Optimalisatie met behulp van een hybride rekenmachine

Hermeling, G.M.W.

Award date:
1969

Optimalisatie met behulp van een hybride rekenmachine.

G.M.W. Hermeling.

Verslag van mijn afstudeerwerk in de groep Meten en Regelen van de Afdeling der Technische Natuurkunde, onder supervisie van Prof. ir. O. Rademaker en Ir. G. Dekker.
November 1968.

Inhoudsopgave.

Lijst van afkortingen en symbolen.

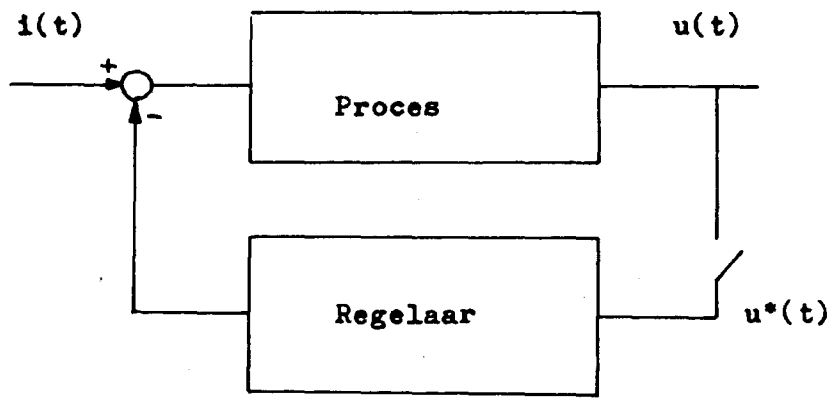
δ	looptijd
τ_1	tijdconstante
τ_2	tijdconstante
T	monstertijd
PID regelaar	proportionele, integrerende en differentierende regelaar
$\sum_{0}^{\infty} \lvert u(nT) \rvert$	somcriterium
$\int_{0}^{\infty} \lvert u(t) \rvert \, dt$	integraal criterium
A R.	analoge rekenmachine
D.R.	digitale rekenmachine
AD-omzetter	analoog naar digitaal omzetter
ADO	
DA-omzetter	digitaal naar analoog omzetter
DAO	
S.J.	sompunt van de ingangsweerstanden van een versterker.
I.C.	Initial Control
μ F	micro Farad
M	mantisse
N	macht van 2
m	aantal monsters dat tijdens de responsieduur wordt genomen.
oct.	octaal

I. Inleiding.

1. 1 Intermitterende regeling.

Door de opkomst van digitale technieken gaat de belangstelling steeds meer uit naar intermitterende regelingen.

Het intermitterende regelsysteem dat bestudeerd is kunnen we in zijn algemene gedaante als volgt weergeven:



Een storing i(t) gaat het systeem in en de bedoeling van de regeling is een bepaald criterium minimaal te maken.

De essentie van intermitterend regelen is, dat niet het gehele foutsignaal u(t), doch slechts monsters $u^*(t)$ voor de regeling worden gebruikt.

Bij digitale regeling hebben we bovendien nog te maken met kwantisatie d.w.z. dat de gebruikte signalen slechts een aantal discrete waarden kunnen aannemen.

1. 2 Nadere omschrijving van het intermitterend systeem.

Een groot aantal processen in de industrie kunnen we vangen onder een noemer n.l. processen met de overdrachtsfunctie:

$$P(s) = \frac{e^{-\delta s}}{(1+\tau_1 s)\,(1+\tau_2 s)}$$

(Een tweede orde proces met looptijd).

Voor de regeling van processen wordt de P.I.D.-regelaar veel toegepast.

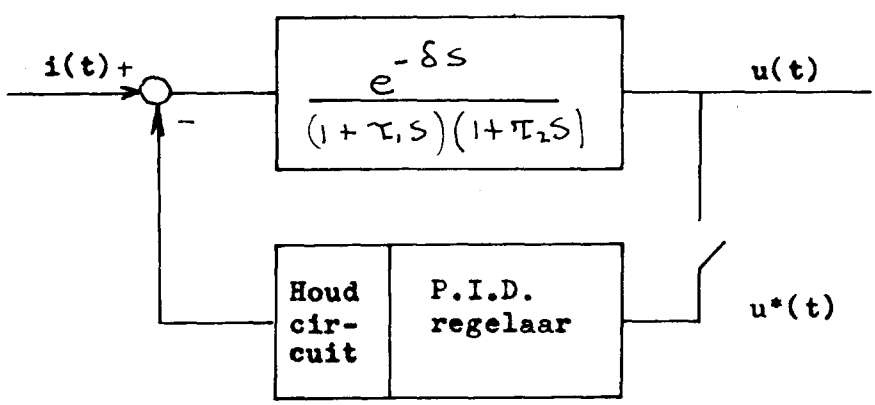
Bij ons onderzoek van intermitterende systemen maken wij gebruik van de intermitterende P.I.D.-regelaar, gevolgd door een houd-circuit dat het uitgangssignaal van de regelaar gedurende de monsterperiode T constant houdt.

