuur 2.1). Het \overline{TL} -sigitaal is hoog wanneer de laserspot zich op het spoor bevindt, en laag als hij tussen de sporen is. Dit sigitaal wordt gebruikt voor het detecteren van een track loss: de arm volgt het spoor niet meer. De μP neemt in dat geval de controle over en regelt de arm weer terug naar het spoor.

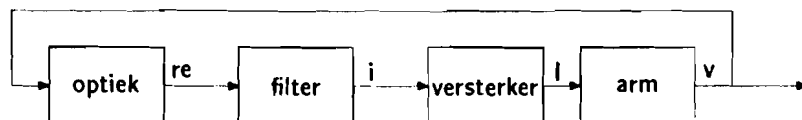
- Het RP-sigitaal (Radial Polarity zie figuur 2.1). Het RP-sigitaal is een sigitaal, dat hoog is als RE positief, en laag als RE negatief is. Het sigitaal is 90 graden in fase verschoven ten opzichte van \overline{TL} en deze geven dus samen eenduidige positie- en richtingsinformatie.



Figuur 2.1: \overline{TL} , RP, en RE-sigitaal als functie van radiale positie.

2.2 Realisatie random access in CM100.

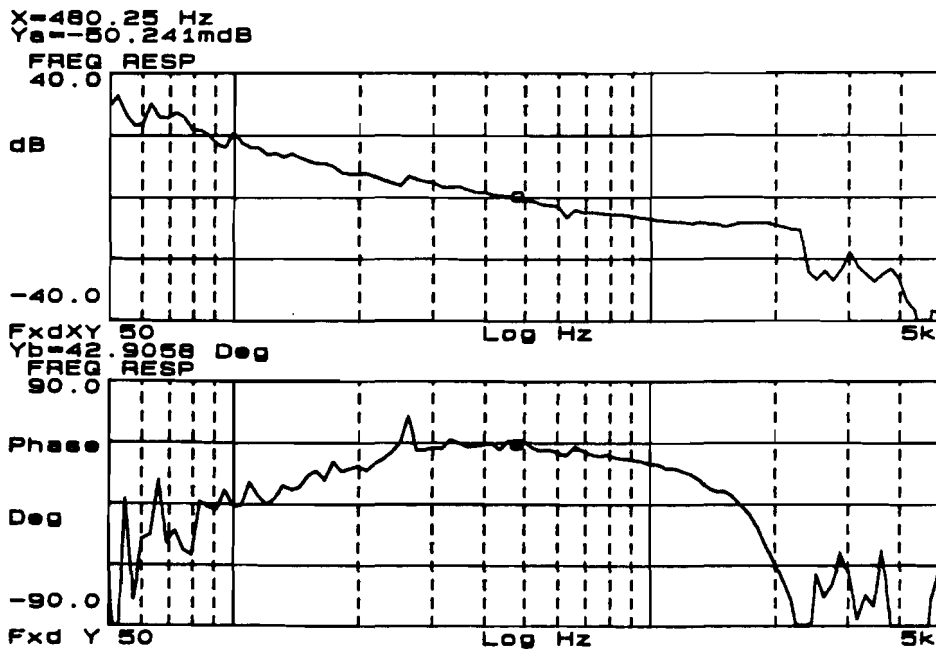
In normaal bedrijf regelt een filter de positie van de arm zo dat hij het spoor volgt, zie figuur 2.2. Het RE-sigitaal is hierbij een maat voor de positieafwijking van de arm ten opzichte van het spoor, en is het ingangssigitaal van de regelaar. Het filter bestaat uit een lag- en een leadfilter. De openloop-overdracht die dit filter realiseert is gegeven in figuur 2.3.



Figuur 2.2: Realisatie radiale regeling.

Wanneer nu de speler de opdracht krijgt om te springen schakelt de μP deze regeling uit en komt de spoor springregeling in bedrijf. In een blokdiagram ziet de regellus er dan uit als figuur 2.4. De volgende blokken zijn hierin aan te wijzen:

optiek De optiek tast met behulp van een laser de plaat af. De signalen die van de fotodiodes komen worden versterkt en verwerkt door elektronica. Dit levert onder meer de signalen RP, en \overline{TL} .



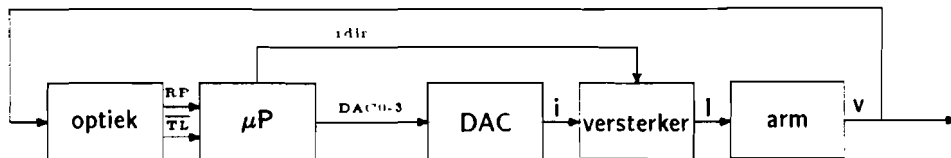
Figuur 2.3: Overdracht radiele regeling.

μP In deze regellus regelt de μP het spoorpringen. Uitgaande van de informatie van TL en RP, worden de kommando's voor de arm berekend. De uitvoer bestaat uit de signalen DAC0-3, die de stroomsterkte bepalen, en een richtingsbit RDIR.

DAC De digitaal-analoog-omzetter.

versterker De vermogenstrap voor de radiele arm. Deze versterkt het signaal van de D/A-omzetter en inverteert het als het richtingsbit RDIR=1 is.

arm De arm is een actuator die de radiele positie van de laserspot bepaalt. Stuurstromen van het blok versterker versnellen en vertragen de arm door middel van een lineaire motor.



Figuur 2.4: Realisatie regelsysteem voor spoorpringen.

Een uitgebreidere beschrijving van deze blokken is te vinden in de verslagen van Menno Riemersma [1] en Govert Geelen [2].

Random access.

Stel: er wordt gevraagd om data van de plaat met een bepaalde tijdcode. Deze tijdcode is op de plaat gecodeerd en geeft zo een absolute referentie over de plaats van de informatie. De sporen zijn van binnen naar buiten genummerd. In een tabel in de μP is van elke minuut opgeslagen op welk spoor die zich bevindt. Interpolatie levert dan van elke tijdcode het spoornummer. Zo kan het spoornummer van het gewenste en het actuele

spoor bepaald worden, en is de spronglengte bekend. Met dit aantal sporen wordt de sporspringprocedure aangeroepen.

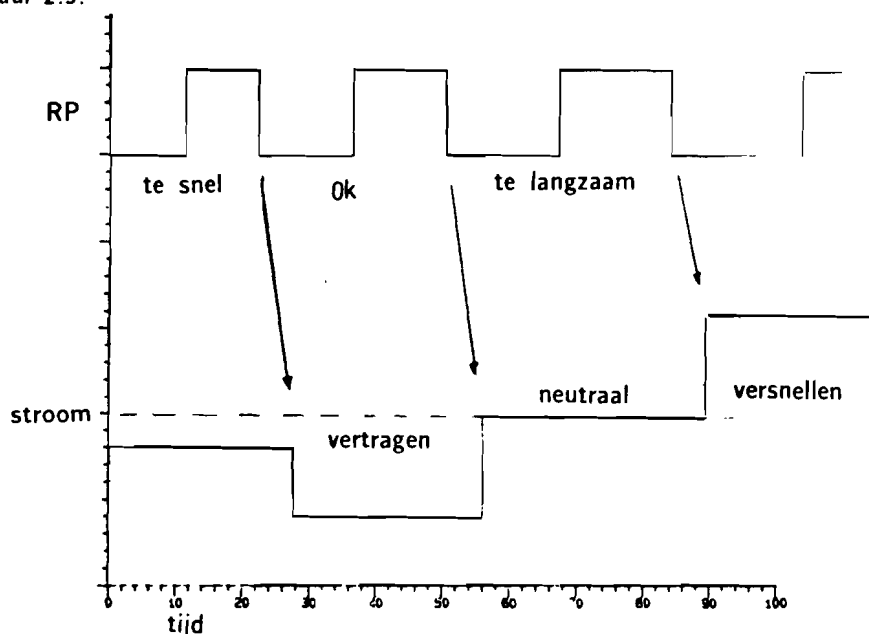
De sporspringprocedure kent drie soorten sprongen. De eerste is een sprong van één spoor. Hij realiseert sprongen kleiner dan 10 sporen door de sprongprocedure een aantal keren na elkaar de opdracht te geven één spoor te springen. De tweede is langer, tot 256 sporen, en heet 'langzaam springen'. En de derde is de langste, boven de 256 sporen, de 'snelle sprong'.

Eén spoor wordt overgestoken door de arm een stroompuls te geven van 0,7 A. De arm gaat nu bewegen naar het naastliggende spoor. Het laaggaan van het \overline{TL} -signaal beëindigt de puls, en zodra het \overline{TL} -signaal weer hoog wordt, geeft de μP puls met dezelfde tijdsduur, maar met tegengesteld teken. De arm komt zo precies tot stilstand boven het naastliggende spoor.

Voor een snelle sprong versnelt de μP de arm door het geven van een puls van 0,7 A met een duur van ongeveer 1,12 ms. Daarna probeert de μP de periodetijd van het RP-sigitaal constant te houden op 80 μs , equivalent met een snelheid van de arm van 20 mm/s. Telling van het aantal opgaande flanken van het RP-sigitaal levert het aantal overgestoken sporen, en zo het moment waarop het afremmen begint. Op 64 sporen van het eindpunt vertraagt een puls van 0,5 ms van 0,7 A de arm en de μP houdt daarna de periodetijd constant op 300 μs , of 5 mm/s. Na deze laatste 64 sporen remt een puls van 0,7 A van 0,88 ms de arm af en de radiale regeling vangt de arm in op een spoor.

Bij een langzame sprong versnelt de μP de arm met een puls van 0,7 A met een duur van 0,88 ms, en regelt de periodetijd daarna op 300 μs , of 5 mm/s. Het afremmen gaat hetzelfde als bij de snelle sprong.

Bij snelheidsregeling meet een timer in de μP de periodetijd, en vergelijkt deze met de gewenste periodetijd. Een tabel in de μP bepaalt bij elke periodetijdafwijking een stuurstroom voor de arm, met een resolutie van 8 μs . Schematisch is dit weergegeven in figuur 2.5.



Figuur 2.5: Schematische weergave snelheidsregeling in CM100.

Na een sprong controleert de μP , door evaluatie van de op de plaat gegeven tijdcode, of er inderdaad op het goede spoor geland is. Als dit niet het geval is, dan vindt een

nieuwe sprong plaats. Zo wordt de gewenste plaats bereikt na een aantal sprongen: een snelle, één of twee langzame, en een paar enkele spoorovergangen. Voor een uitgebreidere beschrijving van deze procedure verwijs ik naar het verslag van Menno Riemersma [1].

2.3 Modelling sporspringen.

Tijdens de sprong zijn er twee principieel verschillende processen aan te wijzen. Ten eerste zijn er de versnel- en vertraagpulsen, hier is geen sprake van terugkoppeling, en ten tweede is er de snelheidsregeling tijdens de sprong, een regeling met terugkoppeling. Deze processen worden nu aan een onderzoek onderworpen.

2.3.1 Snelheidsregeling met terugkoppeling.

Voor het berekenen van de responsie van de snelheidsregeling is het nodig een model te hebben, dat de overdracht van de verschillende delen weergeeft. Voor het model vervangen eerste orde benaderingen de niet-lineariteiten.

In het geval van een snelheidsregeling is figuur 2.4 een mogelijke representatie van het teruggekoppelde systeem. In dit systeem zijn de verschillende onderdelen als volgt gemodelleerd:

Overdracht snelheid arm naar DAC0-3.

De arm heeft een snelheid v , de μP meet die door analyse van het RP-sigitaal. Een interne timer meet de periodetijd van dit sigitaal en een tabel levert het bijbehorende regelsigitaal. Appendix A geeft het verband tussen snelheid en DAC0, DAC1 en DAC2 in het geval van een snelle en een langzame sprong. Bit DAC3 wordt hierbij niet genoemd, omdat dit bit niet wordt gebruikt tijdens het regelen, maar alleen tijdens versnel- en vertraagpulsen (bij grote stromen). De benadering van deze grafieken is een eerste orde benadering in het instelpunt. Dit geeft voor de snelle regeling een overdracht 2.9 mm/s per eenheid en voor de langzame 0.20 mm/s per eenheid. Het feit dat periodetijd omgekeerd evenredig is met snelheid verklaart het grote verschil van deze constanten: een verschil van 8 μs geeft bij een hoge snelheid een veel grotere snelheidsverandering dan een lage snelheid.

Deze manier van regelen introduceert een zero-order-hold: het berekenen van elk nieuw uitgangssigitaal kost een bepaalde tijd waarin het vorige sigitaal nog in gebruik is.

Overdracht DAC0-3 naar I.

Een TTL-register met een aantal weerstanden vormt een D/A-omzetter (zie voor het schema het cd-rom-manual [6]) en levert i . Een opamp versterkt deze stroom 1500 keer tot I . In tabel 2.3.1 staat de resulterende overdracht van bits DAC0, DAC1 en DAC2 naar stroom i en stroom I . Een lineaire benadering van de karakteristiek heeft een helling van 20 mA per eenheid

Overdracht I naar snelheid arm.

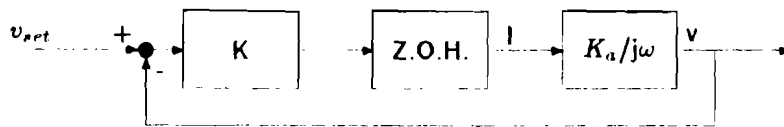
Voor de overdracht van de arm geldt $v/I = K_n / j\omega$, zoals beschreven in [4]. De constante K_n heeft een waarde van 22 m/A, de berekening hiervan is gegeven in appendix B. Bij deze beschrijving wordt de slewrate die ontstaat door de beperkte spanning van de stroombron verwaarloosd.

DAC2-0	dec	i in μA	I in mA
000	0	0,0	0
001	1	13,8	21
010	2	33,3	50
011	3	47,1	71
100	4	47,3	71
101	5	61,1	92
110	6	80,6	121
111	7	94,4	142

Tabel 2.1: Overdracht DAC0, DAC1 en DAC2 naar uitgangsstroom van DAC en powertrap.

2.3.2 Het lineaire model.

Het blokschema met de hierboven beschreven blokken is weergegeven in figuur 2.6. De waarde van K is voor een snelle sprong 6.9 A.s/m, en voor een langzame sprong 100 A.s/m. Als ingangssignaal is nu een snelheid gebruikt zodat nu een lineair model is ontstaan van een niet lineaire regeling.



Figuur 2.6: Blokschema snelheidsregeling spoorsspringen.

2.3.3 Bandbreedte van de snelheidsregeling.

Het model dat nu is gevonden van het spoorsspringen kan gebruikt worden voor het simuleren ervan. Een groot probleem hierbij is het niet tijd-uniform bemonsteren. Daarom wordt dit in het model tijd-uniform gedaan, met de sampletijd die optreedt bij de gewenste snelheid. Deze benadering is voor het grootste deel van de sprong toegestaan, omdat metingen aantonen dat de periodetijd ongeveer constant blijft.

De z-transformatie bij impulsvariante transformatie van de openloop overdracht van het regelsysteem van figuur 2.6 is:

$$G(z) = \frac{K \cdot K_a \cdot T}{z - 1} \quad (2.1)$$

Voor evaluatie van de bandbreedte van het systeem wordt de w-transformatie gebruikt.

$$z = \frac{1 + w}{1 - w} \quad (2.2)$$

Dit levert op:

$$G(w) = \frac{K \cdot K_a \cdot T \cdot (1 - w)}{2 \cdot w} \quad (2.3)$$

Voor de frequentie-karakteristiek wordt w geschreven als $w = \sigma + j \cdot \omega_w$, en alleen het imaginaire deel geëvalueerd. Dan geldt voor de bandbreedte:

$$|G(j\omega_w)| = \frac{K \cdot K_n \cdot T}{2} \sqrt{\frac{1 + \omega_w^2}{\omega_w^2}} = 1 \quad (2.4)$$

of:

$$\omega_w = \sqrt{\frac{K^2 \cdot K_n^2 \cdot T^2}{4 - K^2 \cdot K_n^2 \cdot T^2}} \quad (2.5)$$

Hieruit volgt met $\omega = \frac{2}{T} \arctan(\omega_w)$ dat bij de langzame en snelle sprong de bandbreedte respectievelijk 528 Hz en 25 Hz is. Dit verschil ontstaat door het regelen met behulp van een periodetijdmeting en is in principe ongewenst.

2.3.4 Controle model voor sporspringen.

Metingen aan de CM100-speler kunnen de theoretische analyse ondersteunen.

De K_n van de arm wordt gemeten met behulp van de fourier analyser. Een meting van de overdracht van stroom naar positie van de arm bij 500 Hz levert een K_n van 19 m/A, wat redelijk overeen komt met de theoretisch bepaalde waarde van 22 m/A.

Ook kan gecontroleerd worden of de gevonden bandbreedtes korrekt zijn. De reductie van de excentriciteit van de plaat door de regeling kan hiervoor een maat zijn. Volgens de specificaties van het compactdisc-systeem is de maximale excentriciteit 300 μm . De excentriciteit van de plaat kan dus -worst case- beschreven worden met:

$$exc(t) = 150 \cdot 10^{-6} \cdot \sin \frac{v \cdot t}{r} \quad (2.6)$$

De volgende formule geeft dan de snelheid van de excentriciteit:

$$exc'(t) = 150 \cdot 10^{-6} \cdot \frac{v}{r} \cos \frac{v \cdot t}{r} \quad (2.7)$$

De amplitudo hiervan is maximaal voor $v=1.4$ m/s en $r=25$ mm: 8.4 mm/s bij een frequentie van $\frac{v}{r} = 9$ Hz. Wanneer nu de gevonden bandbreedtes correct zijn, moet worden gevonden dat de excentriciteit bij een langzame sprong geheel wordt gereduceerd, en dat bij een snelle sprong een gedeelte moet overblijven. De bandbreedte van de snelle sprong is 25 Hz, er moet dus verwacht worden, dat bij een omloopfrequentie van 6 Hz van de excentriciteit een kwart overblijft. het RP-sigitaal bepaalt de snelheid van de arm ten opzichte van de plaat. De periode van het RP-sigitaal blijkt bij een snelle sprong te variëren tussen 78 μs en 92 μs . Dit komt overeen met een snelheidsvariatie van 1.7 mm/s. iets minder dan een kwart van 8.4 mm/s.

2.3.5 Versnel- en vertraagpulsen.

Tijdens de sprong vindt er op zekere tijdstippen een snelheidsregeling zonder terugkoppeling plaats. Als start van de sprong wordt de arm versneld met een startpuls, tijdens de sprong is er een vertraagpuls, en hij wordt beëindigd met een rempuls. Hier volgt een bespreking van het effect van de verschillende pulsen.

versnelpuls Deze puls heeft een stroomsterkte van 425 μA van de DAC, dus ongeveer 640 mA naar de arm. De arm heeft een overdracht $v/l=22/j\omega$ en krijgt dus een versnelling van 14 m/s^2 . De versnellingspuls duurt bij een langzame sprong 880 μs en zorgt voor een eindsnelheid van 12 mm/s. sneller dan de gewenste 5.3 mm/s.

Hij duurt voor een snelle sprong 1.12 ms en levert een eindsnelheid van 15.7 mm/s. veel langzamer dan de gewenste 20 mm/s.

vertraagpuls De vertraagpuls wordt alleen gebruikt in het geval van een snelle sprong, tussen het snelle en het langzame gedeelte van de sprong. Hij duurt 0,56 ms en zorgt voor een snelheidsverandering van 7,8 mm/s, veel te weinig om de snelheid van de arm genoeg omlaag te brengen naar 5,3 mm/s voor de langzame sprong.

rempuls De vertraagpuls is analoog aan de versnelpuls voor de langzame sprong, alleen met tegengestelde polariteit.

Conclusie kan dus zijn dat de verschillende pulsen niet optimaal zijn. Dit heeft echter geen merkbare invloed op de kwaliteit van het spoorpringen.

2.4 Opmerkingen over huidige regeling.

Sporen tellen tijdens de sprong.

Tijdens versnellingspulsen, zoals de startpuls, staat de μP in een wachtlus de tijd te meten. De processor telt dan niet de sporen die worden overgestoken tijdens de puls. Er worden zo tijdens de snelle sprong een aantal sporen gemist: 1-2 tijdens de startpuls, minimaal 9 tijdens de afrempuls naar het langzame gedeelte van de sprong en weer 1-2 bij de stoppuls. Dit alles komt de nauwkeurigheid van de sprong niet ten goede. De verwachting is dus dat de arm altijd te ver zal springen.

De minuten naar sporen tabel.

De arm springt elke keer te kort, niettegenstaande het in de vorige paragraaf genoemde feit. Dit ligt aan het feit dat de sporentabel gebaseerd is op een plaat met een spoed van 1,7 μm en een lineaire snelheid van 1,2 m/s. De tabel is zo gemaakt om er zeker van te zijn dat de arm nooit te ver springt. Het zal echter bijna nooit voorkomen dat een plaat hieraan voldoet, de sprong zal altijd te kort zijn. Het verschil in spronggrootte kan oplopen tot bijna 1000 sporen voor een maximale sprong bij een plaat met een spoed van 1,5 μm en een lineaire snelheid van 1,4 m/s.

Betrouwbaarheid signalen.

De signalen RP en \overline{TL} blijken nogal wat storing te bevatten. Door het ontbreken van hysteresis op de comparatoren die zorgen voor generatie van deze signalen treedt er nogal wat jitter op bij de flanken. Deze jitter wordt in de CM100 geelimineerd in het programma voor het spoorpringen.

Naast deze jitter treden er geregeld spikes op. De ervaring leert dat over het algemeen gezegd kan worden, dat spikes korter dan 20 μs genegeerd mogen worden. De CM100 heeft evenmin last van deze spikes door de manier waarop het programma is geschreven.

Ook worden de signalen onbetrouwbaar boven een zekere radiale snelheid van de arm. Voor het RP-signaal wordt over het algemeen een bovengrens van 80 mm/s aangehouden en voor het \overline{TL} -signaal 20 mm/s. De arm in de CM100 beweegt nooit sneller dan 20 mm/s en ondervindt daar dus ook geen last van.

Verhogen van de betrouwbaarheid van deze signalen lijkt wenselijk omdat ze in andere toepassingen voor veel problemen kunnen zorgen, maar dit valt buiten het kader van dit onderzoek.

2.5 Herziene probleemstelling.

Uitgaande van de opdracht en de evaluatie van het huidige systeem van spoorspringen kom ik tot een herziening van de probleemstelling.

Het doel van het onderzoek is het maken van een systeem dat het mogelijk maakt een bepaald adres op een plaat op te zoeken, en vanaf dat adres data vrij te geven. De radiale verplaatsing van de arm vormt hierbij de hoofdzaak, omdat andere leden van de groep de motorregeling ontwikkelen: de tangentiële richting.

Dit systeem behelst dus berekening van de radiale positieverandering van de laserspot en uitgaande hiervan, besturing van de arm naar die positie. Een deel van het onderzoek is dus het maken van regelingen die de arm tijdens een sprong besturen. Eerste eis is dat deze regelingen discreet worden gerealiseerd. Als zodanig wordt dit systeem ingepast in het project 'digitale servo'.

De spoorspringregeling zal als ingangssignalen het RE, RP en \overline{TL} -signaal gebruiken. Uitgaande van deze signalen wordt digitale informatie over de toestand van het systeem bepaald, en daarmee wordt het proces bestuurd. In principe is de uitvoering van dit proces een analogon van de bestaande routine in de CM100, maar analyse van dit systeem brengt een aantal feiten aan het licht die voor verbetering vatbaar zijn. Dit leidt tot de volgende extra eisen voor het spoorspringproces:

- Verhogen van de accuratesse van het spoorspringen. In de CM100 leidt onnauwkeurige berekening van de spronglengte tot meerdere sprongen om een bepaalde positie te bereiken. De reden hiervoor is dat een worst case tabel de basis is voor het uitrekenen van deze spronggrootte. Dit moet beter kunnen.
- Verhogen van de bandbreedte van de snelheidsregeling. Verhoging van deze bandbreedte leidt tot een hogere reductie van stoorsignalen.
- Tijduniforme en lineaire realisatie van de regelaars.

Tevens leiden nieuwe ontwikkelingen in de compactdisc-technologie tot de volgende eis:

- De accesstijd voor een sprong van het begin van de plaat naar het einde ervan, mag niet meer dan een seconde duren, voor een derde van de plaat is dit een halve seconde.

Als laatste wordt -in verband met de betrouwbaarheid- in door Philips gebouwde systemen de volgende eis voor openloop-overdrachten gesteld:

- De fasemarge moet bij elke regeling minimaal 40 graden zijn.