Als criterium voor de kwaliteit van de regeling nemen we of het integraal criterium $\int_0^\infty |u(t)|\,dt$ of

het somcriterium $\sum_0^\infty |u(nT)|$.

De instelling van de regelaar waarbij dit criterium minimaal is noemen we de optimale regelaarinstellingen.

Het systeem dat we willen bestuderen heeft dus de volgende gedaante:



1.3 Doel van het onderzoek.

In onze groep werd aan bovengenoemd systeem o.a. bestudeerd

1) Invloed van de storingsvorm
2) Invloed van δ, τ_1 en τ_2
3) Invloed van de monsterperiode
4) Invloed van andere criteria
5) Invloed van het moment en de plaats van de storing.

Deze studies werden zowel op analoge als op digitale rekenmachines verricht.

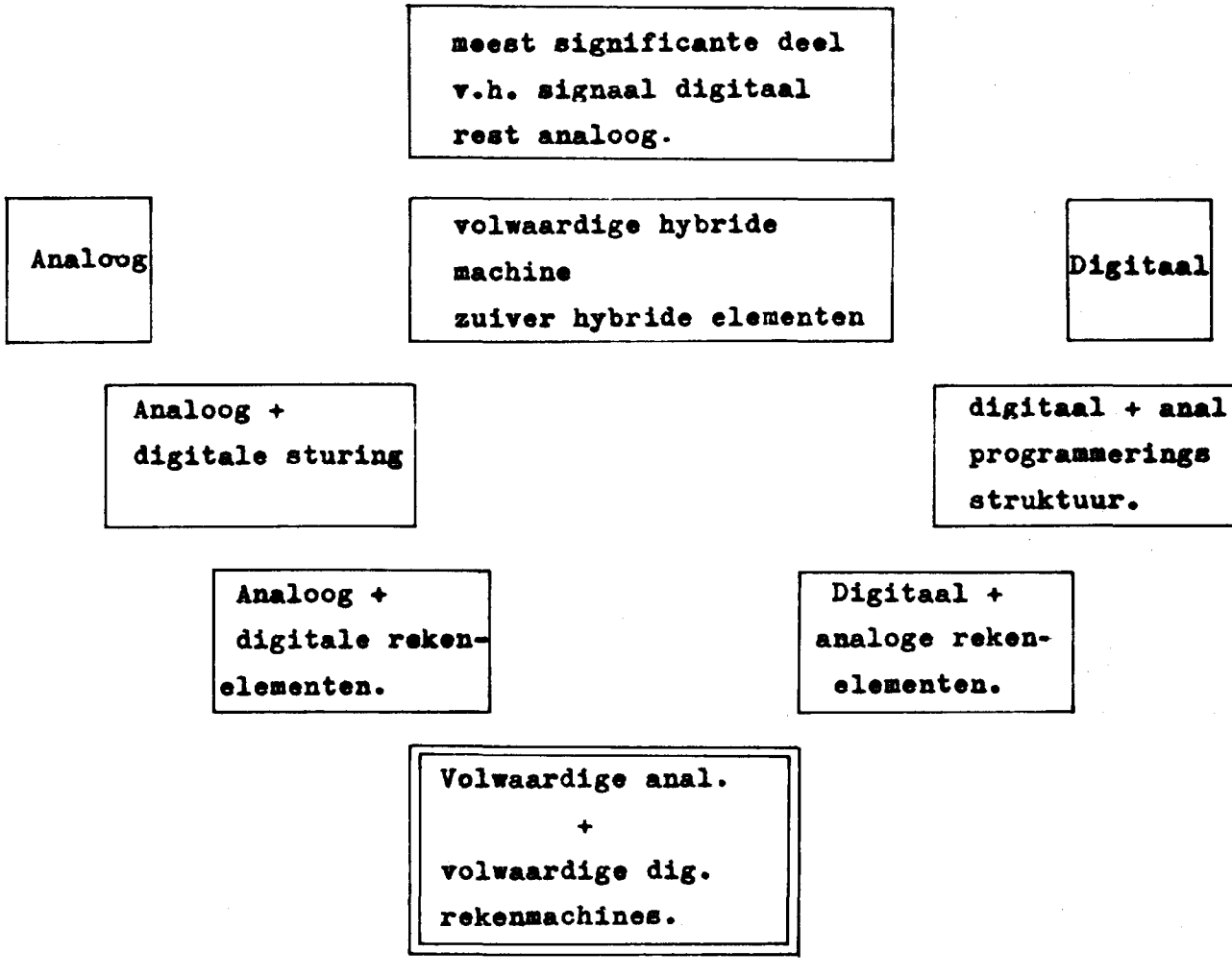
Beide machines hadden hun voor- en nadelen. Daarom werd gedacht aan de mogelijkheid om een A.R. en een D.R. zodanig te combineren tot een hybride rekenmachine, dat elke machine die taken kreeg uit te voeren waarin hij superieur was.