Hoofdstuk 3

Theoretische achtergrond sporspringen.

Nu bekend is hoe de procedure voor random access eruit zal gaan zien is het mogelijk een theoretische basis hiervoor te leggen. Dit hoofdstuk behandelt eerst het berekenen van de spronglengte en daarna het uitvoeren van de sprong.

3.1 Bepaling spronglengte.

De spronglengte is het verschil van het spoornummer van het actuele en het gewenste spoor. Stel nu: het spoor op $t=0$ s heeft nummer 0, en de rest van de sporen wordt van binnen naar buiten genummerd in radiale richting.

De relatie tussen tijd, t , (in seconden) en spoornummer, N , is (zie appendix C):

$$t = \frac{2 \cdot \pi \cdot r_0 \cdot N}{v} + \frac{\pi \cdot s \cdot N^2}{v} \quad (3.1)$$

of:

$$N = \frac{\sqrt{(2 \cdot \pi \cdot r_0)^2 + 4 \cdot \pi \cdot s \cdot t \cdot v} - 2 \cdot \pi \cdot r_0}{2 \cdot \pi \cdot s} \quad (3.2)$$

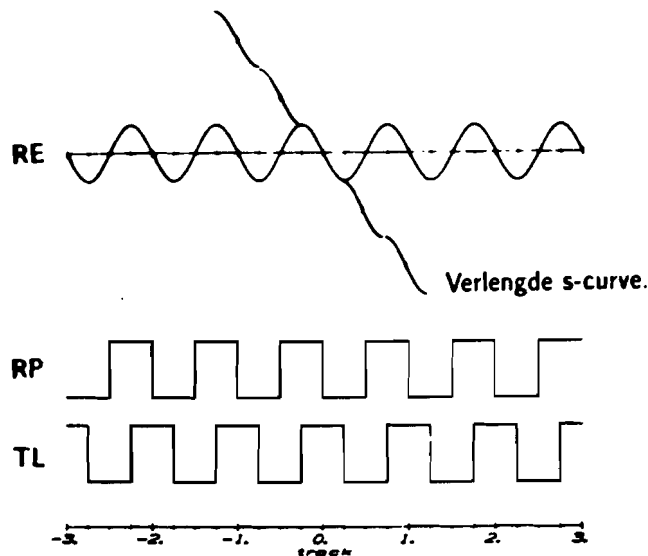
Het is dus nodig dat de spoed, s , en de lineaire opnamesnelheid, v , van de plaat bekend zijn.

3.2 Beschikbare regelsignalen.

Voor het maken van regelingen, is het nodig terugkoppelsignalen te hebben. Voor een positie-regeling gaat het om een radiale positie, en voor een snelheids-regeling om een radiale snelheid.

radiale positie In figuur 2.1 is het verband tussen radiale positie en het RE, TL en RP-signaal weergegeven. RE en TL zijn negentig graden in fase verschoven, en geven dus eenduidige positie-informatie. De resolutie ervan is een kwart spoorsteek. Deze positie-informatie bevat ook richtingsinformatie. De teller die de positie bevat noem ik de vierkwadrantenteller.

In de CM100-speler wordt het RE-signaal gebruikt als foutsignaal. Dit RE-signaal is slechts tot beperkte afstand van het spoor evenredig met de afstand, daarna is ze



Figuur 3.1: Positieinformatie als functie van afstand tot het spoor.

niet meer bruikbaar als positie-signaal. Oplossing van dit probleem is gevonden in het berekenen van de afstand tot het spoor uitgaande van het RE-signaal en de beschikbare positieinformatie in de vierkwadrantenteller. Dit levert een positie-indicatie als aangegeven in figuur 3.1, dit regelsignaal wordt een verlengde s-curve genoemd. Dit signaal kan dus gebruikt worden als regelsignaal tot een willekeurige afstand tot het spoor.

Bepijking van deze positieinformatie is de maximale snelheid van de arm. Deze snelheid mag niet hoger worden dan 20 mm/s, omdat dan het \overline{TL} -signaal niet meer betrouwbaar is.

Boven de 20 mm/s kan positieinformatie worden verkregen door het tellen van het aantal opgaande flanken van het RP-signaal. Bij deze snelheid mag er vanuit worden gegaan dat de richting van de arm hetzelfde blijft, en dat de telling zo een nauwkeurige bepaling van het aantal overgestoken sporen geeft. Dit signaal is betrouwbaar tot een radiale snelheid van 80 mm/s, daarna is het RP-signaal onbruikbaar.

snelheidsinformatie Meting van de periode van het RP-signaal levert de snelheidsinformatie. De snelheid van de arm is immers de spoorsteek gedeeld door de periodetijd. Dit signaal is eveneens betrouwbaar tot een radiale snelheid van 80 mm/s. Dit is een verbetering ten opzichte van de oude situatie omdat daar als regelsignaal de periodetijd van het RP-signaal werd gebruikt, en dit een niet-lineaire regelkarakteristiek tot gevolg heeft.

3.3 Profiel van een sprong.

De beschikbare regelsignalen bepalen het profiel van een sprong. Immers, de meest logische oplossing: een positie-regeling gaat niet op, omdat dan de maximale snelheid niet onder controle is. Het volgende profiel wordt gebruikt voor een sprong:

- In eerste instantie moet het gewenste spoor zo snel mogelijk genaderd worden, dit kan met een snelheid van 80 mm/s. Eerste stap is dus een snelheidsregeling van 80 mm/s.
- De positie-regeling komt in de buurt van het spoor in bedrijf. Dit kan door gebruik te maken van de vierkwadrantenteller en verlengde s-curve. De regeling zorgt ervoor dat de arm invangt op het juiste spoor.

- De positieregeling mag pas worden gebruikt voor snelheden onder de 20 mm/s. De tussenliggende stap is dus een snelheidsregeling naar 20 mm/s, waardoor de snelheid van de arm afneemt van 80 mm/s naar 20 mm/s.

De karakteristieken van de verschillende regelingen bepalen de schakelmomenten van de ene regeling naar de andere. Discussie hiervan vindt verderop in dit verslag plaats. Daarvoor geven de volgende hoofdstukken een beschrijving van de verschillende regelingen die voor de verwezenlijking van de verschillende stappen noodzakelijk zijn.

3.4 De positieregeling.

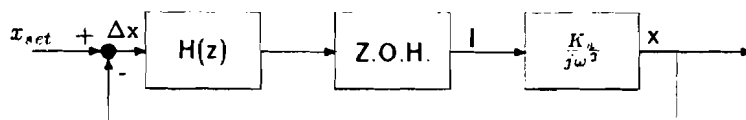
Voor het bepalen van een regelaar moet eerst de overdracht van het proces bekend zijn. In het s-domein geldt voor de overdracht tussen stroom door de arm en de positie ervan:

$$G_a(s) = \frac{K_a}{j\omega^2} \quad (3.3)$$

met $K_a=22$ m.s.s/A (zie appendix B). Dit wordt opgenomen in een systeem als gegeven in figuur 3.2. Om de discrete regelaar te berekenen ga ik over naar het z-domein via de impulsinvariante transformatie. De overdracht van de arm met inbegrip van de Z.O.H. is nu:

$$G_a(z) = K_a \cdot (1 - z^{-1}) \cdot \mathcal{S}\left\{\frac{1}{j\omega^3}\right\} = K_a \cdot \frac{z+1}{(z-1)^2} \cdot \frac{T^2}{2} \quad (3.4)$$

Dit is de openloop overdracht, met $H(z)=1$ A/m, als figuur 3.2 het regelsysteem weergeeft waarin de arm is opgenomen.



Figuur 3.2: Blokschema positieregeling spoorssprongen.

De eisen voor de positieregeling zijn de volgende:

- reductie van de excentriciteit. De amplitudo hiervan is namelijk 100 sporen en moet gereduceerd worden tot minder dan 0.1 spoor, wat een minimum reductie van 60 dB betekent. Excentriciteit heeft een frequenties tussen 3 en 9 Hz.
- de regeling moet zo veel mogelijk kritisch gedempt zijn. Dit vereist een speciale orientatie van de dominante polen en nulpunten. Dit staat nauwkeurig beschreven in Kuo [7].

Realisatie van deze eisen gebeurt door middel van lag- en leadfilters:

$$F(z) = \frac{1 - p_1}{1 - z_1} \cdot \frac{z - z_1}{z - p_1} \quad (3.5)$$

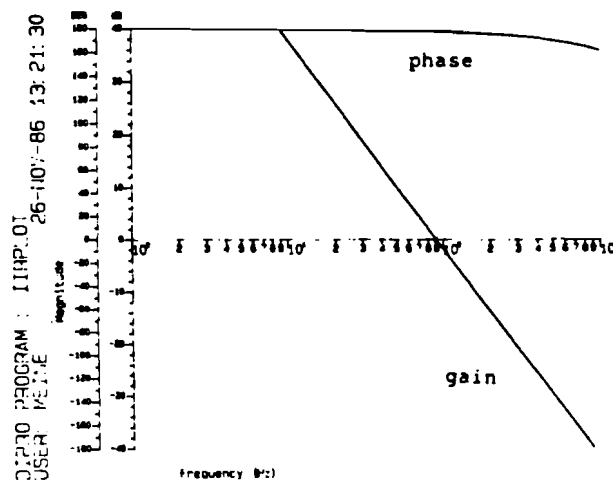
Deze filters hebben geen invloed op het steady-state gedrag van de regeling. Voor lag-filters geldt: $p_1 > z_1$ en voor lead-filters het omgekeerde.

3.4.1 Dimensionering van de positieregeling.

De overdracht van de arm is gegeven in figuur 3.3. De eis van een bandbreedte van 500 Hz en voldoende gain bij lage frequentie leidt nu tot de invoering van een lag-filter met een kantelpunt bij 30 Hz.

$$L(z) = \frac{101z - 99}{z - 1} \quad (3.6)$$

Er is gekozen voor een pool in $z=1$, omdat het leggen van een pool bij lage frequenties meestal problemen geeft met limit-cycles, zie literatuur [15] [16] [17] [18] [19]. Een pool in $z=1$ heeft dit probleem niet.



Figuur 3.3: Overdracht stroom door door arm naar positie van de arm.

De fasemarge van het systeem is nu nog 0 graden, dus het is instabiel bij terugkoppeling. Deze moet voor een betrouwbaar systeem minimaal 40 graden zijn. Een lead-filter met kantelpunten bij 90 en 3000 Hz zorgt voor deze fasedraaiing. Via het w -domein leidt dit tot een filter van de vorm:

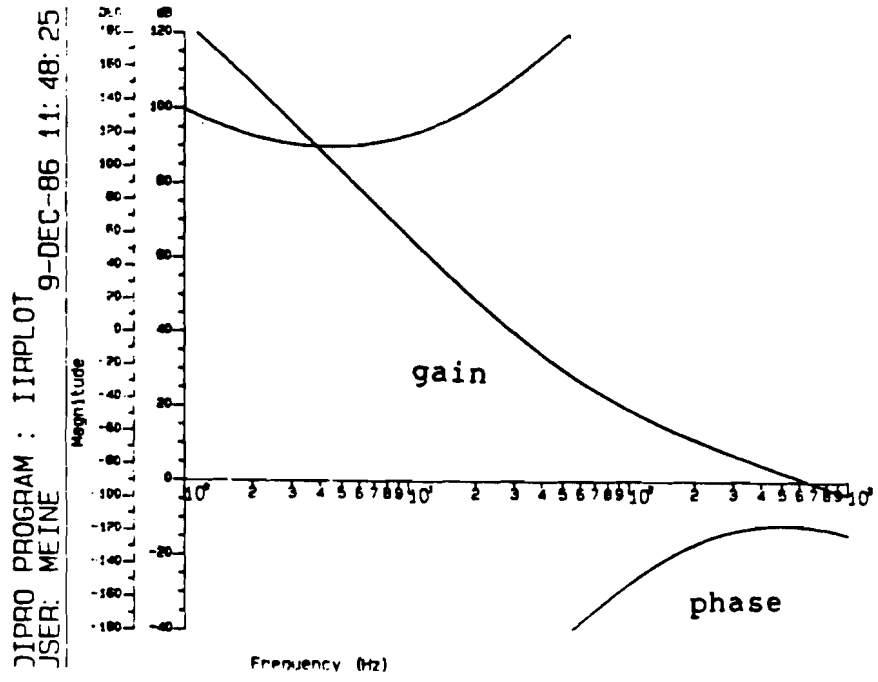
$$M(z) = \frac{36,3z - 34,3}{1,727z + 0,272} \quad (3.7)$$

De rootlocus-methode geeft nu de K-factor van het openloop-systeem. Bij een ζ van 0.7 leidt dit tot een K van 970. De frequentie-overdracht van het zo verkregen systeem is gegeven in figuur 3.4 en de rootlocus in figuur 3.5. Hieruit blijkt dat de fasemarge ongeveer 60 graden is: genoeg voor stabiliteit. De reductie van de excentriciteit is 70 dB, ook volgens specificatie.

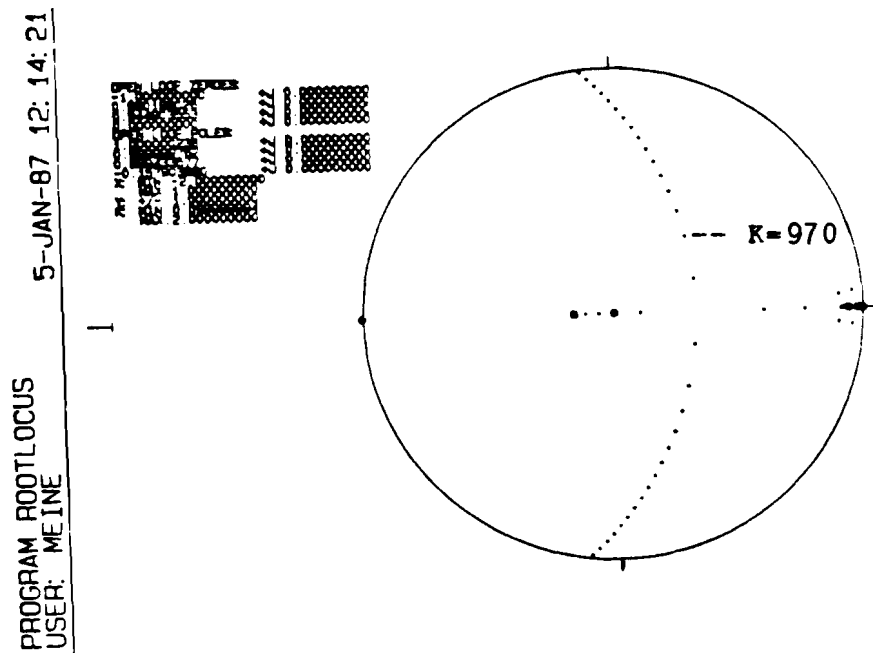
De totale regelaar ziet er dus zo uit:

$$H(z) = 970 \cdot \frac{36,3z - 34,3}{1,727z + 0,272} \cdot \frac{101z - 99}{z - 1} \quad (3.8)$$

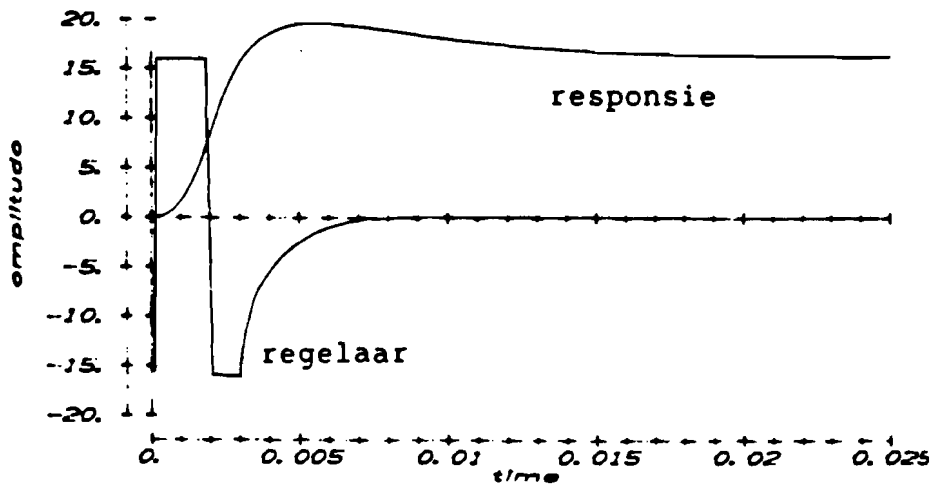
Met het simulatieprogramma Picosim is vervolgens een simulatie gemaakt van deze regeling, dit leidt tot een responsie zoals gegeven in figuur 3.6. Bij een sprong van 16 sporen, wordt het verlangde spoor binnen 25 ms op 0,1 spoor genaderd.



Figuur 3.4: Frequentieoverdracht van de regeling.



Figuur 3.5: Rootlocus van de regeling.



Figuur 3.6: Stapresponsie van de regeling met limiter.

3.5 De snelheidsregeling.

De snelheidsregeling wordt toegepast om een aantal sporen met zo groot mogelijke gecontroleerde snelheid over te steken.

De open loop overdracht van de stroom door de arm naar de snelheid is $\frac{K_a \cdot T}{j\omega}$ met een Z.O.H., na impulsinvariante z-transformatie:

$$H(z) = \frac{K_a \cdot T}{(z - 1)} \quad (3.9)$$

Een frequentiekarakteristiek van deze overdracht is gegeven in figuur 3.7. De regeling is altijd stabiel bij terugkoppeling, omdat de fasemarge 90 graden is. Het enige waarvoor gezorgd moet worden, is voldoende gain, om de snelheidscomponent van de excentriciteit te compenseren, en de eventuele andere verstoringen.

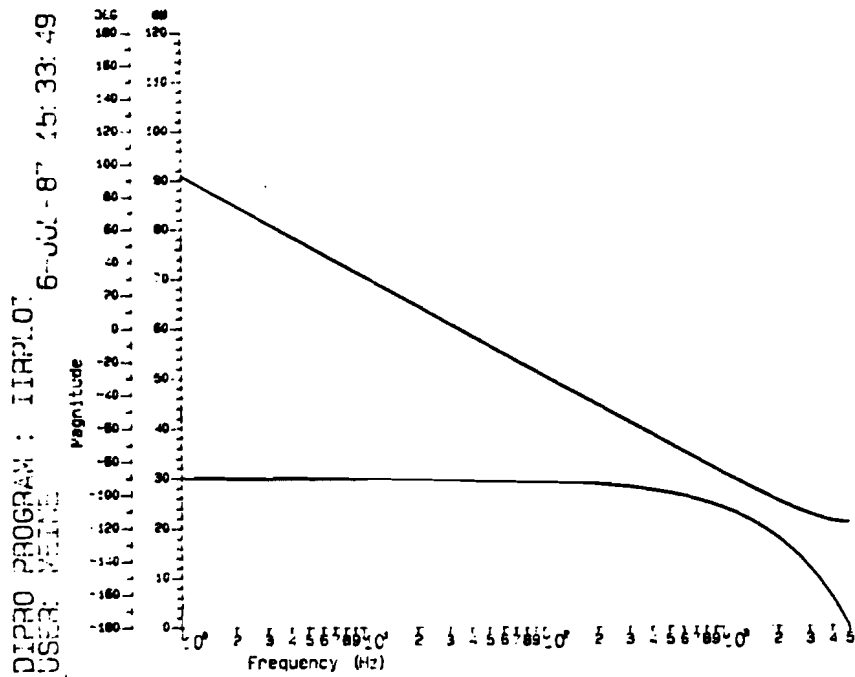
Een bandbreedte van 100 Hz zorgt voor voldoende reductie van de stoorcomponenten in de lage frequenties (excentriciteit). Dit leidt tot een versterking van 60 dB in de regelaar.

3.6 Overgangen tussen de verschillende regelingen.

Het profiel van het spoor springen heeft zoals besproken een paar schakelmomenten. Deze worden bepaald door:

- Het aantal sporen dat de positieregeling nodig heeft om de arm van 20 mm/s af te remmen tot stilstand.
- Het aantal sporen dat regeling nodig heeft om de arm af te remmen tot 20 mm/s.

Simulaties leveren het aantal sporen dat nodig is om af te remmen. Dit blijkt in 10 sporen prima te gaan.



Figuur 3.7: Overdracht stroom door door arm naar snelheid van de arm.

Het aantal sporen dat de regeling nodig heeft om af te remmen gegeven een snelheid v_b van de arm tot een gegeven eindsnelheid v_e kan worden berekend. Noem de maximale vertraging a , het aantal sporen dat wordt overgeslagen tijdens het remmen Δn en de tijd die geremd wordt Δt . Dan gelden de volgende vergelijkingen:

$$\Delta n \cdot s = \int_0^{\Delta t} (v_b - a \cdot t) dt = v_b \cdot \Delta t - \frac{1}{2} \cdot a \cdot \Delta t^2 \quad (3.10)$$

Ook:

$$\Delta t = \frac{v_b - v_e}{a} \quad (3.11)$$

Dan:

$$\Delta n \cdot s = \frac{v_b^2 - v_b \cdot v_e}{a} - \frac{(v_b - v_e)^2}{2 \cdot a} = \frac{v_b^2 - v_e^2}{2 \cdot a} \quad (3.12)$$

De overgang van tussen regelingen rond v_b en v_e is hiermee bepaald.

3.7 Voorsturing omwentelingsnelheid van plaat.

Voor de snelheid van random access is het zeer voordelig als de plaat op het moment dat een bepaald spoor wordt bereikt de juiste omwentelingsnelheid heeft. Lezen is dan immers meteen mogelijk. Dit voorsturen zal in de toekomst door de motorregeling met behulp van een tachokaart mogelijk zijn. De enige informatie die deze regeling nodig heeft is de gewenste snelheid. Deze snelheid is bekend omdat het gewenste spoor uitgerekend is.

De straal waarheen gesprongen wordt is:

$$r = r_0 + N \cdot s \quad (3.13)$$

met r_0 de straal van het binnenste spoor, s de spoed van het spoor, en N het spoor
waarheen gesprongen wordt.

Dan is de omwentelingssnelheid:

$$\omega = \frac{v}{r} = \frac{v}{r_0 + N \cdot s} \quad (3.14)$$

waarin v de schrijfsnelheid van de plaat is.

Hoofdstuk 4

Hardwareimplementatie.

Zoals in het vorige hoofdstuk is beschreven, leveren het \overline{TL} , RP, en RE signaal positie- en snelheidsinformatie. Deze bewerking gebeurt grotendeels softwarematig, maar, omdat de belasting van de anders processor te hoog is, ook met behulp van hardware.

Het gaat hierbij om een vierkwadrantenteller, een snelheidsmeting, en een sporenteller. Behandeling van deze drie schakelingen vindt in dit hoofdstuk plaats. Hierbij komen zowel de schema's als de timingen aan de orde.

Voor het begrijpen van de schakelingen is inzicht nodig in de hardwareomgeving. Een korte beschrijving van de processorkaart en de andere aanwezige kaarten is hiervoor voldoende.

Dan volgt een bespreking van de buslogica, waarlangs de drie meetschakelingen communiceren met de processor. Op deze basis stoelt de beschrijving van de vierkwadrantenteller, de snelheidsmeter, en de sporenteller. De schema's van deze schakelingen staan in appendix D, en de bijbehorende timingen in appendix E.

4.1 Hardwareomgeving.

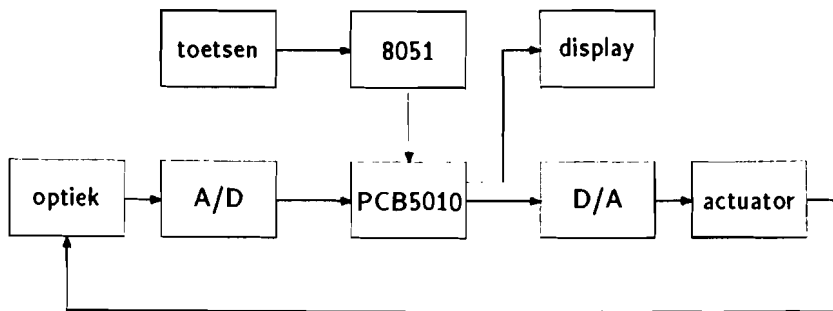
De structuur van de hardware is schematisch voor een regeling weergegeven in figuur 4.1. Naast deze hardware is er in de speler natuurlijk nog meer aanwezig voor bijvoorbeeld signaalverwerking en errordetectie.

De figuur moet als volgt gelezen worden. De optiek krijgt via een laserbundel informatie van de plaat. Elektronica bewerkt deze informatie tot nuttige signalen en deze worden doorgegeven aan een A/D-omzetter. De PCB5010 gebruikt deze digitale vorm voor het uitrekenen van de verschillende regelsignalen voor de verschillende processen, waarvan voor het gemak er maar een is weergegeven in de figuur. Een D/A-omzetter voert de regelsignalen van de PCB5010 naar de actuator. De bewegingen van het mechaniek wordt weer gedetecteerd door de optiek en zo is de regeling teruggekoppeld.

De 8051 is een processor die veel trager is en dient voor het detecteren van de commando's op het toetsenbord. Een seriële lijn geeft de commando's, die bestaan uit een woord instructie en soms een woord data, door aan de PCB5010 en die interpreteert ze en voert ze uit.

A/D-omzetting in het schema moet ruimer worden opgevat als een normale omzetting, er wordt elke conversie van analoge naar digitale informatie mee bedoeld. Dit houdt dus ook bijvoorbeeld de vierkwadrantenteller in.

Voor de A/D-omzetting is er al een kaart beschikbaar, namelijk een ADC-kaart, een kaart met vier analoog-digitaal-omzetters.



Figuur 4.1: Hardware realisatie regelproces.

Voor het blok D/A is er ook een kaart beschikbaar, een DAC-kaart, een kaart met vier digitaal-analoog-omzeters.

Voor het blok display is een kaart gebouwd waarmee een aantal led's kunnen worden aangestuurd. Deze led's kunnen twee woorden (16 bit) zichtbaar maken.

De kaarten die beschikbaar zijn, samen met de kaart in dit hoofdstuk, kunnen gebruikt worden om het spoorspringen te realiseren.

4.2 De buslogica.

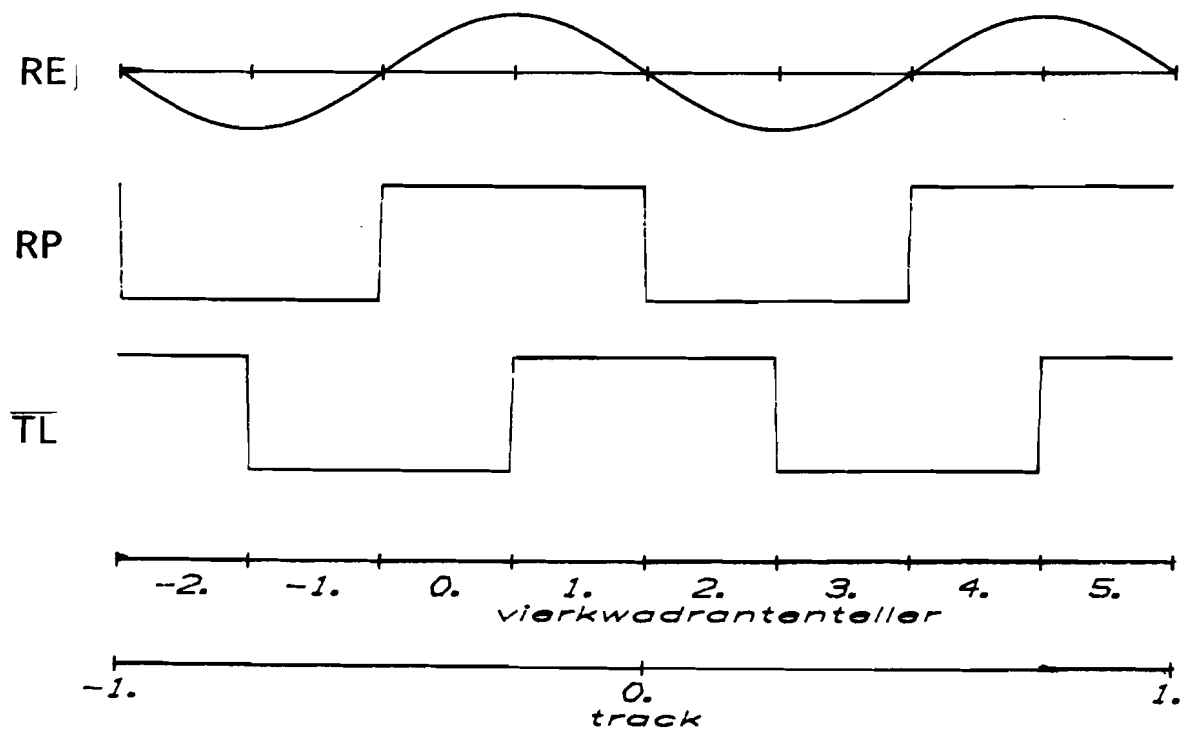
De buslogica verzorgt de communicatie met de bus, waarlangs met de 8051 of de PCB5010 gecommuniceerd kan worden. IC1 is de adresdecoder voor het lezen van van de kaart. In deze PROM staan maximaal acht adressen. De kaart kan informatie op de bus zetten als het BPRD-sigitaal laag is, en een adres op de kaart geselecteerd wordt. IC2 werkt hetzelfde, het is een adresdecoder in het geval van een schrijfactie, maar gebruikt dan het BPWR-sigitaal. IC3 en IC4 zorgen voor buffering van de BPIO-lijnen van de bus. IC5 als laatste buffert en invertteert de kloksignalen van de bus, zodat elke klok op de kaart zowel negatief als positief aanwezig is.

4.3 Vierkwadrantenteller.

De vierkwadrantenteller geeft de positie van de arm op de compact disc, ten opzichte van een in te stellen spoor.

De signalen RP en $\overline{\text{TL}}$ komen binnen in de schakeling van McGee, die deze signalen debouncet. Voor beschrijving van deze schakeling verwijs ik naar zijn verslag [3]. De telstanden als functie van de signalen RP en $\overline{\text{TL}}$ staan in figuur 4.2. Uit deze figuur blijkt dat de 2 LSB van de tellerstand eenvoudig te bepalen zijn met logische functies van $\overline{\text{TL}}$ en RP. Tevens is te zien, wanneer de hogere orde bits moeten tellen: als de twee LSB van 00 naar 11 gaan, naar beneden en als ze van 11 naar 00 gaan, naar boven. De klokpulsen die hierbij horen, worden gemaakt door IC13, een PAL. De inhoud van deze PAL is gegeven in appendix F.

IC7 synchroniseert de twee LSB met de andere telbits d2-d15.



Figuur 4.2: Telstand als functie van plaats spot op compact disc.

IC8-IC11 zijn de tellers van de positie: -up/down counters-. Het is mogelijk deze tellers te laden door te schrijven op het eerste adres van de kaart. Echter: het TL en RP signaal bepalen d0 en d1 en deze veranderen dus niet in het geval van een schrijffactie.

Twee tri-state-registers IC14 en IC15 zetten bij een leesactie op het eerste adres van de kaart de informatie van d0-d15 op de bus.

Als laatste zorgen de twee flipflops van IC12 voor het gesynchroniseerd inklokken van de tellerstand in IC14 en IC15.

4.4 Snelheidsmeting.

Het gedeelte snelheidsmeting op de random-access-kaart bepaalt de snelheid van de arm ten opzichte van de compact disc.

Meting van de periode van het RP-sigitaal en omzetting van deze tijd met behulp van een tabel in een PROM levert een getal dat evenredig is met de snelheid van de arm. Dit gebeurt als volgt. Allereerst synchroniseert IC16a het RP-sigitaal met de andere signalen in de schakeling. Dan wordt door de twee flipflops van IC17 van elke opgaande flank van het RS-sigitaal een puls gemaakt, die de periodetellers IC18-IC20 elke periode reset. Deze tellers tellen na die resetpuls met een frequentie van 10 MHz totdat er weer een resetpuls komt, waarop een nieuwe cyclus begint.

De twee registers IC21 en IC22 slaan de telstand van IC18-IC20 net voor het resetten op. Op de uitgangen hiervan staat nu een getal dat evenredig is met de periodetijd van het RP-sigitaal. Uitgaande van dit getal leveren de twee prom's IC27 en IC29 nu de snelheidsinformatie. De 8 MSB geven het aantal mm/s, en de LSB de fraktie ervan. Dit leidt tot een programmering van de PROM's als:

$$v = \frac{4,096 \cdot 10^6}{s + 2} \quad (4.1)$$

Met s de tellerstand, oftewel de periodetijd in $0,1 \mu\text{s}$, en v de bijbehorende snelheid in $1/256 \text{ mm/s}$. Hierbij is aangenomen, dat de spoed van het spoor $1,6 \mu\text{m}$ is. De 8

LSB van v zijn voor IC29, en de 8 MSB voor IC27. In de noemer staat s+2 omdat zoals uit de timing blijkt, de eerste twee telpulsen worden genegeerd.

IC28 en IC30 hebben dezelfde functie als IC14 en IC15 in het gedeelte van de vierkwadrantenteller: ze verzorgen de communicatie met de bus.

Er blijven nog een aantal IC's over: IC16b, IC23, IC24 en IC26. Zij maken de klokpuls voor de registers IC28 en IC30, en deze mag immers pas komen als de data op de uitgangen van de PROM's stabiel is, en dat duurt na de RS-flank nog minimaal 250 ns.

4.5 Positieteller.

Het laatste deel van de random-access-kaart is de positieteller. Deze bepaalt de positie van de arm op de compact disc met behulp van alleen het RP-signaal. De informatie die van deze teller is dus alleen geldig als zeker is dat de arm gedurende het tellen dezelfde richting houdt.

De positieteller gebruikt het gesynchroniseerde RP-signaal, RS van de snelheidsmeting. Vier tel-IC's, IC31-IC34, tellen het aantal opgaande flanken. Door een schrijfactie op het derde adres van de kaart kan de positieteller gereset worden.

IC36 en IC37 verzorgen de communicatie met de bus, zij werken hetzelfde als IC14 en IC15 in de vierkwadrantenteller.

Als laatste zorgen IC26b en IC35 er voor dat de registers IC36 en IC37 niet geklokt worden op het moment van een schrijfactie of als de tel-IC's tellen.

Hoofdstuk 5

Softwareimplementatie.

Het schrijven van de software kan beginnen nu de theorie van het spoorpringen is ontwikkeld en de hardware aanwezig is. Nu is het zo dat de programmatuur van het spoorpringen niet op zichzelf staat, maar dat de algehele structuur van de software de opzet ervan bepaalt. Daarom opent dit hoofdstuk met een beschrijving van de structuur van het gehele programma, waarbij de delen die belangrijk zijn voor het spoorpringen meer aandacht krijgen.

De samplefrequentie is 10 kHz, gerealiseerd door het geven van een interrupt om de 100 μ s. Het hoofdprogramma wordt bij elke interrupt gestopt, en de interruptroutine rekent de nieuwe regelacties voor de regelingen uit, uitgaande van de signalen die van de compactdisc-speler komen.

Samenvattend: de interruptroutine rekent de regelacties uit, uitgaande van coëfficiënten en variabelen die van het hoofdprogramma komen, en het hoofdprogramma doet de niet tijd-kritische zaken, zoals het uitrekenen van variabele versterkingen, setpoints voor regelingen, het uitrekenen van spronglengte's etc.

5.1 Het hoofdprogramma.

Het hoofdprogramma is een gesloten lus. Het functioneren ervan zal na de beschrijving van het flowdiagram (figuur G.1) duidelijk zijn.

In het flowdiagram zijn de volgende blokken aan te wijzen:

initialisatie Zoals gebruikelijk staat op de eerste plaats de **initialisatie**. Dit houdt in dat de speler wordt gereset en alle regelingen gedisabled.

seriele input De 8051 detecteert commando's van een toetsenbord en geeft ze via een seriele lijn door aan de PCB5010.

command translator De binnengekomen commando's veranderen de gewenste status van het systeem. Het gevolg hiervan is dat het systeem probeert de actuele status gelijk te maken aan deze gewenste status.

address read Dit blok leest de absolute tijdcode van de compact disc, en test haar op geldigheid. De output van dit blok is dus een tijdcode en een valid-bit(ADV).

player De playerroutine probeert de actuele status gelijk te maken aan de gewenste. Dit doet ze door de acties die daarvoor nodig zijn uit te voeren. Dit zijn: het aan- en uitschakelen van regelingen, het verrichten van random access, het veranderen van setpoints, en het opstarten van regelingen.

display Dit blok kan informatie die op een of andere manier in de signaalprocessor aanwezig is zichtbaar maken op een LED-display. Dit is gemakkelijk in verband met debuggen van de software.

Het is niet zinnig alle details van het programma te geven, alleen die informatie die nodig is bij het schrijven van bepaalde routines ten behoeve van het spoorpringen zal worden gegeven op het moment dat ze noodzakelijk is.

5.2 De playerroutine.

Zoals aangegeven is de playerroutine een onderdeel van het hoofdprogramma. Een flowdiagram geeft verduidelijking over de werking ervan. In het flowdiagram, figuur G.2, zijn een aantal blokken aan te wijzen, namelijk:

cstr=pstr dit blok vergelijkt de actuele status van de compactdisc-speler met de gewenste status. Als die niet gelijk zijn aan elkaar stellen de **focus**, **turntable**, of de **radialroutine**, afhankelijk van de gewenste verandering, de regelingen opnieuw in.

focus de focusroutine kan de focusregeling aan en uitzetten, en kan het focuspunt zoeken.

turntable de turnableroutine kan de aansturing van de motor aan en uitzetten, en het toerental ervan regelen met een tacho- of een mces-regeling.

radial de radialroutine kan de besturing van de arm aan- en uitzetten, en kan de besturing ervan opstarten door naar het midden van de compact disc te gaan, in de lead-in te springen en na het lezen ervan in het datagebied te springen.

calc de calcroutine kan in de eerste plaats de spronglengte bepalen, in geval van een sprong naar een bepaalde tijdcode, en ten tweede welke hoeksnelheid de motor moet hebben als de arm aankomt op het gewenste spoor.

jumproutine de jumproutine verzorgt het spoorpringen. Ze krijgt hiervoor het aantal sporen en de richting binnen.

De onderdelen van de playerroutine die voor het spoorpringen van belang zijn, zijn **calc** en **jumproutine**.

5.3 De calcroutine.

Het flowdiagram van de calcroutine is gegeven in figuur G.3. De werking hiervan is als volgt:

Als eerste controleert de routine of het adres, dat van de plaat is gelezen, geldig is -**ADV=1**-, als dat niet het geval is dan is de calcroutine afgelopen omdat die dan niets uit kan rekenen.

Als niet gezocht moet worden naar een bepaald adres -**CS=0**-, en niet op een bepaald adres in pauze worden gewacht -**CPS=0**-, dan is de calcroutine klaar.

Nu volgt een test op het adres dat gelezen is van de plaat (**psec.pfram**). Is dit adres hetzelfde als het gewenste adres (**csec.cfram**), dan wordt **CS=0** gemaakt, wat betekent dat de sprong klaar is, en data kan worden vrijgegeven. Als de speler in pauze staat, **CPS=1**, springt de arm nu terug naar het vorige spoor. De speler wacht nu net voor het gewenste adres.

Is dit adres niet hetzelfde, dan wordt 'calcul' aangeroepen. Na het uitvoeren ervan bevat geheugenlocatie 'cjump' de spronglengte en afhankelijk van de richting geldt $crv=1$ (achterwaarts) of $cfw=1$ (voorwaarts). Ook berekent hij de hoeksnelheid van de motor na een sprong. De motorregeling krijgt dan het commando de plaat met die hoeksnelheid te laten draaien.

5.3.1 De calculroutine.

Het flowdiagram van de calculroutine staat in figuur G.4. Deze routine maakt gebruik van twee functies. De eerste wordt in het flowdiagram aangegeven door f, en berekent het spoor waarop een bepaalde tijdcode zich bevindt. De tweede wordt aangegeven door g, en berekent de hoeksnelheid van de motor behorend bij een bepaald spoor.

Voor een nadere bespreking van het flowdiagram volgt eerst een beschrijving van de functies.

Het verband tussen tijdcode en spoor.

Het verband tussen t en N is als volgt worden opgeschreven (zie appendix C):

$$t = \frac{2 \cdot \pi \cdot r_0 \cdot N}{v} + \frac{\pi \cdot s \cdot N^2}{v} \quad (5.1)$$

Als s en v bekend zijn kan N uitgerekend worden als t bekend is, met de formule:

$$N = \frac{\sqrt{(2 \cdot \pi \cdot r_0)^2 + 4 \cdot \pi \cdot v \cdot s \cdot t} - 2 \cdot \pi \cdot r_0}{2 \cdot \pi \cdot s} \quad (5.2)$$

De PCB5010 kan echter niet worteltrekken, en gebruikt dus een benadering. Hiervoor gebruik ik het Newton-Raphson algoritme oplossen van het probleem $f(N) = 0$:

$$N_{n+1} = N_n - \frac{f(N_n)}{\frac{df(N_n)}{dN_n}} \quad (5.3)$$

Noem:

$$f(N) = 2 \cdot \pi \cdot r_0 \cdot N + \pi \cdot s \cdot N^2 - v \cdot t \quad (5.4)$$

dan:

$$N_{n+1} = N_n - \frac{\pi \cdot r_0 \cdot N_n + \pi \cdot s \cdot N_n^2 - v \cdot t}{2 \cdot \pi \cdot r_0 + 2 \cdot \pi \cdot s \cdot N_n} \quad (5.5)$$

Het programma is gegeven in appendix H. Het programma gebruikt als input de tijd gegeven in seconden en aantal frames. De benadering vindt plaats binnen vijf iteraties en is dan op 2 sporen nauwkeurig, mits s en v nauwkeurig genoeg bekend zijn. De uitkomst is monotoon stijgend en heeft een deadband kleiner dan plus of min een tiende spoor. Verkleining van woordlengtes heeft tot gevolg dat de performance qua nauwkeurigheid en deadband snel afneemt.

Berekening hoeksnelheid van motor.

De hoeksnelheid van de motor is gerelateerd aan het spoor waarop de arm zich bevindt met de volgende vergelijking:

$$\omega = \frac{v}{N \cdot s + r_0} \quad (5.6)$$

Hetgeen overeenkomt met de lineaire snelheid gedeeld door de straal. Het programma is gegeven in appendix H.

Nu kan 'calcul' worden besproken. De routine begint met een test $CS=1$. Als dit zo is, dan wordt om een sprong gevraagd, en volgt uitvoering van het programma.

N_1 is het spoornummer van de gewenste tijdcode en N_2 het spoornummer van de actuele tijdcode. De absolute waarde van het verschil tussen deze twee getallen is de spronggrootte en wordt in 'cjump' opgeslagen. De sprong is afhankelijk van het teken van dit getal wordt naar voren, $cfw=1$, of naar achteren, $crv=1$.

Dan als laatste volgt berekening van de hoeksnelheid van de motor op de plaats van bestemming en de motor krijgt het commando die snelheid aan te nemen.

5.4 De jumproutine.

De jumproutine wordt aangeroepen door statusbitjes te zetten en de gewenste spronglengte in een register te zetten. De uitvoering van de sprong kent een aantal fases: de start, de snelheidsregeling, het vertragen, en het invangen. Als de tijdcode is gevonden wordt $CS=0$.

De start van de sprong. Met behulp van de positierегeling start de sprong. De positierегeling krijgt een setpoint ver van de actuele positie en zal de arm in die richting gaan versnellen.

De snelheidsregeling. De snelheidsregeling schakelt in zodra 10 sporen zijn overgestoken. De snelheidsregeling stuurt de arm over de sporen met een zo hoog mogelijke snelheid.

Het vertragen. Het vertragen begint zodra het overstapcriterium is bereikt. Dit gebeurt eenvoudig door het veranderen van het setpoint van de snelheidsregeling naar 20 mm/s.

Het invangen. Het invangen begint wanneer het gewenste spoor op 10 sporen is genaderd. Deze afstand volgt uit de tellerstand van de positieteller. De positierегeling vangt de arm in op het juiste spoor. Dit gebeurt in 10 sporen. Na beëindiging hiervan neemt het hoofdprogramma, na het goed zetten van de status, de controle weer over.

Wanneer een sprong kleiner is dan 20 sporen (N_{ref}) wordt meteen begonnen met positierегeling, omdat dit dan de snelst mogelijke manier van springen is.

5.5 De interruptroutine.

De interruptroutine rekest de regelacties uit. Daarvoor leest die eerst de nodige informatie in, en daarna rekest hij met behulp van een PID-routine het regelsignaal uit.

De interruptroutine rekest alle regelingen uit, maar voor het spoorspringen is de radiele van belang en dit verslag behandelt dan ook die regeling. Het berekenen van het regelsignaal kan op twee manieren gebeuren. In de eerste plaats is er de positierегeling. Hiervoor leest de PID-routine het re-signaal en de telstand van de vierkwadrantenteller in, en berekest de positie. Dan wordt het regelsignaal uitgerekend met het algoritme van een verlengde s-curve. In de tweede plaats is er de snelheidsregeling. In dit geval leest de routine de snelheid in.

De regelaar gebruikt een PID-structuur. Uit de manier waarop tot nu toe de filters uitgerekend zijn zou een cascaderstructuur van een lead- en een lagfilter te verwachten zijn. Een PID-filter heeft echter minder vertragingselementen -slechts twee- en voert dezelfde

funkties uit. Omdat ook voor een PID-filter is gekozen voor de andere regelingen, is die voor de radiale regeling ook ingevoerd.

De algehele structuur van een PID-regelaar is schematisch weergegeven in figuur 5.1.

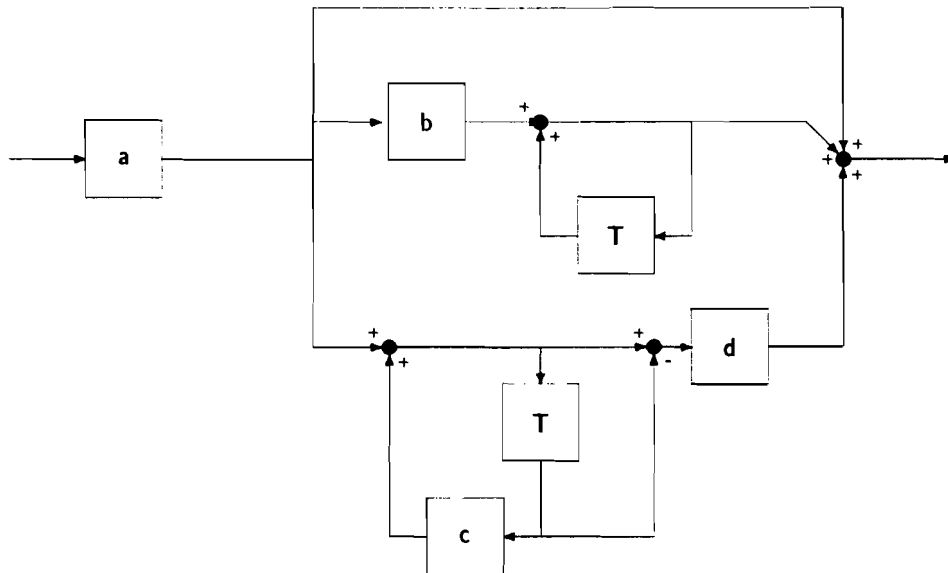
Hetingangssignaal is het verschil tussen de gewenste en actuele waarde van de regelgrootheid, het uitgangssignaal is het stuursignaal. De blokken a, b, c, en d zijn vermenigvuldigingen en de blokken T vertragen huningangssignaal een sample. De bovenste sectie vertegenwoordigt het P-stuk, de middelste het integrerende I-stuk en de onderste het differentierende D-stuk.

De overdracht van de PID-regelaar kan vanuit figuur 5.1 opgeschreven worden als:

$$H(z) = a \left(1 + \frac{b}{1-z^{-1}} + d \frac{1-z^{-1}}{1-cz^{-1}} \right) \quad (5.7)$$

Of:

$$H(z) = a \frac{(1+b+d) - (1+c+bc+2d)z^{-1} + (c+d)z^{-2}}{(1-z^{-1})(1-cz^{-1})} \quad (5.8)$$



Figuur 5.1: PID-regelaar.