Het doel van mijn afstudeerwerk was een hybride systeem te realiseren zodanig dat de optimale regelaarinstellingen bepaald konden worden.

II. Hybride rekentechnieken.

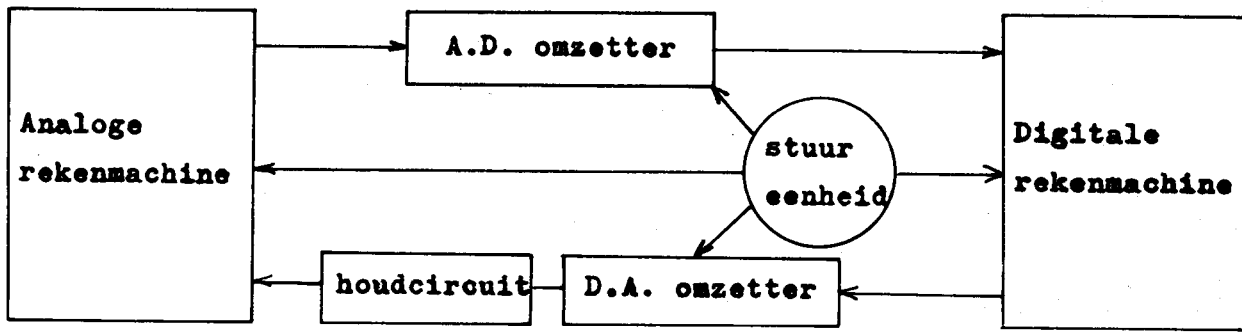
2.1 Inleiding.

Onder hybride technieken verstaan wij de combinatie van analoge en digitale technieken. De mate waarin analoge en digitale elementen tot een hybride machine samengevoegd kunnen worden is hieronder in een spectrum van elektronische rekenmachines weergegeven.



Onze interesses gaan hierbij uit naar de koppeling van een volwaardige analoge rekenmachine met een volwaardige digitale rekenmachine.

In het volgende schema wordt deze koppeling geillustreerd:



Achtereenvolgens zal worden ingegaan op de A.R., de D.R. en de koppelapparatuur.

2.2 De analoge rekenmachine.

Van belang zijn vooral die eigenschappen van de A.R. die bij de D.R. in mindere mate aanwezig zijn.

Deze eigenschappen zijn:

1. De A.R. kan integreren naar de tijd zonder gebruik te maken van ingewikkelde wiskundige benaderingsformules.

2. De A.R. kan snel complexe systemen van vergelijkingen oplossen, waarbij de maximale complexheid wordt bepaald door de grootte van de machine.

3. De A.R. levert direct grafieken en geeft bovendien meteen inzicht in het effect van de verandering van een parameter.

4. De A.R. is bijzonder geschikt voor het simuleren van continue regelsystemen.

2.3 De digitale rekenmachine.

De eigenschappen van de D.R. die voor ons van belang zijn en meestal duiden op een ongunstig alternatief bij de A.R. zijn de volgende:

1. De complexheid van de op te lossen problemen is in mindere mate afhankelijk van de grootte van de machine, de onnauwkeurigheid wordt bepaald door de woordlengte en wordt dus veelal bepaald door de grootte van de machine en de rekentijd.

2. De maxima en minima van variabelen mogen ver uit elkaar liggen.

3. De D.R. leent zich in het bijzonder voor systemen of berekeingen waarin logische beslissingen een belangrijke rol spelen.

4. Daar de D.R. geheugen-elementen bezit is deze machine juist

geschikt voor opslag van tussenresultaten, functiegeneratie van meerdere variabelen etc.