De filters die uitgerekend zijn als serieschakeling van een lead- en een lagfilter hebben cascadevorm en zijn dus van de vorm:

$$H(z) = k \frac{(e - f.z^{-1})(g - h.z^{-1})}{(1 - z^{-1})(1 - i.z^{-1})} \quad (5.9)$$

Of:

$$H(z) = \frac{p - q.z^{-1} + r.z^{-2}}{(1 - z^{-1})(1 - s.z^{-1})} \quad (5.10)$$

Met:

$$p = k.e.g \quad (5.11)$$

$$q = k.(e.h + f.g) \quad (5.12)$$

$$r = k.f.h \quad (5.13)$$

$$s = i \quad (5.14)$$

Om dus de cascaderstructuur van formule 5.9 om te schrijven naar PID-structuur van formule 5.7 moeten zij aan elkaar gelijk zijn en geldt:

$$p = a.(1 + b + d) \quad (5.15)$$

$$q = a.(1 + c + b.c + 2.d) \quad (5.16)$$

$$r = a.(c + d) \quad (5.17)$$

$$s = c \quad (5.18)$$

Dit levert de volgende vergelijkingen voor a, b, c, en d waaruit ze gemakkelijk kunnen worden opgelost als k, e, f, g, h, en i bekend zijn:

$$a = \frac{q - 2.r + r.s - p.s}{s^2 - 2.s + 1} \quad (5.19)$$

$$b = \frac{p - r + a.s - a}{a} \quad (5.20)$$

$$c = s \quad (5.21)$$

$$d = \frac{r - a.s}{a} \quad (5.22)$$

De constanten a, b, c, en d staan in ROM. De routine krijgt bij aanroep het adres van deze constanten en het ingangssignaal voor de PID-sectie. Dit ingangssignaal mag maximaal 14 bits breed zijn. Het resultaat van de berekening van de PID-routine is de regelactie.

Het uitgangssignaal wordt door middel van een D/A-omzetter omgezet in stuurspanningen.

Hoofdstuk 6

Test van spoorspringprocedure.

Na de ontwikkeling en implementatie van de spoorspringprocedere volgt de evaluatie ervan. Deze evaluatie geschiedt door het doen van metingen aan het proces, en deze te vergelijken met de verwachting zoals beschreven in voorgaande hoofdstukken.

De opzet van het hoofdstuk is als volgt. Er zijn verschillende delen van het spoorspringen die getest moeten worden. Verschillende experimenten evalueren die delen. Elk experiment bevat een bespreking van het doel, het resultaat, en de conclusie. Eventueel volgt op zo'n conclusie weer een experiment. Het is de bedoeling dat de serie experimenten die zo ontstaat, optimale regelingen ontstaan.

6.1 De testopstelling.

De testopstelling die gebruikt is voor de metingen ziet is als volgt gerealiseerd:

- Van de optica komt het RE-sigitaal bij de A/D-omzetter. Deze converteert signalen met een amplitudo binnen de plus en min vijf volt naar een getal met een breedte van 10 bits. Dit betekent dus een resolutie van 10 mV.
- De generatie van het positie-sigitaal gebeurt met de zeven meest significante bits van het digitale RE-sigitaal. Dit resulteert in een resolutie van 124 stappen per spoor, oftewel 13 nm per LSB.
- De sigitaalprocessor heeft de beschikking over de signalen die komen van de hardware die is beschreven in hoofdstuk 4.
- De instelling van de coëfficiënten a, b, c, en d wordt bij elk experiment vermeld.
- De pid-sectie voert het uitgangssigitaal zonder normering aan de D/A-omzetter toe.
- De D/A-omzetter heeft een ingang met een breedte van 12 bits en de uitsturing ervan is van min naar plus vijf volt.
- De uitgang van de D/A-omzetter bepaalt de uitsturing van de stroombron. Deze stroombron heeft een overdracht van 0.2 A/V. De maximale stroom die geleverd kan worden is dus 1 A.
- De stroombron levert direct aan de radiale actuator. Het loopwerk is van het type CDM-1.
- De elektronica in de CM100 regelt tijdens de experimenten de motor en de focussering.

De software van de jump routine is gegeven in appendix H, evenals het programmagedeelte in de interrupt routine dat de regelactie voor de arm voor zijn rekening neemt.

6.2 Radiele regeling.

6.2.1 Meting overdracht radiele regeling.

Meting van de overdracht van het openloop-systeem regelaar-arm van de regeling zoals aanwezig in de CM100. Het doel hiervan is de overdrachten van de digitale regelingen hiermee later te kunnen vergelijken.

Experiment.

Een spectrumanalyser bepaalt de openloop-overdracht van het systeem. Deze bepaalt hij door het optellen van een sinus van 300 mV bij het RE-sigitaal en uit de reductie van deze storing de overdracht te berekenen.

Waarneming.

De overdracht van het openloop-systeem is gegeven in figuur I.1. De fasemarge is 45 graden bij een bandbreedte van 500 Hz. Het systeem vertoont een aantal oscillaties bij frequenties boven 1 kHz.

6.2.2 Evaluatie berekende regeling.

Meting van de overdracht van de radiele regeling die gevonden is in hoofdstuk 3. Vergelijking van meting met verwachte overdracht.

Experiment.

Overdracht-meting analoog aan vorig experiment.
(a=00c0H, b=0002H, c=ffd8H, d=0ec2H)

Waarneming.

De overdracht van het open loop-systeem met de regelaar is gegeven in figuur I.2. De fasemarge van de regelaar is 36 graden bij een bandbreedte van 500 Hz. De maximale fase die optreedt is 45 graden bij 300 Hz. De gesimuleerde regeling heeft een fasemarge van 45 graden, terwijl de minimaal toelaatbare fasemarge 40 graden is.

Conclusie.

De regeling is niet bruikbaar voor spoorvolgen of invangen omdat de fasemarge te laag is. De fasemarge van de regeling moet vergroot worden tot minimaal 40 graden. De top van deze fase ligt bij 300 Hz en is 45 graden. Het systeem zou dus verbeterd kunnen worden door deze top te verschuiven naar een hogere frequentie, bijvoorbeeld door het verschuiven van het lead-filter.

6.2.3 Evaluatie geoptimaliseerde volgregeling.

Uitgaande van de conclusie van het vorige experiment wordt een nieuw leadfilter uitgerekend dat begint bij 150 Hz, en doorloopt tot 2,5 kHz. Het lagfilter blijft gelijk aan de berekende versie.

Experiment.

Overdracht-meting analoog aan vorig experiment.

(a=0130H, b=0004H, c=ff01H, d=10ffH)

Waarneming.

De overdracht van dit openloop-systeem is weergegeven in figuur 1.3. Uit deze figuur blijkt dat het systeem een fasemarge heeft van 42 graden, bij een bandbreedte van 500 Hz. De overdracht van het openloop-systeem lijkt tot 1 kHz zeer op de overdracht van het analoge systeem. Boven deze frequentie worden de effecten van bemonsteren zichtbaar. De fase neemt snel af, en de gain vertoont niet zo'n sterke afval als in het analoge geval. De verschillen in het hoge frequentiegebied met de analoge regeling hebben tot gevolg dat tijdens bedrijf geruis hoorbaar is, omdat deze frequenties minder onderdrukt worden.

Conclusie.

Deze regeling is bruikbaar voor radiale regeling van de arm tijdens volgen. Wanneer de bemonsteringsfrequentie toeneemt, wat in de toekomst het geval zal zijn, vermindert het geruis omdat dit de mogelijkheid biedt een beter filter te realiseren. Tevens zou vervanging van de stroombron door pulsbreedte-aansturing dit effect kunnen hebben.

6.2.4 Test van berekende regeling tijdens spoorspringen.

Evaluatie van de berekende radiale regeling bij een stapvormige excitatie.

Experiment.

Een programma in de PCB5010 laadt de vierkwadrantenteller met een waarde die overeenkomt met een afstand van 18 sporen. De regelaar probeert het foutsignaal dat nu wordt gemeten te reduceren en een stapresponsie is het gevolg. Dit experiment bekijkt het RE-signaal en het berekende foutsignaal en probeert een uitspraak te doen over de kwaliteit van de regeling voor spoorspringen.

(a=00c0H, b=0002H, c=ffd8H, d=0ec2H)

Waarneming.

Het resultaat van een sprong van 18 sporen vooruit is gegeven in figuur 1.4. De overshoot is op te maken uit de figuren en bedraagt 3 sporen. De reactietijd is ongeveer 20 ms, hetgeen redelijk overeenkomt met de simulatie in figuur 3.6. Er is nog iets te zien: Het gewenste spoor wordt langzaam benaderd. Tijdens deze nadering blijkt de arm steeds even hangen als het TL-signaal omklapt, oftewel als de s-curve verlengd wordt.

Conclusie.

De regeling is niet optimaal voor sporspringen. De overshoot is veel groter dan in de simulatie. Ook is de langzame nadering van het spoor is ongewenst, de oorzaak hiervan is het lagfilter (de integrator vindt langzaam zijn waarde). Deze eigenschappen horen echter bij de berekende regeling en zijn zelfs nog sterker aanwezig bij de geoptimaliseerde versie. De conclusie is dat de berekende regeling niet geschikt is voor sporspringen.

De tweede conclusie moet zijn dat de s-curve niet goed is verlengd. Er zitten te grote discontinuïteiten in. Dit leidt tot oscillaties tijdens het regelen naar het setpoint. De verlenging van de s-curve moet beter aansluiten. Correctie van dit euvel leidt in het vervolg tot betere metingen.

6.2.5 Test van geoptimaliseerde regeling voor sporspringen.

De conclusie van het vorige experiment leidt tot het gebruiken van een andere regeling tijdens de sprong, dan tijdens het volgen van het spoor. Deze regeling heeft geen lagfilter, omdat dit leidt tot het langzame naderen van het spoor, en reductie van de excentriciteit van meer dan 60 dB niet van belang is voor het sporspringen. Het leadfilter loopt van 70 Hz tot 2,2 kHz.

Experiment.

Meting overdracht van regeling geoptimaliseerd voor sporspringen, analoog voorgaande overdrachtmetingen.

($a=00c0H$, $b=0000H$, $c=0020H$, $d=14c2H$)

Waarneming.

De overdracht van de regeling is gegeven in figuur 1.5. Uit de figuur blijkt dat de overdracht van het systeem een fasemarge van 50 graden heeft bij een bandbreedte van 400 Hz.

Conclusie.

De regeling is stabiel, ze heeft een fasemarge ruim groter dan geeist.

Experiment.

Evaluatie van het afremgedrag na een sprong van 200 sporen. De arm heeft dan een gedefinieerde snelheid van 16 mm/s en de vierkwadrantenteller is gepreset op 10 sporen afstand tot het gewenste spoor. Een regeling die geoptimaliseerd is voor sprongen remt de arm dan af. Dit laatste gedeelte van de sprong wordt geevalueerd.

($a=00c0H$, $b=0000H$, $c=0020H$, $d=14c2H$)

Waarneming.

De responsie tijdens het afremmen is weergegeven in figuur 1.6. De reactietijd is nu veel beter: 3 ms. De overshoot is eveneens sterk gereduceerd. Wel blijkt dat een sprong naar binnen een andere responsie heeft dan een naar buiten, bij een binnenwaartse sprong treedt een kleine overshoot op, in het andere geval geen. Uit verschillende waarnemingen blijkt dat de arm binnen 30 ms altijd het spoor weer keurig volgt.

Conclusie.

De regeling is snel en heeft weinig overshoot. Het verschil in gedrag voor een binnen- en buitenwaartse sprong is niet zodanig groot dat de regeling ervoor afgekeurd hoeft te worden. Samen met de constatering dat de regeling stabiel is, is de conclusie dat deze regeling bruikbaar is voor afremmen.

6.2.6 Evalueren geoptimaliseerde sprongregeling.

Nu er twee verschillende regelingen zijn voor de radiale volging, vindt er bij het afremmen een overschakeling plaats van de ene regeling naar de andere. Dit gebeurt 5 ms na het bereiken van het gewenste spoor (als het \overline{TL} -signaal hoog wordt van dat spoor).

Experiment.

Meting van het afremgedrag analoog aan vorige meting.

Waarneming.

Het afremgedrag is weergegeven in figuur 1.7. De sprong heeft weinig overshoot, de arm is binnen 30 ms ingevangen. Uit het vergelijken van een aantal metingen blijkt dat de reproduceerbaarheid zeer groot is.

Conclusie.

De sporsprongregeling zoals die nu gevonden is, is zeer goed bruikbaar. Enerzijds is de regeling tijdens de sprong zeer snel, veel sneller dan de gesimuleerde sprong, anderzijds is de regeling tijdens spoorvolgen zeer accuraat. Beide regelingen hebben voldoende fasemarge, en vertonen geen oscillaties.

6.3 De spronglengte en sprongtijd.

6.3.1 Sprongnauwkeurigheid bij sprong kleiner dan N_{ref} .

Evaluatie van het gedrag van de regeling bij een sprong kleiner dan N_{ref} . Een sprong van deze lengte vindt volledig plaats in de positierregeling, met behulp van de verlengde s-curve.

Experiment.

Uitvoeren van een sprong met een lengte kleiner dan N_{ref} , namelijk 18 sporen. Meting van het RE-sigitaal en het berekende foutsigitaal tijdens de sprong.

Waarneming.

Een waarneming van deze sprong is gegeven in figuur 1.8. De sprong heeft precies de goede lengte en heeft weinig overshoot. De arm is weer in volging na 10 ms. Bij herhaling van de sprong blijkt dat elke sprong precies hetzelfde beeld geeft, de reproduceerbaarheid is zeer groot.

Conclusie.

De regeling is geschikt voor het doen van sprongen van een lengte kleiner dan N_{ref} .

6.3.2 Sprong bij een lengte groter dan N_{ref} .

Evaluatie van een sprong bij een lengte groter dan N_{ref} . Tijdens deze sprong schakelt het programma van positieregeling naar snelheidsregeling en weer terug. Tegelijkertijd vindt er dan omschakeling van positieteller plaats. Controle van de sporentelling.

Experiment.

Uitvoeren van een sprong met een lengte groter dan N_{ref} , namelijk 30 sporen. Meting van het RE-sigitaal en het berekende foutsigitaal tijdens de sprong.

Waarneming.

Een waarneming van deze sprong is gegeven in figuur 1.9. De lengte van de sprong is meestal een spoor te groot. Soms is ze exact. Dit lijkt te liggen aan het moment waarop de snelle positieteller wordt gereset. Dit gebeurt niet precies op het goede moment, en daardoor ontstaat een telfout. Verder is de sprong reproduceerbaar en vertoont nauwelijks overshoot.

Conclusie.

De sprongregeling is bruikbaar voor sprongen groter dan N_{ref} , met dien verstande dat een verbetering in accuratesse mogelijk moet zijn. Er wordt normaliter niet meer dan een spoor te veel gesprongen, wat al een significante verbetering is ten opzichte van de routine in de CM100.

6.3.3 Relatie spronglengte en sprongtijd.

Het is interessant om de accesstijd te weten bij een bepaalde spronglengte. Hierbij moet worden vermeld dat de sprongsnelheid tijdens de sprong nu nog maximaal 50 mm/s is, omdat de elektronica niet is berekend op hogere snelheden. In principe moet 80 mm/s haalbaar zijn.

Experiment.

Evaluatie van sprongen van verschillende lengte. Meting van de tijd van de start van de sprong tot invangen op het gewenste spoor -tot het moment dat TL hoog wordt- (dit is niet de accesstijd).

Waarneming.

De tabel met metingen is gegeven in tabel 1.1. Naast deze metingen is de tijd per spoor weergegeven. De tijd per spoor daalt per spoor en nadert asymptotisch naar 33 μ s per spoor: 50 mm/s. Met de gerealiseerde sprongtijden kan random access bij een sprong over een hele plaat binnen de seconde blijven, en een sprong over een derde van de plaat binnen een halve seconde, mits de motor snel genoeg op toeren is. Wanneer de snelheid wordt opgevoerd naar 80 mm/s wordt het geheel natuurlijk nog sneller.

Conclusie.

De regeling kan qua sprongtijden mee met de nieuwste ontwikkelingen.

Hoofdstuk 7

Conclusies en aanbevelingen.

De groep 'digitale servo' van Philips werkt aan het digitaliseren van de verschillende regelingen in een compactdisc-speler. Hiervoor is een ontwikkelsysteem gebouwd met een digitale signaalprocessor van het type PCB5010. Binnen dit projekt is gewerkt aan digitale implementatie van random access: het uitvoeren van een sprong naar een bepaalde tijdcode op een compact disc. Hierin kunnen twee onderdelen worden onderscheiden: het uitrekenen van de spronggrootte en het spooorspringen zelf.

Voor het spooorspringen zijn radiele regelingen ontworpen voor snelheids- en positieregeling. Met deze regelingen zijn snelheidsprofielen voor de sprong gerealiseerd. Daarnaast is een algoritme gegeven voor het uitrekenen van de spronggrootte, wanneer er naar een bepaalde tijdcode gesprongen moet worden.

Het spooorspringen is geëvalueerd door vergelijking met de bestaande implementatie, nieuwe specificaties, en verwachtingen uit simulaties, berekeningen en beredeneringen.

Regelingen.

Er zijn verschillende regelingen gerealiseerd voor snelheids- en positieregeling. Elke regeling is geoptimaliseerd voor zijn specifieke toepassing. De fasemarge van alle regelingen is groter dan 40 graden, zodat stabiliteit gegarandeerd is.

De regelingen krijgen informatie over de positie of snelheid van de arm via speciale hardware en software. Een algoritme, de zogenaamde verlengde s-curve, geeft de positieinformatie. Een periodetijdmeting geeft via een tabel in een PROM de snelheidsinformatie.

De snelheidsregeling onderscheidt zich van de oude implementatie door het feit dat hij nu is geïmplementeerd als digitale regeling met tijduniforme bemonstering en een lineair systeem.

De positieregeling heeft een overdracht die sterk lijkt op die van de analoge versie. Het grootste verschil zit in de frequenties boven de 1 kHz. De reductie van de digitale regeling is daar minder. Dit heeft tot effect dat tijdens het spoorvolgen een geruis hoorbaar is. Realisatie van filters met meer reductie in de hoge frequenties is mogelijk bij verhogen van de samplefrequentie, wat inderdaad zal gebeuren, en het toepassen van pulsbreedte-modulatie voor de stroomsturing.

Het RE, RP en *overline*TL-sigitaal leveren alle regelsignalen. Gezien vele problemen met de verwerking van die signalen moet worden aanbevolen een onderzoek te doen naar de generatie ervan, en van de signalen die er van afgeleid worden. Hierbij kan bijvoorbeeld gedacht worden aan een systemschatter die de snelheid en de positie van de arm ten opzichte van de compact disc schat, uitgaande van de sturing van de arm, de dynamische

eigenschappen ervan, en de signalen van de fotodiodes.

De strategie van het verlengen van de s-curve heeft een bijzonder goed effect op de positierегeling. Deze is nu in staat veel grotere verstoringen op te vangen. Wel brengt gebruik ervan specifieke problemen met zich mee. De s-curve sluit nooit exact aan doordat de amplitudo van het RE-sigitaal niet constant is. Dit veroorzaakt een discontinuiteit in het regelsigitaal die door de differentiator wordt getransformeerd in een spike op de sturing van de arm. Dit kan leiden tot kleine oscillaties in het regelgedrag. Bij goede tuning van de parameters kan dit voorkomen worden.

Accuratesse.

Het is in principe mogelijk sprongen met een gegeven lengte exact uit te voeren. In de huidige implementatie is dit nog niet het geval, omdat het overschakelen van de ene positieteller op de andere nog niet geheel vlekkeloos verloopt. Hierdoor kan het zijn dat bij een grote sprong een fout van twee sporen wordt gemaakt. Dit is echter verwaarloosbaar ten opzichte van de fout die in het oude systeem altijd optrad: er werden daar altijd minimaal 15 sporen gemist. Bij nauwkeurige tuning moet zelfs deze fout voorkomen kunnen worden.

Bij sprongen naar een gegeven tijdcode moet een significante verbetering in accuratesse mogelijk zijn, doordat de berekening van de spronglengte uitgaande van de tijdcode waarop de arm zich bevindt veel beter is dan de tot nu toe gebruikte. Als gevolg hiervan traden fouten tot 1000 sporen op.

De accuratesse van de sprong is in mede afhankelijk van de kwaliteit van het RP-sigitaal. Bij toenemende snelheid van de arm neemt die af. Hierdoor worden dan te veel sporen gesprongen. Ook dit pleit voor een onderzoek naar de opwekking van dit sigitaal.

Accesstijd.

Met de sprongtijden zijn accesstijden van een seconde voor de hele plaat en een halve seconde voor een derde van de plaat realiseerbaar. Verhoging van de accuratesse van de sprong en de hogere snelheid tijdens de sprong leveren de grootste bijdrage aan de accesstijdverbetering. De accesstijden zijn in alle gevallen beter dan die van de CM100.

Digitaal regelen als vervanger van analoog regelen.

De ervaringen bij de implementatie van het spoorpringen met de digitale testopstelling zijn positief te noemen. Het is mogelijk regelingen te ontwerpen die voldoen aan een breed scala van eisen. Door de mogelijkheid van het schakelen van regelingen kan onder elke omstandigheid de optimale regeling gekozen worden. Het resultaat is dus een flexibeler opzet, die gemakkelijker aan meer eisen kan voldoen dan een analoge uitvoering. Daarnaast biedt de testopstelling de gelegenheid tot het doen van metingen. De sigitaalprocessor kan via een D/A-omzetter elk willekeurig sigitaal beschikbaar stellen voor meting en triggering. Ook is het nu mogelijk complexe berekeningen uit te voeren door de aanwezigheid van de sigitaalprocessor.

Eindconclusie.

Het is zeer goed mogelijk een procedure te maken voor random access binnen het concept 'digitale servo' die voldoet aan de eisen die daar op dit moment aan gesteld worden. De

testopstelling biedt ten opzichte van de oude implementatie de volgende voordelen:

- gemakkelijker integreerbaar,
- flexibeler in het gebruik van meerdere regelingen en
- er is een krachtige processor aanwezig waardoor het uitvoeren van complexe berekeningen, zoals het uitrekenen van spronglengte, mogelijk is.

De procedure die binnen dit concept gerealiseerd is, heeft de volgende kenmerken:

- zeer accuraat en daardoor snel,
- regelingen zijn goed gedefinieerd en stabiel,
- access voor de hele plaat kan binnen de seconde, en voor een derde van de plaat binnen een halve seconde geschieden.

Er zijn suggesties gedaan voor verder onderzoek aan de procedure. Deze suggesties hebben vooral betrekking op zaken die tijdens het afstudeerwerk zijn blijven liggen.

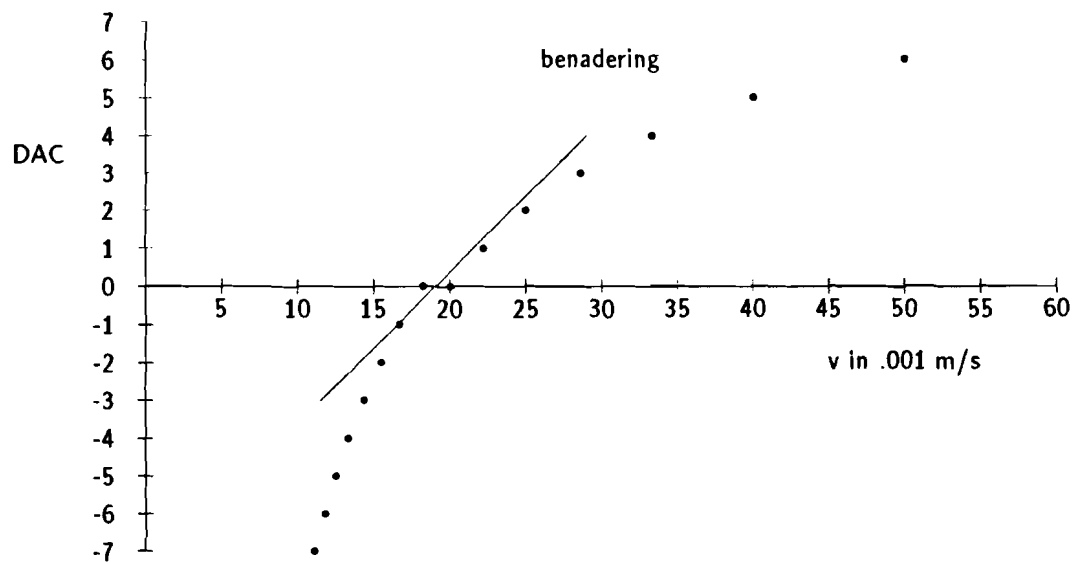
Referenties

- [1] Riemersma, M. *Random access tijd CD-ROM*. Stageverslag HTS Eindhoven, maart 1985.
- [2] Geelen, G. *Onderzoek aan de leesversterker van de servo-regellus van de compact disc speler*. Afstudeerverslag TH Eindhoven, 1983.
- [3] McGee, P. *A fast trackloss recovery strategy for compact disc*. Afstudeerverslag TU Eindhoven, mei 1987.
- [4] Van Doren, E.D.M. *Radiale en verticale volgsystemen voor een nieuwe generatie compact disc spelers*. Afstudeerverslag HTS Eindhoven, juni 1986.
- [5] Bouwhuis, G., J. Braat, A. Huijser, J. Pasmaan, G. van Rosmalen, K. Schouhamer Immink *Principles of optical disc systems*. Adam Hilger Ltd, Bristol, USA, 1985.
- [6] Service Consumer Electronics Philips *CD ROM player CM100 service manual*, 4822 725 20626.
- [7] Kuo, B.C. *Digital control systems*. Holt-Saunders International Editions, Japan, 1980.
- [8] Soto, J.C., M. de la Sen *Non-linear oscillations in nonperiodic sampling systems*. Electr. Lett. (GB), vol.30, no.20, p816-8, 27 sept 1984.
- [9] De la Sen, M. *On the input output modelling of nonperiodic-sampling systems*. Proc. of the 1984 American Control Conference (IEEE Cat.No.84CH2024-8) San Diego, CA, USA, 6-8 june 1984, (New York USA: IEEE 1984) p862-5, vol.2.
- [10] De la Sen, M. *A time-varying difference equation for nonperiodic sampling systems*. Proc. IEEE (USA), vol.72, no.4, p537-9, april 1984.
- [11] De la Sen, M. *A new modelling for aperiodic systems*. Int.J.Sci.(GB), vol.15, no.3, p315-328, march 1984.
- [12] Koenraads, A. *Documentation for the interactive simulation program picosim on VAX/VMS oper. system*. Ned. Philips Bedrijven, 4 jan 1984.
- [13] Electronic components and materials Philips. *Single-chip digital signal processor, PCB5010, PCB5011.*, 9397 102 30142, feb 1986.
- [14] Van den Enden, A., A. Verhoeckx *Tijddiscrete signaalbewerking. Basistheorie*. Ned. Philips Bedrijven, jan 1984.
- [15] Rabiner, L.R., B. Gold *Theory and application of digital signal processing*. Prentice-Hall inc., Englewood Cliffs, New Jersey, 1975.