5. De resultaten verkregen door een berekening op de D.R. zijn meestal beter reproduceerbaar.

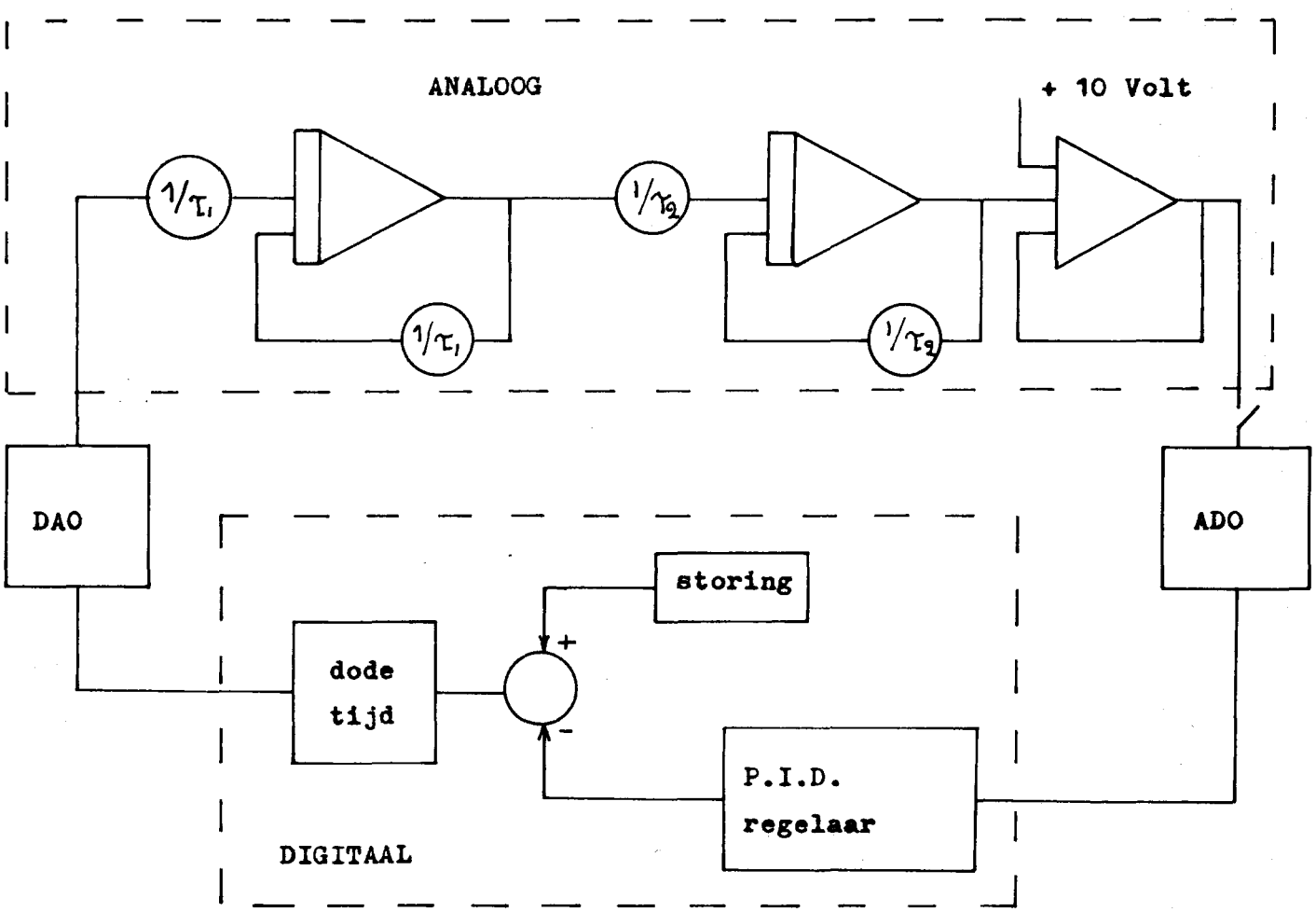
2.4 De koppelapparatuur.

Nadelen van de koppelapparatuur zijn de kwantisatie en de omzettingstijd.

Echter bij de koppeling van twee machines met een zeer verschillend karakter kunnen we niet zonder deze apparatuur.

2.5 De realisering van het hybride systeem.

Indien wij het geregeld tweede orde proces met looptijd op de hybride machine willen verwezenlijken, zullen we moeten trachten de systeemfuncties die bij uitstek vervuld kunnen worden door een bepaalde machine, ook op die machine onder te brengen en komen zo tot de volgende configuratie.