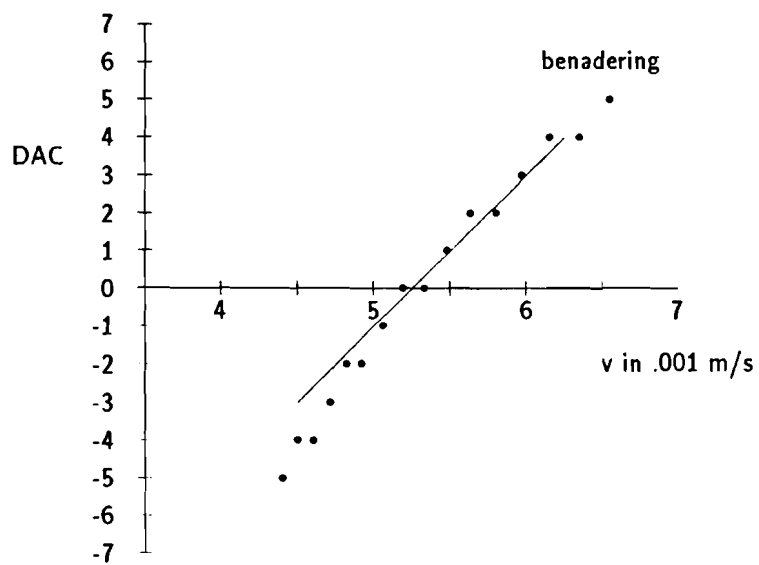
- [16] Parker, S.R., S.F. Hess *Limit cycle oscillations in digital filters*. IEEE trans. circ. theory, CT18, p687-96, november 1971.
- [17] Sandberg, I.W. *A theorem concerning limit cycles in digital filters*. Proc. 7th Allerton conf. on circuit and system theory, p63-7, 1969.
- [18] Brubaker, T.A., J.N. Gowdy *Limit cycles in digital filters*. IEEE trans. on automatic control, vol. 17, no. 5, p675-7, oktober 1972.
- [19] Jackson L.B. *An analysis of limit cycles due to multiplication rounding in recursive digital (sub)filters*. Proc. 7th annual Allerton conf. on circuit and system theory, p69-78, 1969.
- [20] Athans, A., P.L. Falb *Optimal Control*. McGraw-Hill inc., New York, 1966.
- [21] Feng, D., B.H. Krogh *Acceleration-constrained time-optimal control in n dimensions*. IEEE trans. on automatic control, vol. AC31, no. 10, p955-8, oktober 1986.
- [22] Canon, D.C., C.D. Cullum, E. Polak *Theory of optimal control and mathematical programming*. McGraw-Hill inc., New York, 1970.

Appendix A

Verband snelheid arm en
[DAC2,DAC1,DAC0].



Figuur A.1: Overdracht bij snelle sprong.

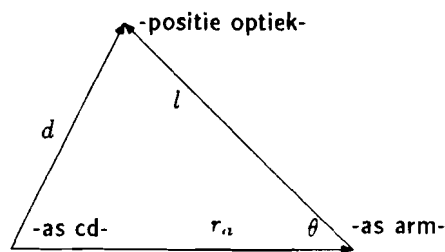


Figuur A.2: Overdracht bij langzame sprong.

Appendix B

Overdracht stroom door arm naar snelheid ten opzichte van compact disc.

stel. zie figuur B:



Figuur B.1: Schematische weergave van opstelling arm.

- d =afstand optiek tot as van compact disc.
- l =lengte arm.
- r_a =afstand as compact disc tot as arm.
- θ = hoek tussen arm en vector as arm naar as compact disc.

Dan geldt:

$$d^2 = l^2 \sin^2 \theta + r_a^2 - 2 \cdot r_a \cdot l \cdot \cos \theta + l^2 \cos^2 \theta = l^2 + r_a^2 - 2 \cdot r_a \cdot l \cdot \cos \theta \quad (\text{B.1})$$

of:

$$d = \sqrt{l^2 + r_a^2 - 2 \cdot r_a \cdot l \cdot \cos \theta} \quad (\text{B.2})$$

De verhouding tussen afgelegde weg op de plaat en afgelegde weg van de arm is $\frac{dl}{d(l \cdot \theta)}$.

of

$$\frac{dd}{d(l.\theta)} = \frac{r_a \cdot \sin \theta}{\sqrt{l^2 + r_a^2 - 2 \cdot r_a \cdot l \cdot \cos \theta}} \quad (\text{B.3})$$

Voor het loopwerk van het CDM-1 type geldt: $r_a = 43$ mm en $l = 35$ mm. In het werkgebied, 35,5-95,5 graden, geldt: $\frac{dd}{d(l.\theta)}$ ligt tussen 0,74 en 1,0. De snelheid van de arm ten opzichte van de plaat is dus altijd tussen 0,74 en 1,0 keer de snelheid van de arm. In het vervolg wordt gecorrigeerd met 0,87.

Volgens [1] geldt: het koppel $T = K \cdot I$. Ook $\frac{d^2 \theta}{dt^2} = \frac{I}{J}$. Dus:

$$v = v_a \cdot 0,88 = \frac{d\theta}{dt} \cdot 1,0,87 = \frac{K \cdot I \cdot 1,0,87}{J \cdot j\omega} \quad (\text{B.4})$$

Met $K = 40$ mNm/A, en $J = 55 \cdot 10^{-6}$ kg/m.m levert dit het eindresultaat:

$$v = \frac{22 \cdot I}{j\omega} \quad (\text{B.5})$$

Appendix C

Verband tussen tijd en aantal sporen op een compact disc.

Wanneer s de spoed, en v de lineaire snelheid van een compact disc zijn, dan kan het verband tussen de tijd en het aantal sporen dat van het binnenste spoor naar het spoor van die tijdcode moet worden overgestoken worden bepaald.

De lengte van de afgelegde weg over de spiraal van het begin tot de tijdcode is, als de sporen als concentrische cirkels worden gezien met straal r_n :

$$L = \sum_{n=0}^N 2 \cdot \pi \cdot r_n \quad (\text{C.1})$$

De afgelegde weg L is gelijk aan $v \cdot t$, met t in seconden. De straal van de cirkels r_n is gelijk aan $r_0 + s \cdot n$, met r_0 de straal van het binnenste spoor, dus:

$$v \cdot t = \sum_{n=0}^N 2 \cdot \pi \cdot (r_0 + s \cdot n) = 2 \cdot \pi \cdot r_0 \cdot (N + 1) + \pi \cdot s \cdot (N^2 + N) \quad (\text{C.2})$$

Dan is echter de tijd bekend na het afleggen van het N -de spoor, ik wil echter de tijd weten aan het begin van het N -de spoor. De tijd nodig voor het laatste spoor is:

$$t_l = 2 \cdot \pi \cdot r_N / v = 2 \cdot \pi \cdot (r_0 + s \cdot N) / v \quad (\text{C.3})$$

Dus:

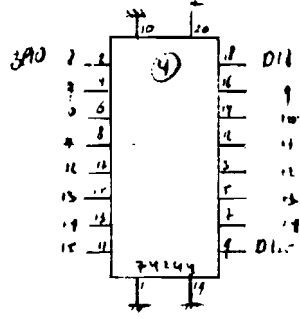
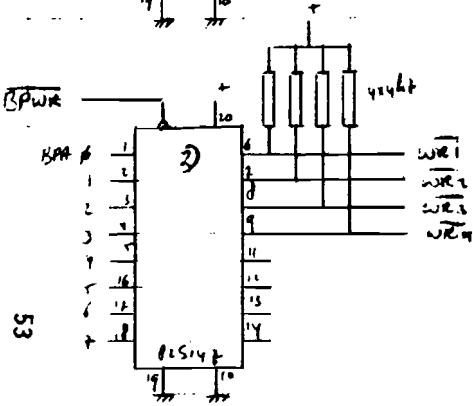
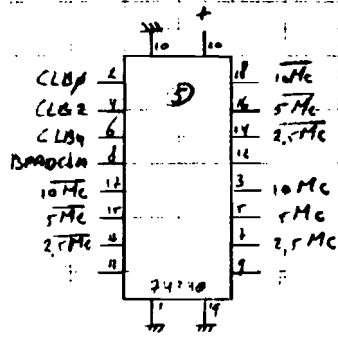
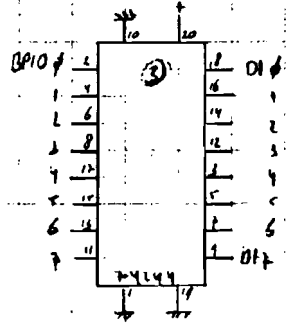
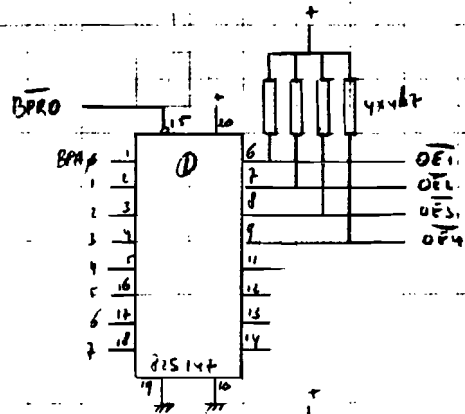
$$t = \frac{2 \cdot \pi \cdot r_0 \cdot N}{v} + \frac{\pi \cdot s \cdot N^2}{v} \quad (\text{C.4})$$

Appendix D

Schema's van de hardware.

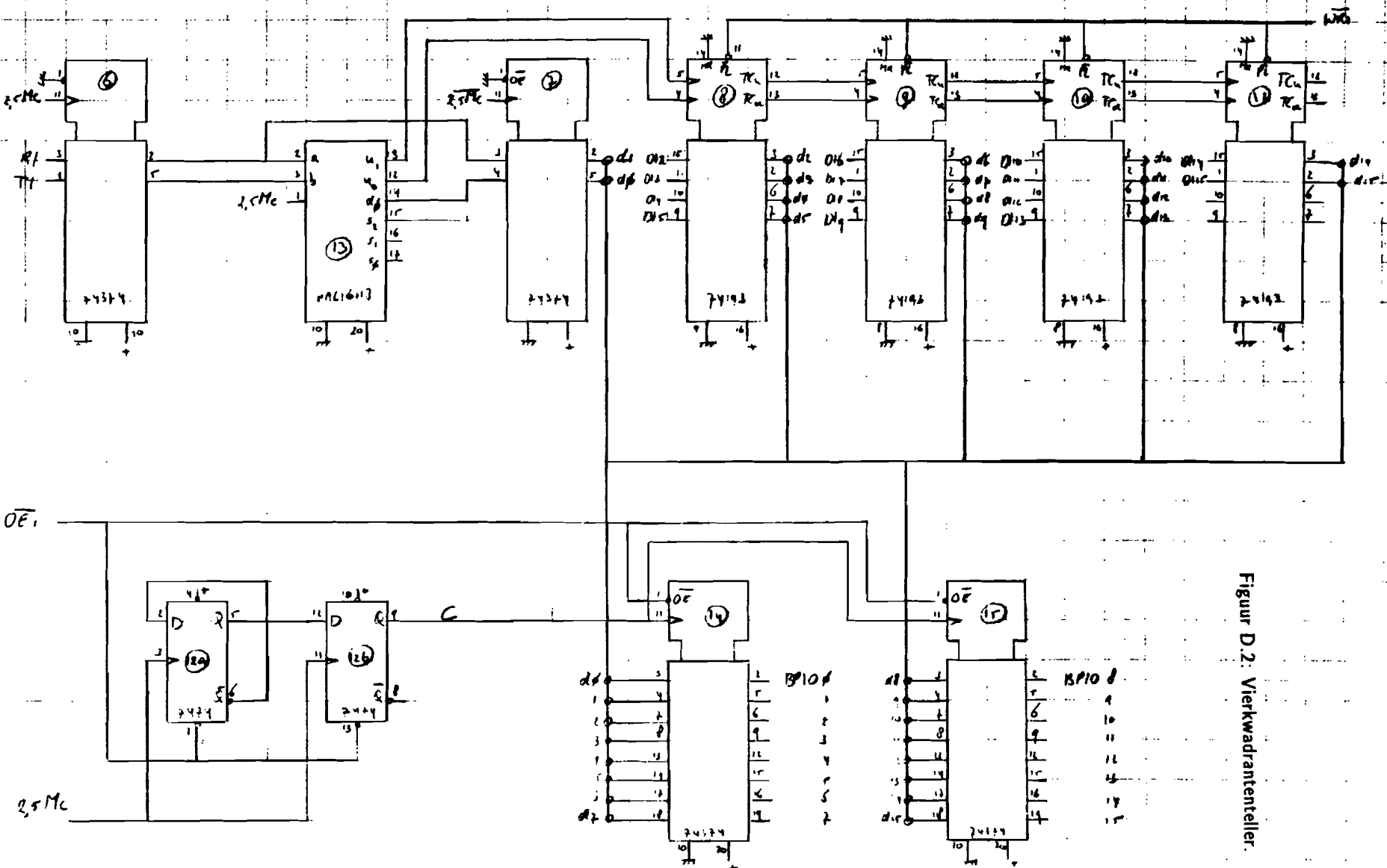
nr	type	plaats	schema
1	82S147	a1	1
2	82S147	b1	1
3	74LS244	a2	1
4	74LS244	b2	1
5	74LS240	a3	1
6	74LS374	c3	2
7	74LS374	b3	2
8	74LS193	b4	2
9	74LS193	b5	2
10	74LS193	b6	2
11	74LS193	b7	2
12	74LS74	a6	2
13	PAL16L8	c4	2
14	74LS374	a4	2
15	74LS374	a5	2
16	74LS74	c1	3
17	74LS74	d1	3
18	74LS163	e1	3
19	74LS163	e2	3
20	74LS163	e3	3
21	74LS374	d2	3
22	74LS374	d3	3
23	74LS74	c7	3
24	74LS163	d7	3
26	74LS08	f3	3
27	2732	d4	3
28	74LS374	c5	3
29	2732	d6	3
30	74LS374	c6	3
31	74LS161	e4	4
32	74LS161	e5	4
33	74LS161	f4	4
34	74LS161	f5	4
35	74LS74	f2	4
36	74LS374	e6	4
37	74LS374	e7	4
38	74LS123	p1	5
39	74LS123	p2	5
40	74LS08	p3	5
41	74LS74	p4	5
42	74LS74	p5	5

Tabel D.1: Onderdelenlijst.



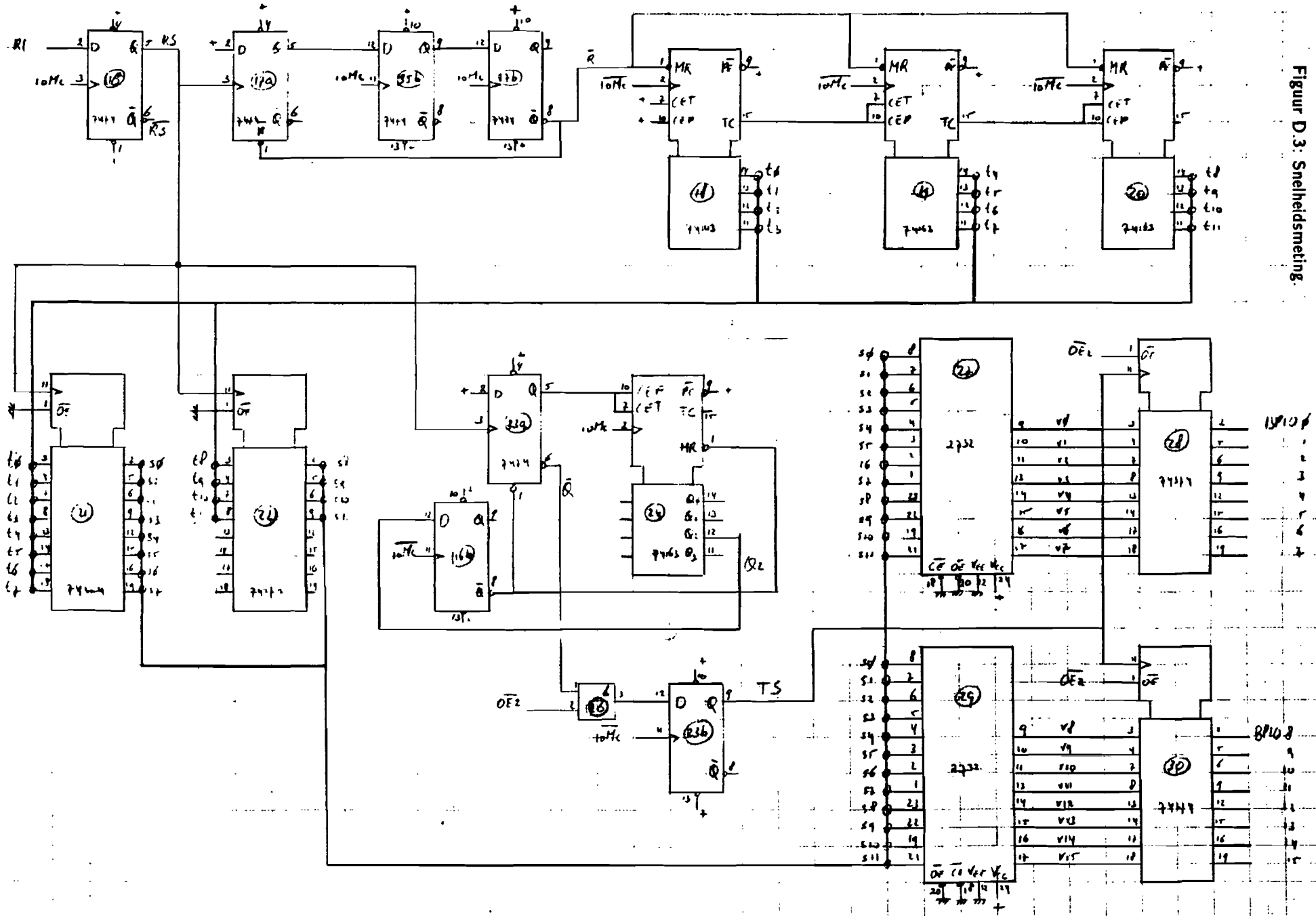
buslogica ①

Figur D.1: Buslogica.



Figur D.2: Vierkantenteller.

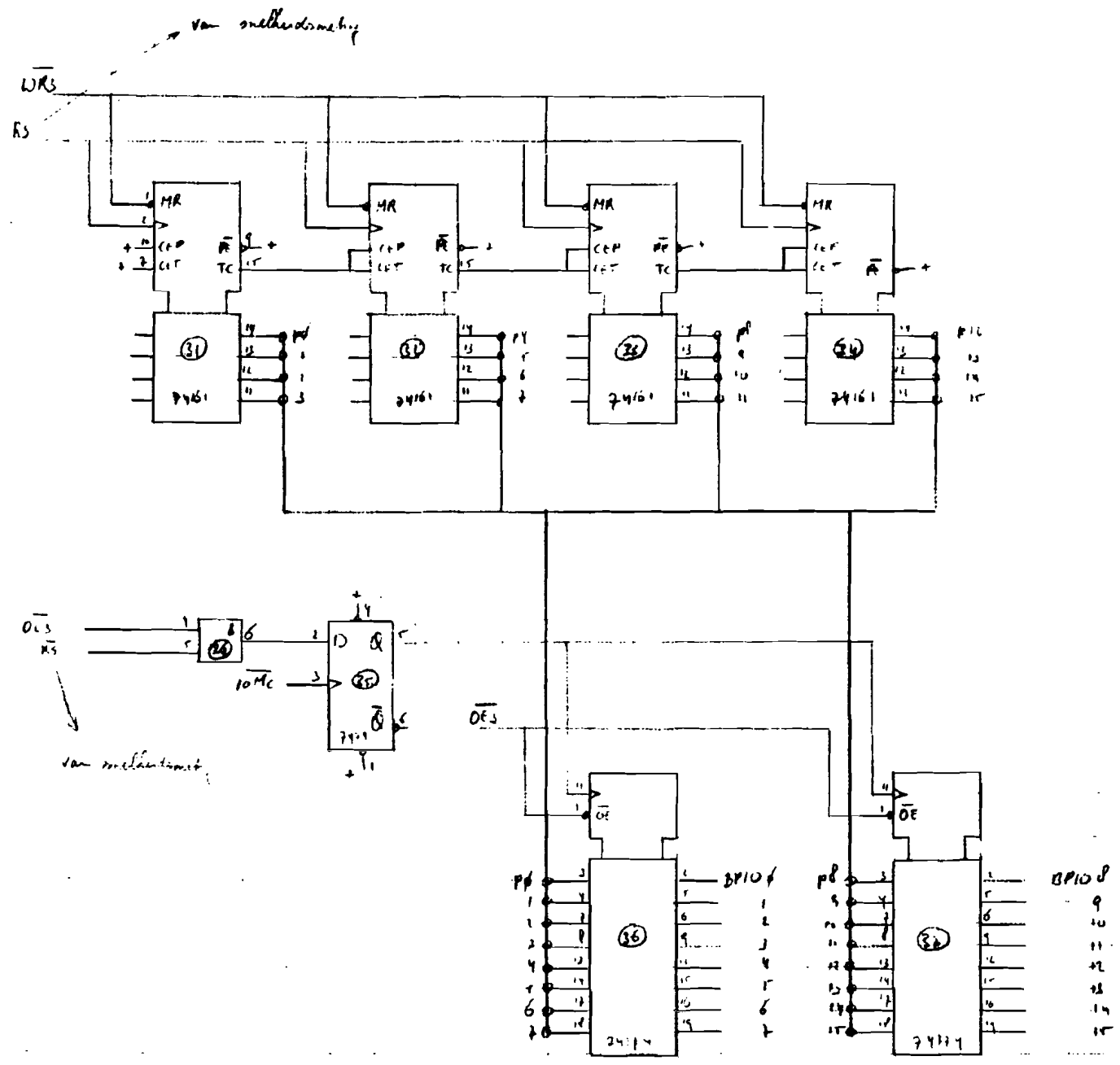
Vierkantenteller (2)



Figur D.3: Snehedsmeting.

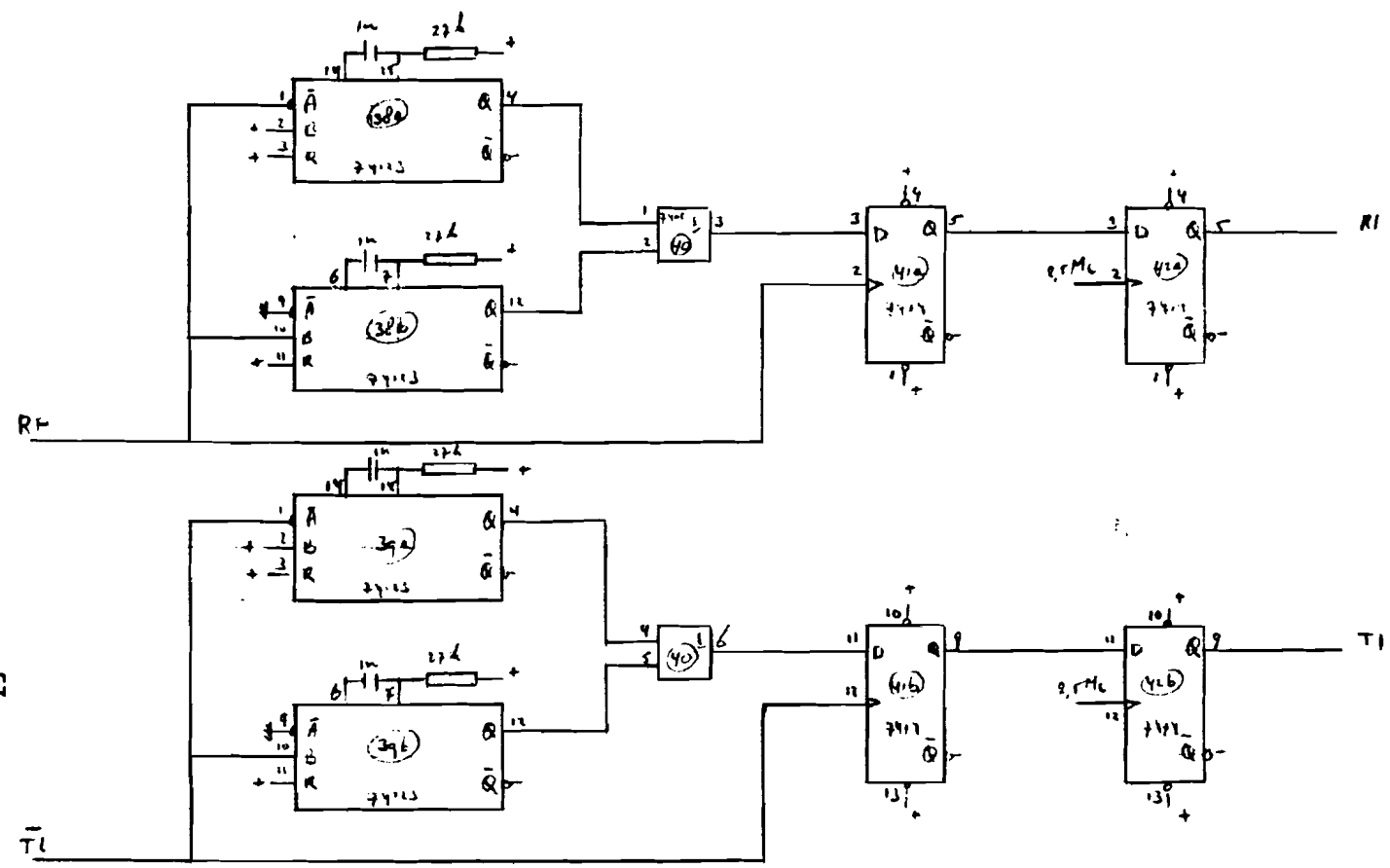
pos. hækler

(U)



Figur D.4: Positieller.

⑤ debouncing RE & TL



Figur D.5: Debounce-schaltung.

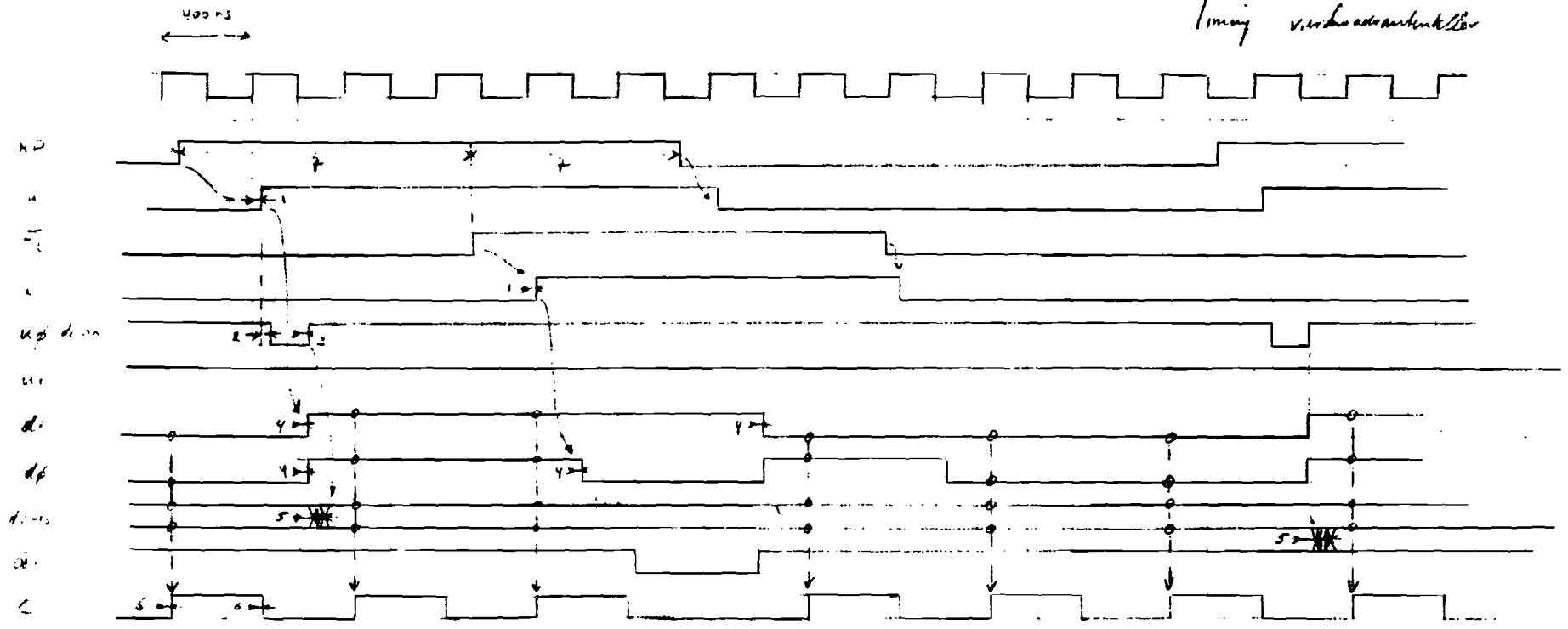
Appendix E

Timing van de hardware.

nr	description	min in ns	max in ns
1	clock high to A or B transition		30
2	A high to U0 low		30
3	clock low to U0 transition		30
4	clock low to D0 or D1 transition		30
5	U0 high to D2-D15 transition		100
6	clock high to C transition		40
7	RP transition to \overline{TL} transition	500	
8	clock high to RS transition		40
9	clock high to \overline{R} transition		40
10	clock low to Qx count transition		30
11	clock low, \overline{R} low, to Qx zero		40
12	RS transition to Vx transition		30
13	Sx transition to Vx transition		40
14	clock high to \overline{Q} low		40
15	clock high to Qx transition		25
16	Q2 high to \overline{Q} high		40
17	clock low to TS transition		40
18	TS transition to BPIO transition		30
19	clock high to RS transition		40
20	RS high to Px count transition		25
21	clock high to QC transition		40
22	QC high to BPIO transition		30
23	$\overline{WR3}$ to Px low		30

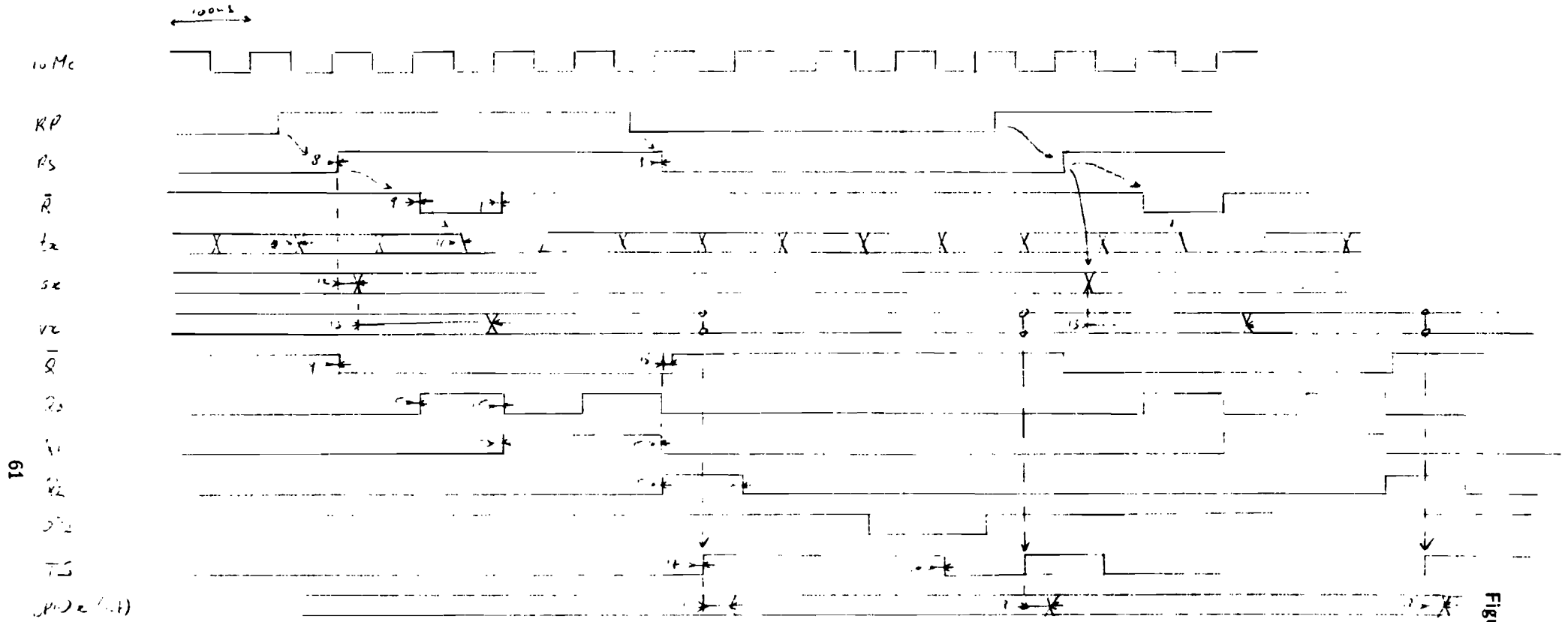
Figuur E.1: Tijden in timing-diagrammen.

Timing vierkwadrantenteller



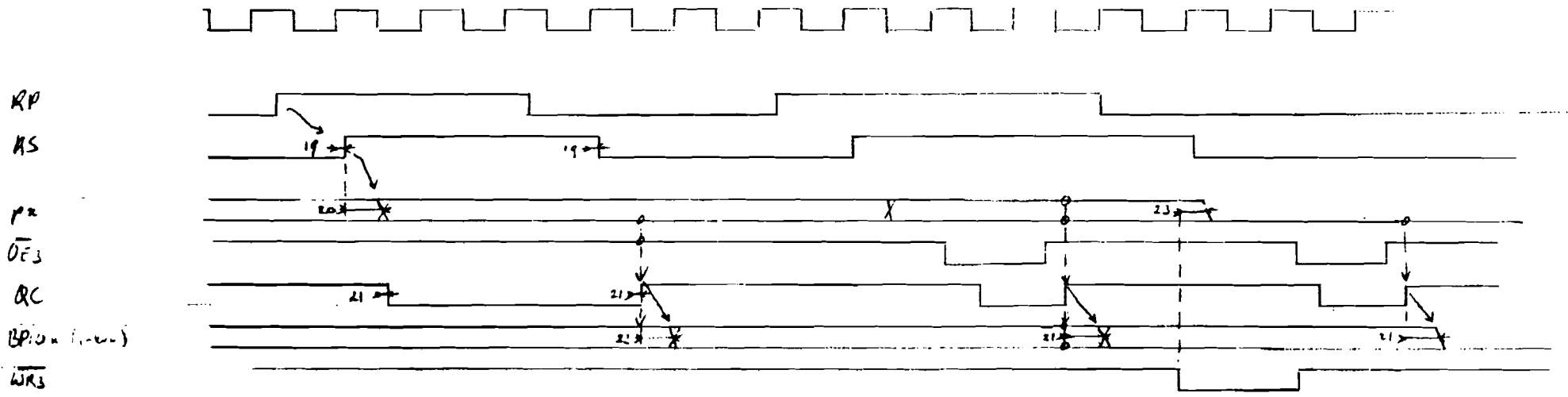
Figuur E.1: Timing vierkwadrantenteller.

Timing verification



Figuur E.2: Timing snelheidsmeting.

Timing posreteller



Figur E.3: Timing posreteller.

Appendix F

Programming PAL.

ABEL(tm) Version 1.10 - Document Generator
clock-pulse generator for four-quadrant-counter
programmer: Meine van der Meulen
date: 24 november 1986
Equations for Module counter

Device IC1

Reduced Equations:

S2 = !(C1 # RP & !S1 # !S0 & !S1 # !S1 & !S2);

S1 = !(!C1 & !S1
C1 & TLN
S0 & S2
!RP & !S0 & !S1
RP & S0 & !S2);

S0 = !(!RP & TLN
!RP & !S0 & !S1
!RP & !S1 & S2
RP & S1 & !S2
RP & !S0 & !S1 & !TLN);

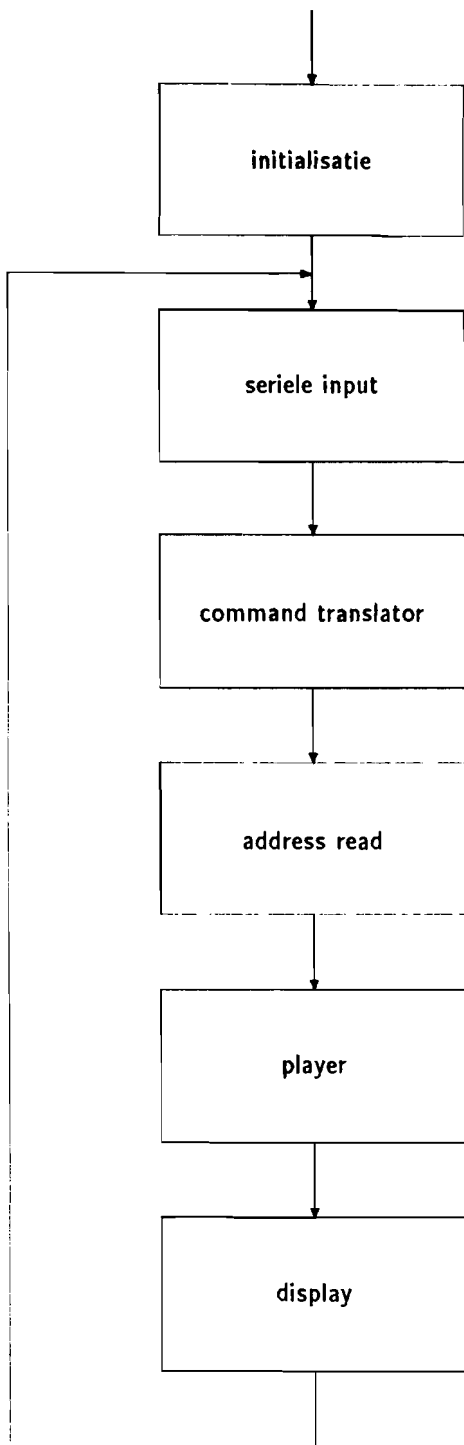
U1 = !(S0 & S1 & !S2);

U0 = !(!S0 & S1 & !S2);

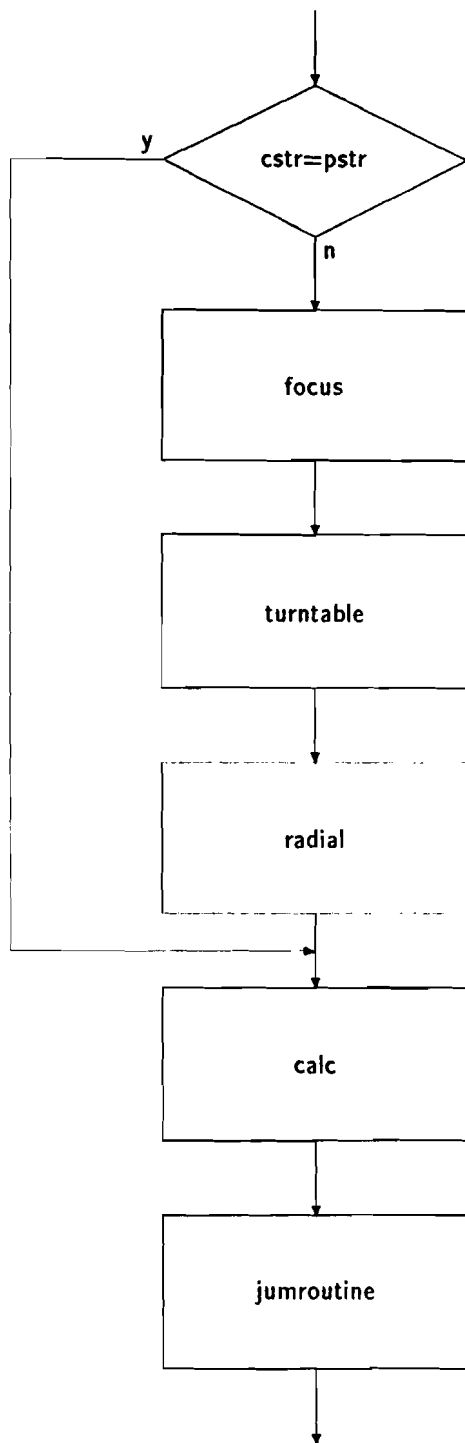
D0 = !(!RP & !TLN # RP & TLN);

Appendix G

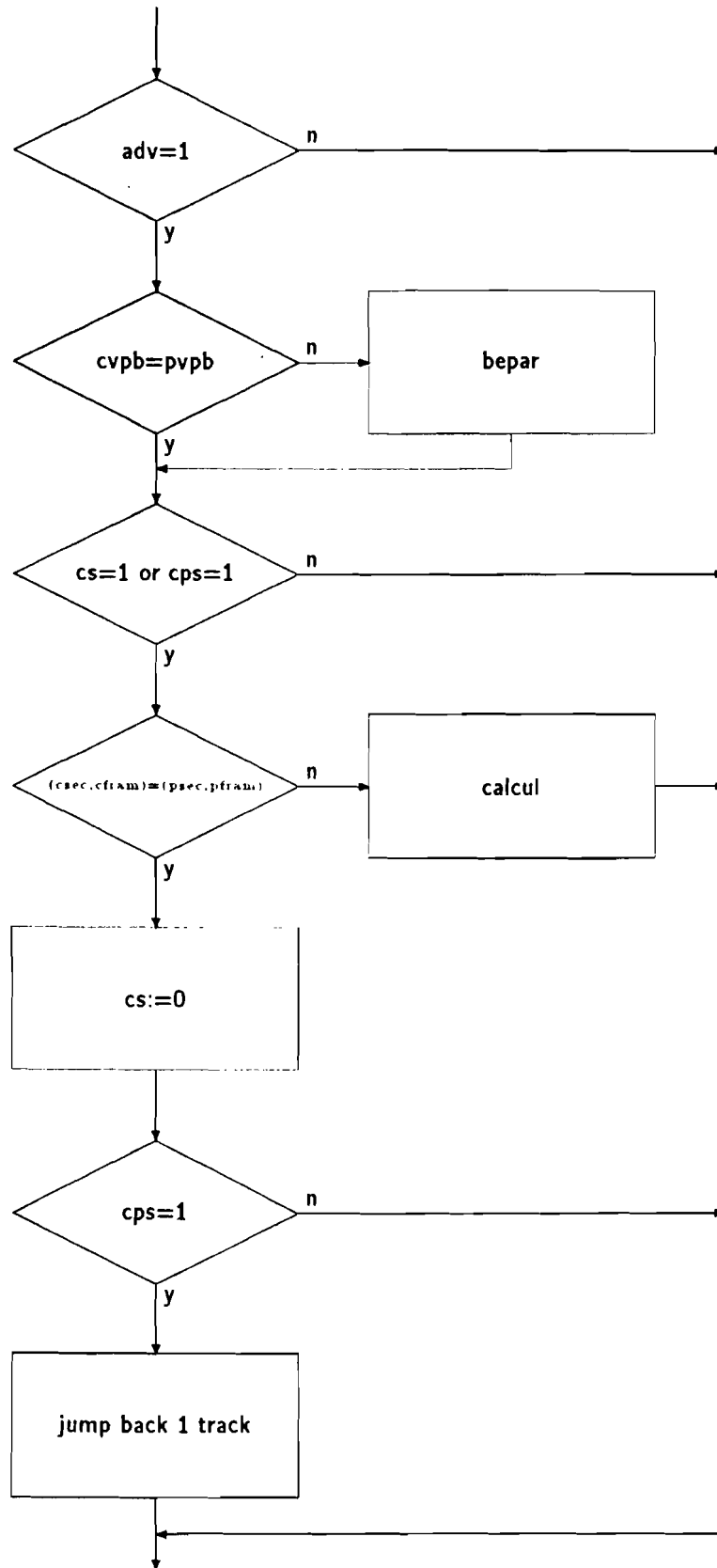
Flowdiagrammen.



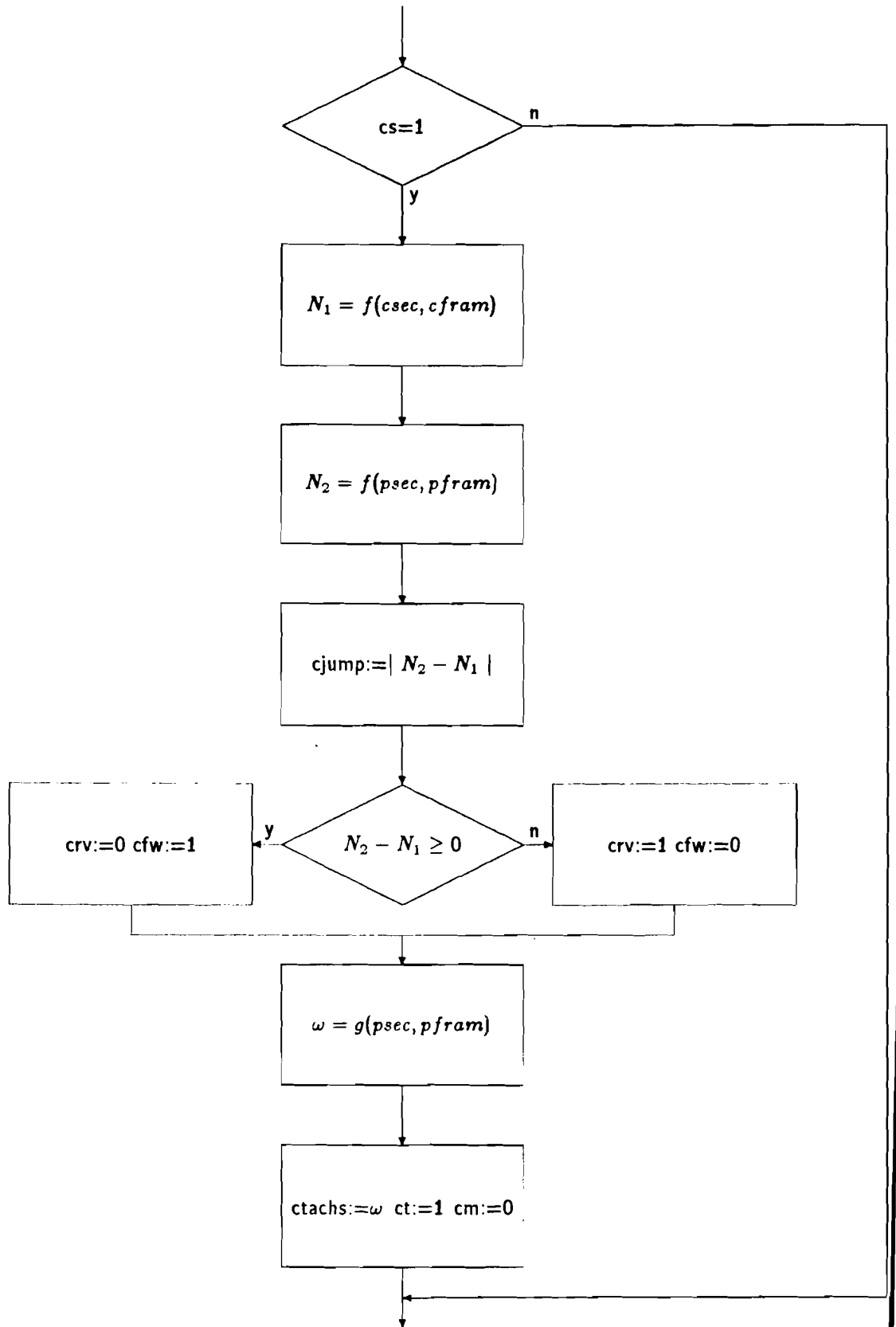
Figuur G.1: Flowdiagram van hoofdprogramma.



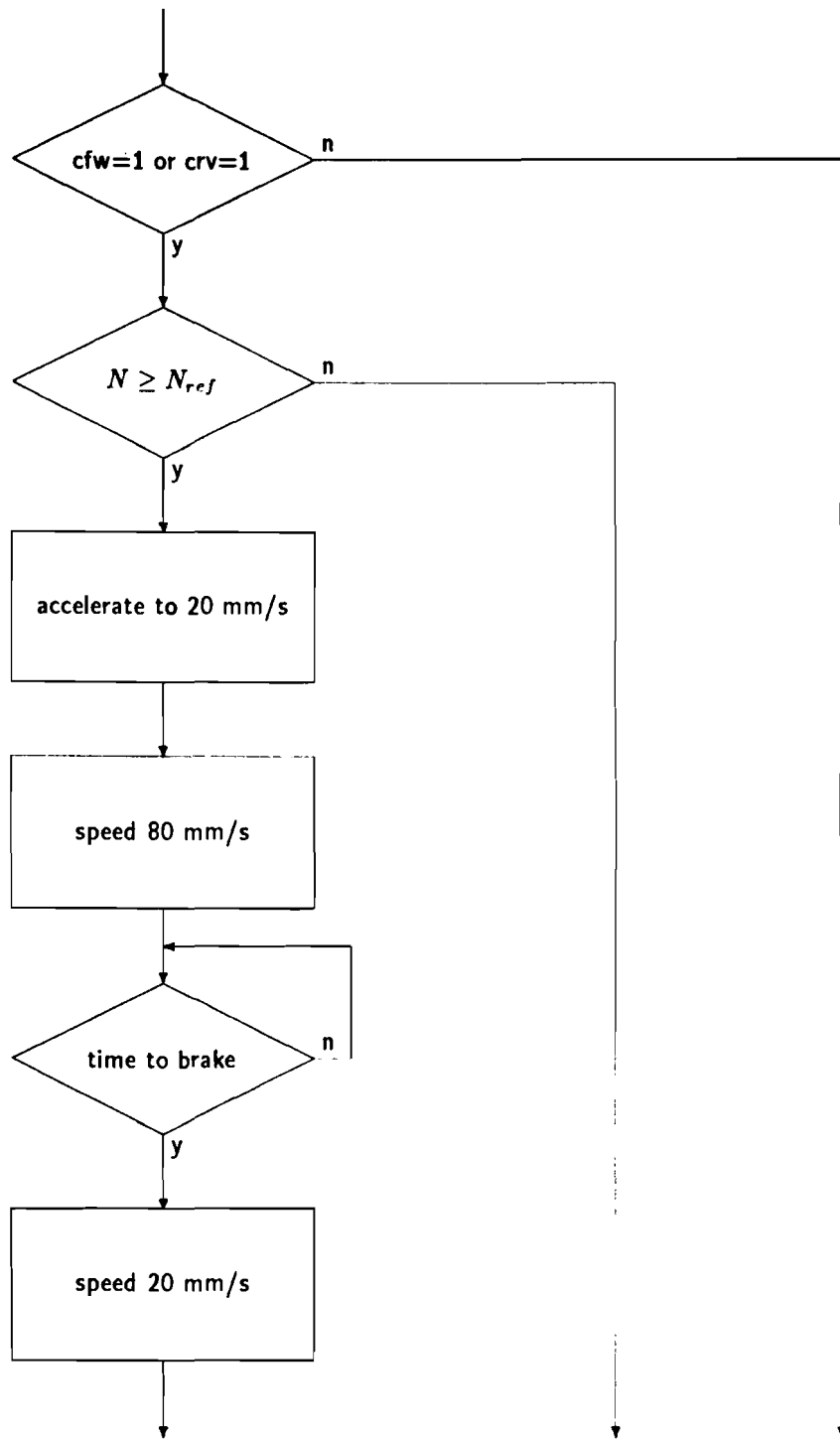
Figuur G.2: Flowdiagram van playerroutine.

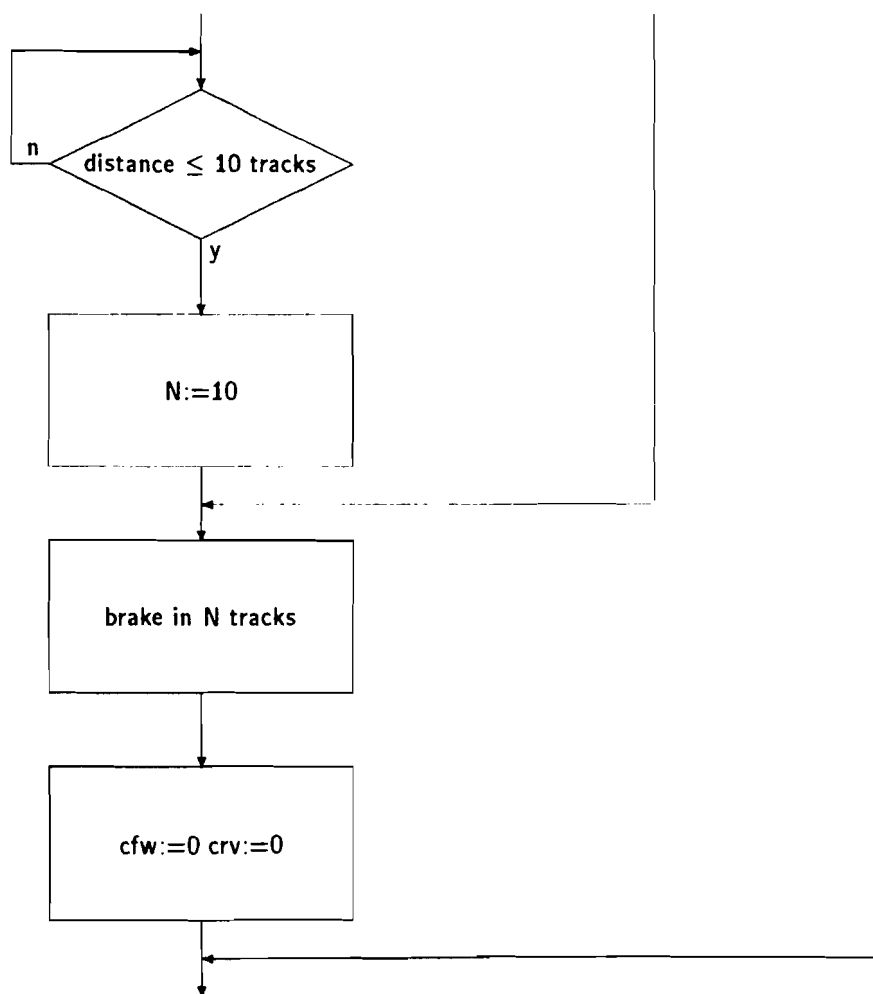


Figuur G.3: Flowdiagram van calcroutine.



Figuur G.4: Flowdiagram van calculroutine.





Figuur G.5: Flowdiagram van jumproutine.

Appendix H

programma's voor PCB5010.


```

(* This is the jumproutine. It moves the arm corresponding crv or cfw in
r14, in a reverse or forward direction. The number of tracks that is being
jumped is stored in cjump. During this routine r3 is used to store the number
of tracks that have to be jumped, the direction of the jump is available
in r4. R1 and r2 are used for calculations.
*)

```

```

jump:
    pg:=1;
    r1:=#300;
    acu(rama,aar):=dac4;      (* triggerpuls op dac4      *)
    call dac4;
    r1:=maskcfw;
    and(r1,r14,r4);          (* r4=0 is jump backwards  *)
    if not z goto setforwards;
    r1:=maskcrv;
setforwards:
    or(r15,r1,r15);          (* set jumpbit in status    *)
    acu(ramb,aar):=cjump;    (* pointer to #tracks       *)
    r1:=18;                  (* maximum number of tracks in posmode *)
    r3:=ramb;                (* r3 = #tracks             *)
    sub(ramb,r1);            (* #tracks > max ?         *)
    if n goto jumpinposloop; (* no: jump in posloop     *)
    call intdis;            (* disable interrupt       *)
    acu(rama,aar):=squarecounter; (* aar -> slowpos         *)
    pass(r4);                (* jump forwards ?         *)
    if z goto backacc;
forwacc:po:=-100;           (* yes: forward accelaration *)
    goto setacc;
backacc:po:=100;           (* no: backward accelaration *)
setacc: acu(ramb,aar):=rcp;  (* set pid in positioncontrol mode *)
    ramb:=rnointcontrol;
    pst:=r7;                (* enable interrupt       *)
wait8tracks:               (* accelerate during 8 tracks *)
    acu(rama,aar):=squarecounter ;
    ;
    pass(pi,r1);            (* if |r1|<68 then 8 tracks *)
    (* are passed          *)
    if not n goto absolute;
    neg(r1,r1);
absolute:r2:=68;
    sub(r1,r2);
    if not n goto wait8tracks;
    r1:=8;

```

```

        sub(r3,r1,r3);          (* r3=#tracks left to be jumped *)
speedref:
pg:=1;
acu(rama,aar):=-fastposcounter;
po:=-#0000;                    (* reset fastpos *)
call intdis;                    (* disable interrupt *)
acu(ramb,aar):=-rcp;           (* set pid in velocity mode *)
ramb:=-rspeedcontrol;

acu(ramb,aar):=rsp;           (* give setpoint *)
pass(r4);                       (* jump forwards ? *)
if z goto setspeedbackwards;
ramb:=800;                       (* 50 mm/s * 16 *)
goto speedrefcont;
setspeedbackwards:
ramb:=-800;                       (* -50 mm/s * 16 *)
speedrefcont:
pst:=r7;                         (* enable interrupt *)
test: goto waittillslowspeed;
waittillslowspeed:
acu(rama,aar):=-fastposcounter;
r1:=400;
sub(r3,pi,r2);                  (* calculate distance to track *)
sub(r2,r1);                      (* distance > 400 tracks ? *)
if not n goto waittillslowspeed; (* yes: wait *)
setslowspeed:
acu(ramb,aar):=-rsp;           (* give setpoint 13 mm/s *)
pass(r4);                       (* jump forwards ? *)
if z goto setslowspeedbackw;
ramb:=-256;                       (* 16 mm/s * 16 *)
goto waittillneartrack;
setslowspeedbackw:
ramb:=-256;                       (* -16 mm/s * 16 *)
waittillneartrack:
acu(rama,aar):=-fastposcounter;
r1:=10;
sub(r3,pi,r2);                  (* calculate distance to track *)
sub(r2,r1,r2);                  (* distance > 10 tracks ? *)
dec(r2);
if not n goto waittillneartrack; (* yes: wait *)
r3:=10;                          (* jump rest of tracks *)
pg:=1;
jumpinposloop:
r1:=0;                          (* triggerpuls on dac4 *)
acu(rama,aar):=dac4;
call daca;
asl(r3,r3);                      (* #tracks * 4 *)

```

```

    asl(r3,r3);
    call intdis;          (* disable interrupt          *)
    acu(ramb,aar):=rcp;   (* pid in position control      *)
    ramb:=rnointcontrol ac(ramb,inca);
    ramb:=#0 ac(ramb,inca); (* clear integrator          *)
    ramb:=#0;
    pass(r4);            (* jump forwards ?            *)
    if z goto setpoint;
    neg(r3,r3);          (* yes: r3:=-r3              *)

setpoint:
    acu(rama,aar):=squarecounter; (* set slowpos          *)
    po:=r3;
    acu(ramb,aar):=rsp;   (* pos setpoint = 0        *)
    ramb:=0;
    pst:=r7;            (* enable interrupt        *)
    goto waittillontrack;

waittillontrack:
    dec(pi,r1);          (* wait till pos = 1 or 2  *)
    asr(r1,r1);
    if not z goto waittillontrack;
    acu(ramb,aar):=time; (* time := -5ms           *)
    ramb:=-50;
wait5ms: pass(ramb);    (* wait 5 ms              *)
    if n goto wait5ms;
    pg:=1;
    acu(ramb,aar):=rcp;   (* activate position control *)
    ramb:=rposcontrol;
    rl:=#fcff;          (* reset crv or cfw        *)
    and(r14,r1,r14);
    and(r15,r1,r15);

exitjump:
    pg:=0;
    goto play6;         (* resume playeroutine     *)

```

```

(*-----*)
(* interrupt routine   save pst en alle adresregisters van de acus *)
(*               Registers R8,R9,R10,R11,R12,R13 zijn *)
(*               interruptregisters , wanneer meer registers nodig zijn moet men *)
(*               ze in de initialisatie save net als pst *)
(*-----*)
intrpt: r11:=pst ;
        r10:=acu(rama);
        pst:=pstint ;
        acu(rama,aar):=pstack;          (* stack in rama   fast-mode *)
        rama:=r11 r11:=acu(rom) ac(rama,deca);  (* ar pst *)
        rama:=r10 ac(rama,deca);          (* ar rama *)
        rama:=r11 ac(rama,deca);          (* ar rom *)
        rama:=acu(ramb) r11:=pg ac(rama,deca); (* ar ramb *)
        rama:=r11 r10:=bsr ac(rama,deca);  (* page *)
        acu(ramb,aar):=pidstat;          (* acu(ramb) --> PIDSTATUS *)
        rama:=r10 ;                      (* bsr *)
        r8:=ramb ;                        (* r8:=pidstat voor de hele int *)
        acu(ramb,aar):=fcp;               (* acu(ramb) --> FCP *)
        pg:=#0005;                        (* sigma-pi busmaster *)
        pass(ramb);
        if z goto intrad;                  (* if FCP=0 goto intrad *)
        pass(r8) ac(ramb,deca);
        r11:=#0000 ac(ramb,deca) ;        (* indien FIN0=0 *)
        if N goto intl ac(ramb,deca) ;    (* if FIN0 goto intl *)
        acu(rama,aar):=adcl;
        call adc;                          (* r11:= adc-input *)
        r12:=#0004 ;
        or(r8,r12,r8) ;                    (* FPID:=1 t.b.v. mainprogr *)
intl:   ramb:=r11 ac(ramb,inca);           (* FPIN:=r11 *)
        r12:=#0020 ;                       (* masker voor FDIF in pidstatus *)
        sub(ramb,r11,r11) ac(ramb,inca);  (* r11:= FSP-FPIN -FERR *)
        call pid ;
        acu(rama,aar):= dacl;              (* zet adres voor dacl klaar *)
        call dac ;                         (* dac-output := r11 *)
intrad: acu(rama,aar):=cntpid ;
        pg:=3 ;
        dec(rama,r11) ;
        asr(r11) rama:=r11 ;
        if not C goto intmot ;             (* if cntpid=even goto intmot *)
        pg:=7 ;
        acu(ramb,aar):=rcp ;
        pass(ramb) ;
        if Z goto endint ;                 (* if rcp=0 goto endint *)
        R11:=#4000 ;                       (* masker voor RIN0 in pidstat *)
        and(r11,r8) ;
        R11:=#0000 ;
        if not Z goto int2 ;              (* if RIN0=1 goto int2 *)

        acu(ramb,aar):=rcp;                (* which controlmode ? *)
        r12:=rposcontrol;
        sub(ramb,r12);
        if z goto calcpossp;              (* positioncontrol *)
        r12:=rnointcontrol;
        sub(ramb,r12);
        if z goto calcpossp;              (* positioncontrol *)
        r12:=rspeedcontrol;
        sub(ramb,r12);
        if z goto calcspeedsp;            (* speedcontrol *)
        goto intmot;

```

```

calcpossp:
acu(rama,aar):-squarecounter; (* aar points to poscounter *)
acu(rom,aar):-remax; (* arr -> remax *)
r12:=pi; (* read slowpos *)

(* limit function of slowpos *)
pass(pi,r12); (* read slowpos again *)
if not n goto counterpositive; (* test #tracks<0 *)

counternegative: (* compare with limit *)
r11:=110;
add(r11,r12);
if not n goto extendscurve; (* arm between limits -> go on *)
r11:=-3920; (* otherwise : r11 = max *)
goto int2;

counterpositive:
r11:=113; (* pos test #tracks>tracklimit *)
sub(r11,r12); (* no : go on *)
if not n goto extendscurve; (* yes : r11 = max *)
r11:=3920;
goto int2;

(* extend s-curve *)
extendscurve:
dec(r12,r12); (* r12:= trunc (slowpos-1) *)
asr(r12,r11);
asl(r11,r12); (* offset for extended curve *)
mpy(rom,r12,0) ac(rom,inca);
bsr:=1;
and(r12,rom,r10) ac(rom,inca); (* extract t1 from slowpos *)
r13:=1sp; (* store offset in r13 *)
acu(rama,aar):-adc2; (* r11:= adc-input *)
call adc;
bsr:=12; (* normalize r11 *)
mpy(r11,rom,0);
pass(r10); (* t1 = 0 ? *)
if z goto radtlz; (* no : subtract re-signal *)
sub(r13,1sp,r11);
goto int2; (* yes: add re-signal *)
radtlz: add(r13,1sp,r11);
goto int2;

(* read speed *)
calcspedsp:
acu(rama,aar):-speed;
r12:=maskcrv;
r11:=pi;
r11:=pi;
rpr:=4;
asr(r11,r11);
and(r14,r12);
if z goto calcforwards; (* neg speed if reverse jump *)
neg(r11,r11);

calcforwards:
goto int2;

int2:

```

```

r13:=r11 ;
asr(r11,r11);
acu(rama,aar):=dac3 ;
call dac ;
r11:=r13 ;

acu(ramb,aar):=rpin ;
ramb:=r11 ac(ramb,inca) ;      (* RPIN:=r11          *)
r12:=#0010 ;                  (* masker voor RDIF in pidstat *)
sub(r11,ramb,r11) ac(ramb,inca) ; (* RERR := RPIN-RSP          *)
call pid ;
acu(rama,aar):=dac2 ;
call dac ;                    (* output radiele regeling    *)
goto endint ;
intmot: acu(rama,aar):=cntpid ;
pg:=3 ;
acu(ramb,aar):=mcp ;
pass(rama) ;
if not Z goto endint ;        (* if cntpid=0 goto endint    *)
rama:=#000A ;                 (* preset cntpid              *)
r11:=#2000 ;
pass(ramb) ;
if Z goto endint ;           (* if mcp=0000 goto endint    *)
and(r11,r8) ;
r11:=#0000 ;
acu(ramb,aar):=mpin ;
if not Z goto int3 ;          (* if MIN0=1 goto int3        *)
(* voorbereking voor motorregeling *)
int3: ramb:=r11 ac(ramb,inca) ; (* MPIN:=r11                  *)
r12:=#0008 ;                 (* masker voor MDIF in pidstat *)
call pid ;
(* nabewerking *)
endint: acu(ramb,aar):=time ;  (* increment time             *)
;
inc(ramb,r10);
ramb:=r10;

acu(ramb,aar):=pidstat ;
ramb:=r8 ;
acu(rama,aar):=#007A ;
ac(rama,inca);               (* restore                    *)
bsr:=rama ac(rama,inca);     (* bsr                         *)
pg:=rama ac(rama,inca);      (* page                        *)
acu(ramb,aar):=rama ac(rama,inca); (* acu(ramb)                  *)
acu(rom,aar):=rama ac(rama,inca); (* acu(rom)                   *)
r10:=rama;
r11:=rama ;
acu(rama,aar):=r10 ;         (* restore acu(rama)          *)
pst:=r11;                    (* pst                         *)
reti;

end;

```

```

program term;

(* This routine calculates track pitch, s, en tracking velocity, v, *)
(* using two measurements (n1,t1) and (n2,t2), with n2=2*n1: *)
(* *)
(* v = 4*pi*r0*n1/(4*t1-t2) *)
(* s = (t1*v-2*pi*r0*n1)/(pi*n1*n1) *)
(* *)
(* v is stored in r15 and s in r6 *)

const
n1 = 20;
t1 = 40;
t2 = 41;
r0 = 80;          (* 2*pi*r0 2**14 *)
pi10= 81;        (* pi* 2**10 *)

libmacro divdpl;

begin
fqr:=1;
acu(rama,m),acu(ramb,m):-#ffff;
acu(rom,m):-#ffff;
acu(rama,aar):=n1;
rama:=10000;
acu(ramb,aar):=t1;
acu(rom,aar):=r0;
ramb:=1754 ac(ramb,inca);
ramb:=4398 ac(ramb,deca);

bsr:=1;
mpy(rama,rom,0);          (* acr = 2*pi*r0*n1 *)
asl(ramb,r4);            (* r4 = t1*t2 *)
r2:=msp;                (* move acr to (r2::r3) *)
r3:=lsp ac(ramb,inca);  (* pointer ramb points to t2 *)
asl(r4,r4);             (* r4 = t1*t4 *)
sub(r4,ramb,r4) ac(rom,inca); (* pointer rom points to pi *)
                          (* r4 = t1*t4-t2 *)

asr(r4,r4);

call divide;
r15:=r6;                (* calculated v in r15 *)

(* divide (r2::r3)/r4 -> v* 2**14 in r6 *)
s
acu(ramb,aar):=t1;
mpy(rom,rama,0);        (* acr = pi*n1* 2**10 *)
bsr:=12 ac(rom,deca);  (* arr to r0 *)
mpy(rama,lsp,0);       (* acr=pi*n1*n1* 2**1 *)
bsr:=15;
r4:=lsp;               (* r4=pi*n1*n1* 2**15 *)
mpy(ramb,r15,0);       (* acr := v*t *)
mpy(rama,-rom,acr);    (* acr := v*t-n1*2*pi*r0*n *)
bsr:=1;
r3:=lsp ac(rom,deca);  (* (r2::r3) * 32 -> (r2::r3) *)
r2:=msp;
mpy(r3,rom,0) ac(rom,deca);
r3:=r3;
r3:=lsp;
mpy(r2,rom,acrs);

```

```
    bsr:=2;
    r2:=lsp;
    (* divide (r2::r3)/r4 -> s* 2**34 in r6 *)
    call divide;
stop:  goto stop;
divide: divdpl(r2,r3,r4,r5,r6,r7,r8,r9,r10,ramb);
    ret;
end;
rom(78)=64;
rom(79)=32;
rom(80)=2574;
rom(81)=3217;
```



```

program newtonraphson;

(* This program calculates a solution of the formula: *)
(* *)
(*      2*pi*r0*n1+pi*s*n1*n1-v*t=0 *)
(* *)
(* The solution of this problem gives the tracknumber corresponding *)
(* to a certain timecode on disc that is given in ramb. *)
(* *)
(* This solution is found by using the algorithm of newton_raphson: *)
(* Procedure approx in the program calculates: *)
(* *)
(*      f(n1)=(2*pi*r0*n1*n1+pi*s*n1-v*t)/(2*pi*r0+2*pi*s*n1) *)
(* *)
(* Using this formula, a next best solution of the approximation *)
(* of a track corresponding to a certain time on the disc can be *)
(* found by: *)
(* *)
(*      n2 = n1 - f(n1) *)
(* *)
(* with f(n1) the formula explained above. *)
(* *)
(* after execution the calculated tracknumber is stored in cjump *)
(

const
(* these constants give the address of the parameters in ram and rom, they
are arbitrary chosen and can be changed *)
cjump      =83;
v14address =12;
s34address =13;
timeaddress =84;
frameaddress=85;
pil0       =40;

libmacro divdpl;          (* include double precision division *)

begin
    acu(rama,m),acu(ramb,m):=#ffff;    (* initialisation *)
    acu(rom,m):=#ffff;
    acu(rama,aar):-v14address;
    acu(ramb,aar):-timeaddress;
    rama:=19660 ac(rama,inca);          (* v14=1,2 m/s *)
    rama:=29205;                        (* s34=1,7e-6 m *)
    ramb:=2400 ac(ramb,inca);          (* t=5 min*)
    ramb:=0;

    (* This is the start of the program *)

    r1:=10000;                          (* first approximation *)

newraph:call approx;
    asr(r6,r6);
    if z goto exit;
    sub(r1,r6,r1);
    goto newraph;

```

```

divide: divdpl(r2,r3,r4,r5,r6,r7,r8,r9,r10,ramb);
ret;

approx: acu(rom,aar) := pi10;
acu(rama,aar) := s34address;
acu(ramb,aar) := timeaddress;
mpy(rama,rom,0); (* s*pi in acr *)
bsr:=14; (* *n
mpy(r1,lsp,0) ac(rama,deca); (* ara --> v17 *)
bsr:=15 ac(rom,inca); (* arr --> 4 *)
r4:=lsp; (* s*pi*n in r4 *)
mpy(ramb,-rom,0) ac(rom,inca); (* acr := -4*t
(* arr --> r016 *)

bsr:=1;
mpy(lsp,rama,0) ac(ramb,inca); (* acr := -v*t*(2**16)
(* arb --> #frames *)
mpy(r1,rom,+acr) ac(rom,inca); (* acr + r0*n1
(* arr --> v_75 *)
mpy(r1,r4,+acr); (* acr + s*pi*n*n *)
mpy(ramb,-rom,+acr) ac(rom,inca); (* acr-v_75*#frames
(* arr --> halfr016*)
r3:=r3; (* nop in pipelined mode *)
r2:=msp; (* (r2::r3):=f(n1) *)
r3:=lsp;
add(r4,rom,r4); (* r4 + r0 *)

call divide;

ret;

exit: acu(ramb,aar):=cjump;
ramb:=r1;
r3:=r3;

end;

(* parameters if 8*t can be stored in one word
rom(40)=3217; (* pi *)
rom(41)=8; (* 8 *)
rom(42)=20598; (* r017 *)
rom(43)=2097; (* v_75 1,2 m/s *(2**17)/75 *)
rom(44)=10295; (* halfr017 *)
*)

rom(40)=1608; (* pi*(2**9) *)
rom(41)=4; (* 8 *)
rom(42)=10294; (* r016 2*pi*r0*(2**13) *)
rom(43)=1049; (* v_75 1,2 m/s *(2**16)/75 *)
rom(44)=5147; (* halfr016 r016/2 *)

```

program omega;

(*

This routine calculates the radial velocity of the compact disc corresponding to a certain track. It uses the trackvelocity and track pitch of that disc. The number of the track has to be stored in register r1 when this routine is invoked.

The formula used to determine omega is:

$$(\text{omega} * 4) = \frac{(v * (2^{14})) * (2^6)}{(s * (2^{34})) * (2^{(-2)}) * N * (2^{14}) + r0 * (2^{18})}$$

$$= \frac{4 * v}{N * s + r0}$$

So the result of the calculation is 4*omega and this value is stored in r6.

*)

```
(_ist
vaddress=12;          (* address of trackvelocity *)
saddress=13;         (* address of trackpitch *)
r0=80;               (* address of radius of first track *)
```

```
libmacro divdpl;
```

```
begin
```

```
acu(rama,aar):-12;          (* initialisation *)
acu(rom,m):-#ffff;
acu(rama,m):-#ffff;
rama:=19660 ac(rama,inca);  (* v*(2**14)=1,2*(2**14) m/s *)
rama:=29205;               (* s*(2**34)=1,7*(2**34) mm/s *)
acu(rom,aar):-80;

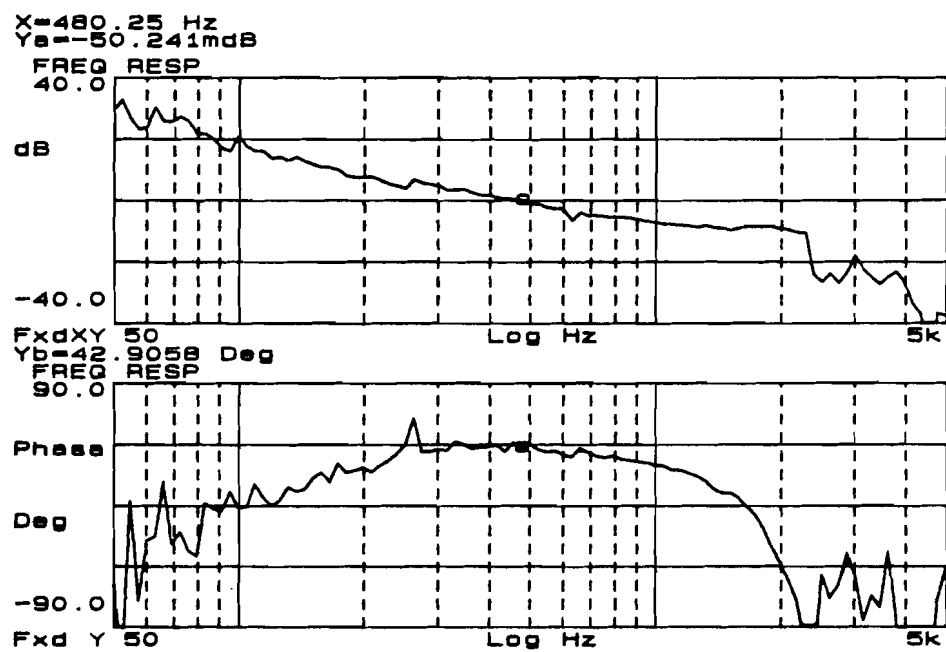
(* ara --> s*(2**34) *)
(* arr --> r0 *)
r2:=rama ac(rama,deca);    (* r2:= track pitch -s*(2**34)- *)
r3:=r3;                   (* nop in pipelined mode *)
asr(r2,r2);               (* r2:=s*(2**34)/2 *)
asr(r2,r2);               (* r2:=s*(2**34)/4 *)
mpy(r1,r2,0);             (* acr:= s*(2**34)*N/4 *)
bsr:=15;
r4:=lsp;
add(r4,rom,r4) ac(rom,inca);
(* r4:= s*N*(2**18) *)
(* r4:= s*N*(2**18) + r0*(2**18)
arr --> 2**6 *)
(* nop *)
(* acr:=v*(2**14)*64 *)

(* (r2::r3):= v*(2**14)*64 *)
(* r6:=omega*4 *)
eop: goto eop;
```

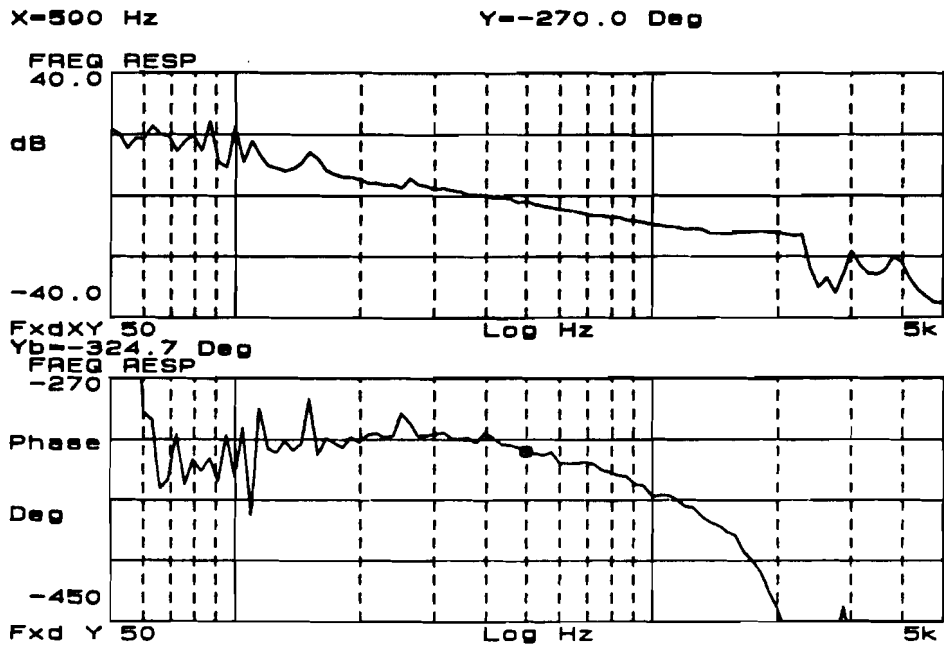
```
divide:                                (* double precision divide  
                                       (r5::r6):=(r2::r3)/r4 *)  
    divdpl(r2,r3,r4,r5,r6,r7,r8,r9,r10,ramb);  
    ret;  
  
end;  
  
rom(80)=6554;                          (* r0*(2**18) *)  
rom(81)=64;                             (* 2**6 *)
```

Appendix I

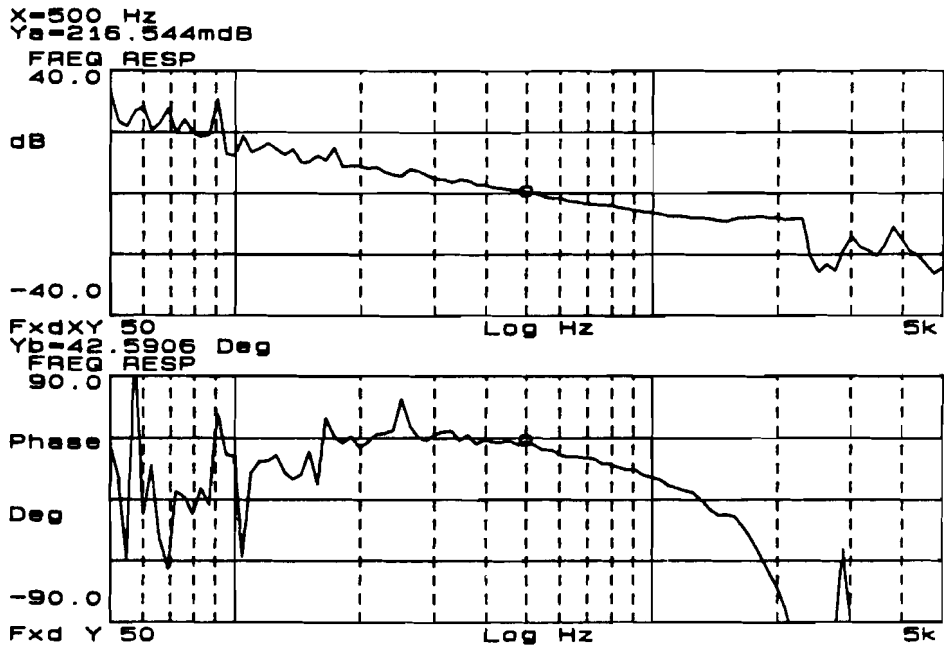
Resultaten en metingen.



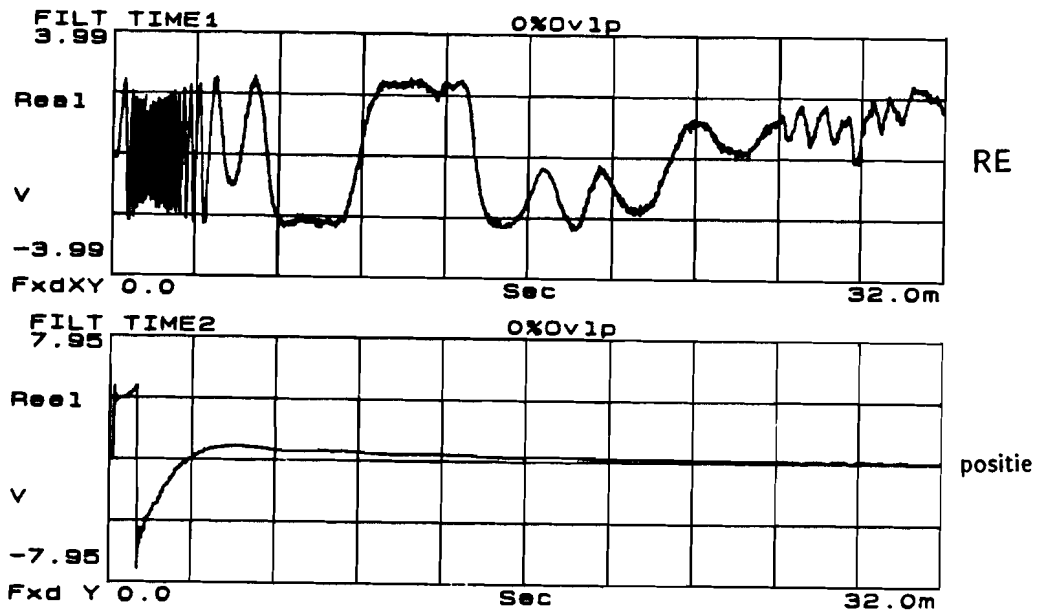
Figuur I.1: Overdracht van radiele regeling in CM100 speler.



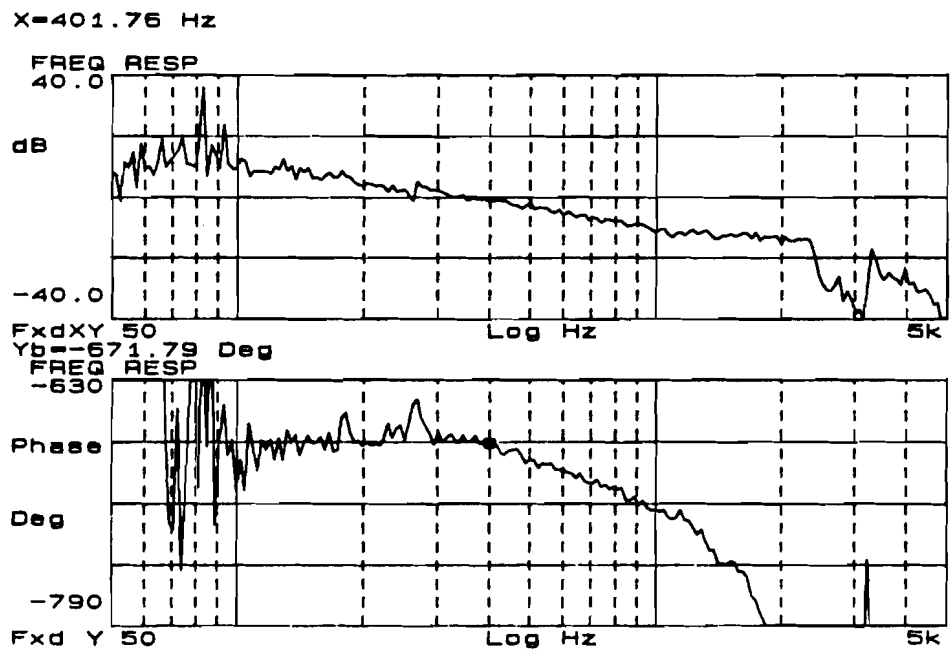
Figuur I.2: Overdracht open loop-systeem met berekende regelaar.



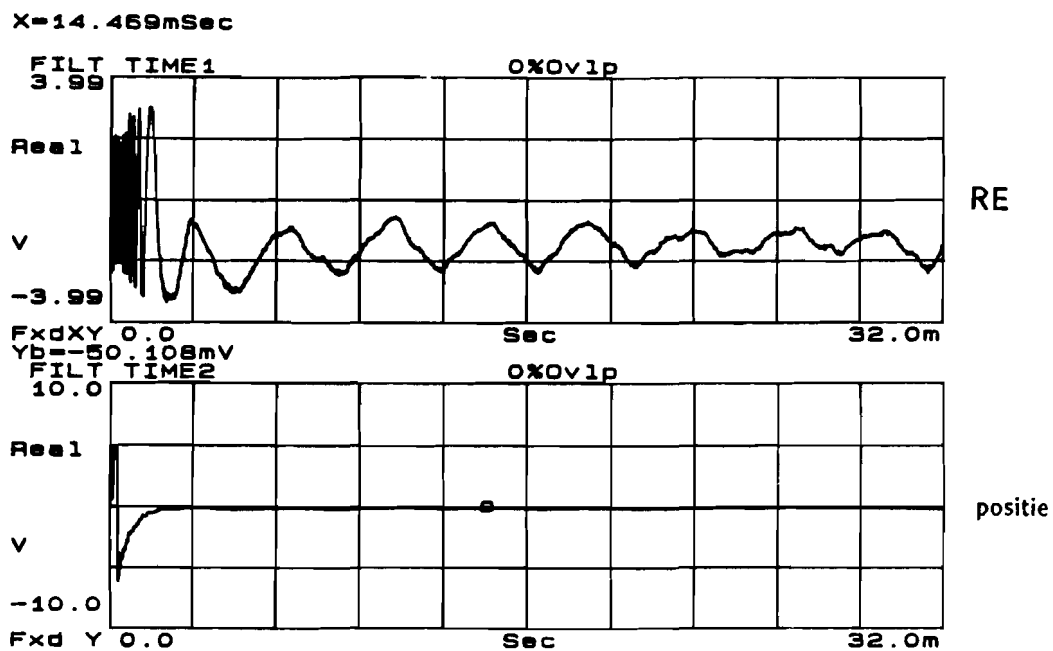
Figuur I.3: Overdracht geoptimaliseerde radiale regelaar.



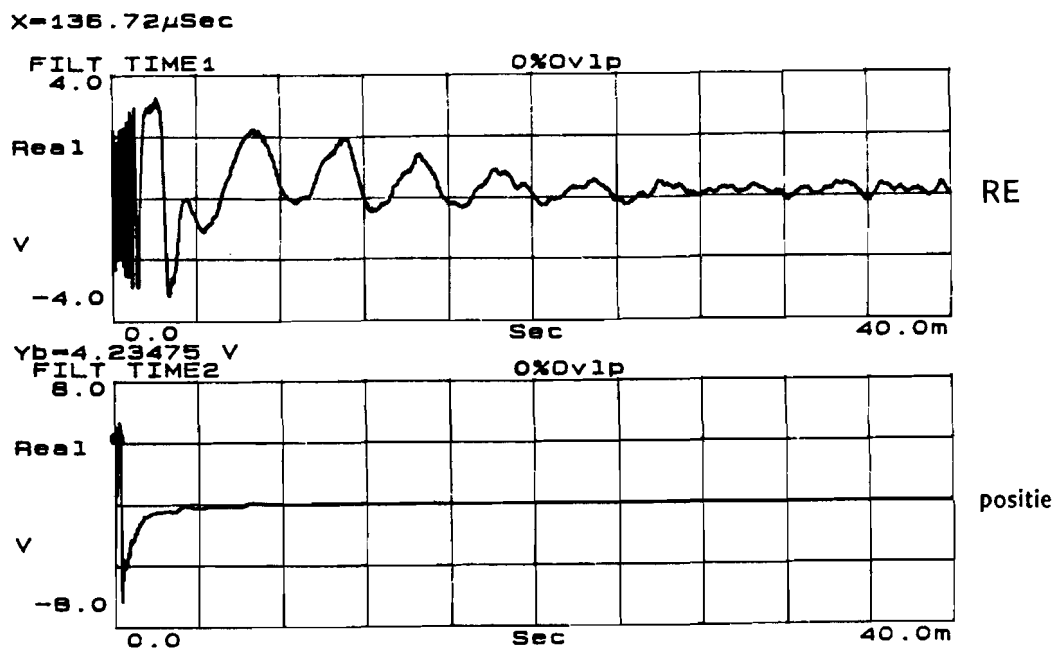
Figuur I.4: Stapresponsie van berekende regeling.



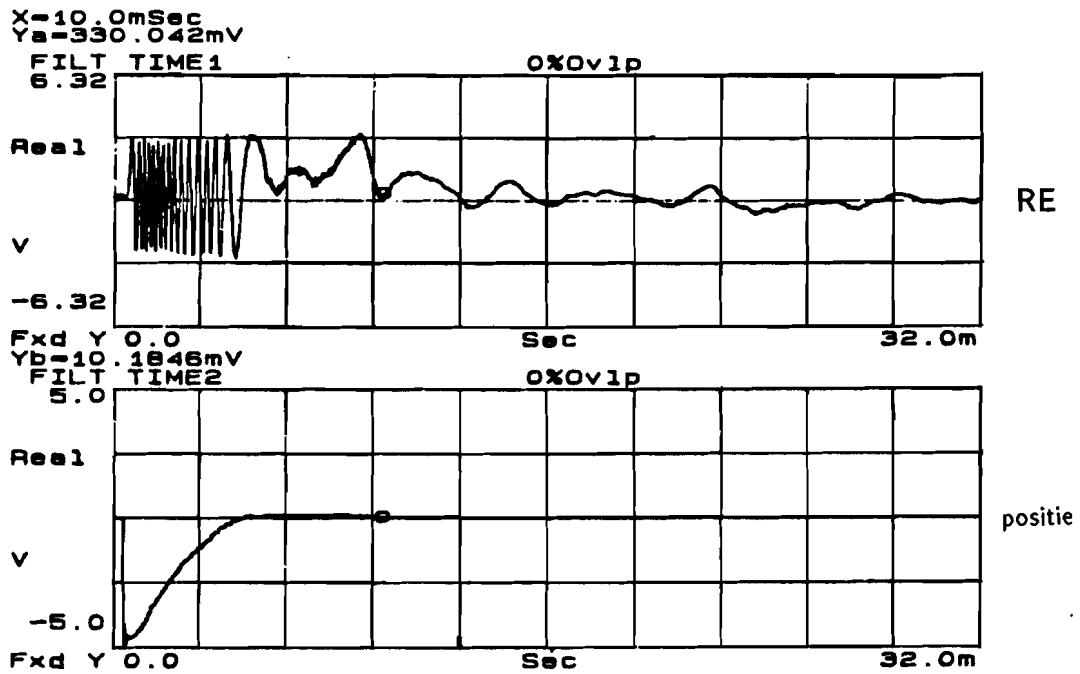
Figuur I.5: Overdracht van regeling geoptimaliseerd voor sporsprongen.



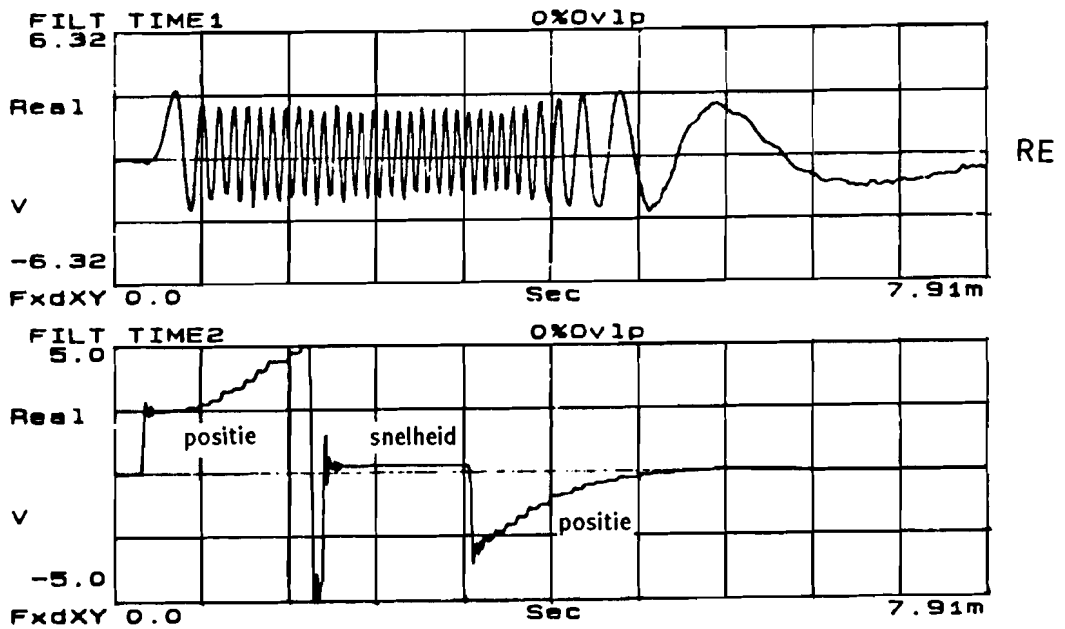
Figuur I.6: Afremmen van arm met geoptimaliseerde regeling voor sprongen.



Figuur I.7: Afremgedrag van geoptimaliseerde sprongregeling.



Figuur I.8: Sprong bij spronglengte kleiner dan N_{ref}



Figuur I.9: Sprong bij spronglengte groter dan N_{ref}

lengte in sporen	tijd in ms	tijd per spoor in μs
1	0,8	800
2	2,5	1250
3	3	1000
5	4	800
10	4,5	450
20	7,2	360
30	8	267
50	10	200
100	15	150
200	25	125
300	35	117
500	45	90
1000	65	65
2000	100	50
3000	130	43
5000	200	40
10000	370	37
15000	540	36

Tabel I.1: Sprongtijd als functie van spronglengte.