Het tweede orde proces is erg geschikt om analoog gesimuleerd te worden (tweede orde differentiaalvergelijking), terwijl de dode tijd weer het beste op de digitale gesimuleerd kan worden (tabel manipulatie).

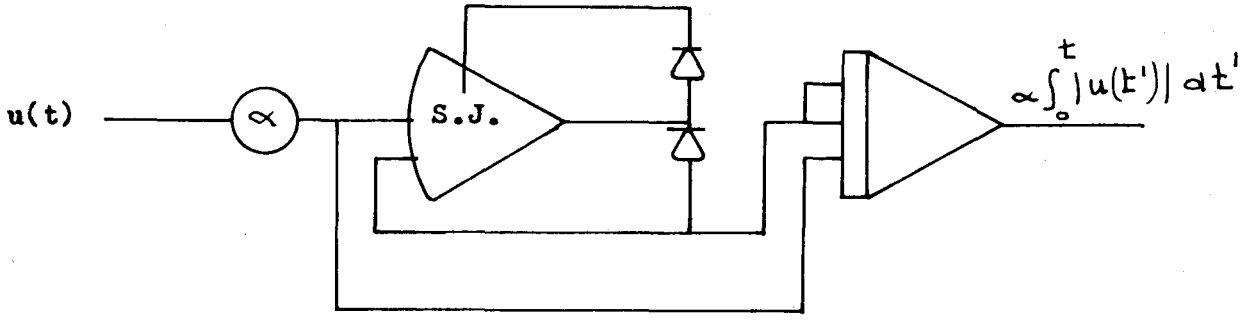
Als monsternemer gebruiken we de door de digitale klok gestuurde ADO.

De P.I en D-acties worden voorgesteld door getallen in de digitale computer en kunnen daarom gemakkelijk gevarieerd worden.

De storing wordt ook digitaal gegenereerd. Daardoor is het mogelijk de invloed van verschillende storingsvormen op de optimale regeling te bestuderen.

Een functie die weer bij uitstek door de A.R. vervuld kan worden, is de bepaling van het integraalcriterium $\int_0^t |u(t')|\,dt'$

Dat gebeurt met behulp van de volgende schakeling:



III De programma's.

3.1 De aard van de programma's.

Het digitale programma bestaat uit 2 gedeelten:

1. Het simulatieprogramma, dat ervoor zorgt dat we een proces met constante parameters kunnen simuleren en regelen met een PID-regelaar

2. Het optimaliseringsprogramma, dat dient om op grond van eerder verkregen resultaten nieuwe regelacties te bepalen.

Het optimaliseringsprogramma is het eigenlijke hoofdprogramma. Van dit programma wordt iedere keer gesprongen naar het simulatie programma.

Het optimaliseringsprogramma wordt onder 3.2 en 3.3 besproken .

Het simulatieprogramma kunnen we beschouwen als een interrupt subroutine die bij iedere nieuwe regelaarinstelling m maal wordt aangeroepen; en dan de nieuwe criteriumwaarde geeft. Het simulatie-programma wordt onder 3.4 besproken.

Wat de techniek van het optimaliseren betreft zijn we te werk gegaan volgens een trial and error methode , waarvan op grond van studies in onze groep is gebleken dat hij goed voldeed.

3.2 De gedeeltelijke optimalisering van een bepaalde regelparameter.

Bij de optimalisering, veranderen we, uitgaande van stabiele startwaarden voor de I, P en D-actie, achtereenvolgens deze acties. De manier waarop we één bepaalde actie gaan variëren is weergegeven in stromingsschema 1 en zal hier kort toegelicht worden. (Deze sub-routine noemen we subroutine onbeperkt). De actie die we gaan variëren heeft al een bepaalde waarde SO (Semi Optimum). Deze waarde is de startwaarde of de tot dan toe optimale waarde van die bepaalde actie. Vervolgens wordt de actie groter gemaakt met een bedrag Δ

We gaan nu het proces simuleren met de 3 regelparameters waarvan er dus een met Δ veranderd is. Het resultaat van deze verandering wordt ons medegedeeld in de vorm van verandering van het criterium.

Is het criterium beter dan zijn we kennelijk op de goede weg en verhogen we S.O. weer met Δ , maar niet alvorens we Posi met 1 verhoogd hebben. Posi is een vlag die gezet wordt indien verhogen van de regelactie succes heeft.

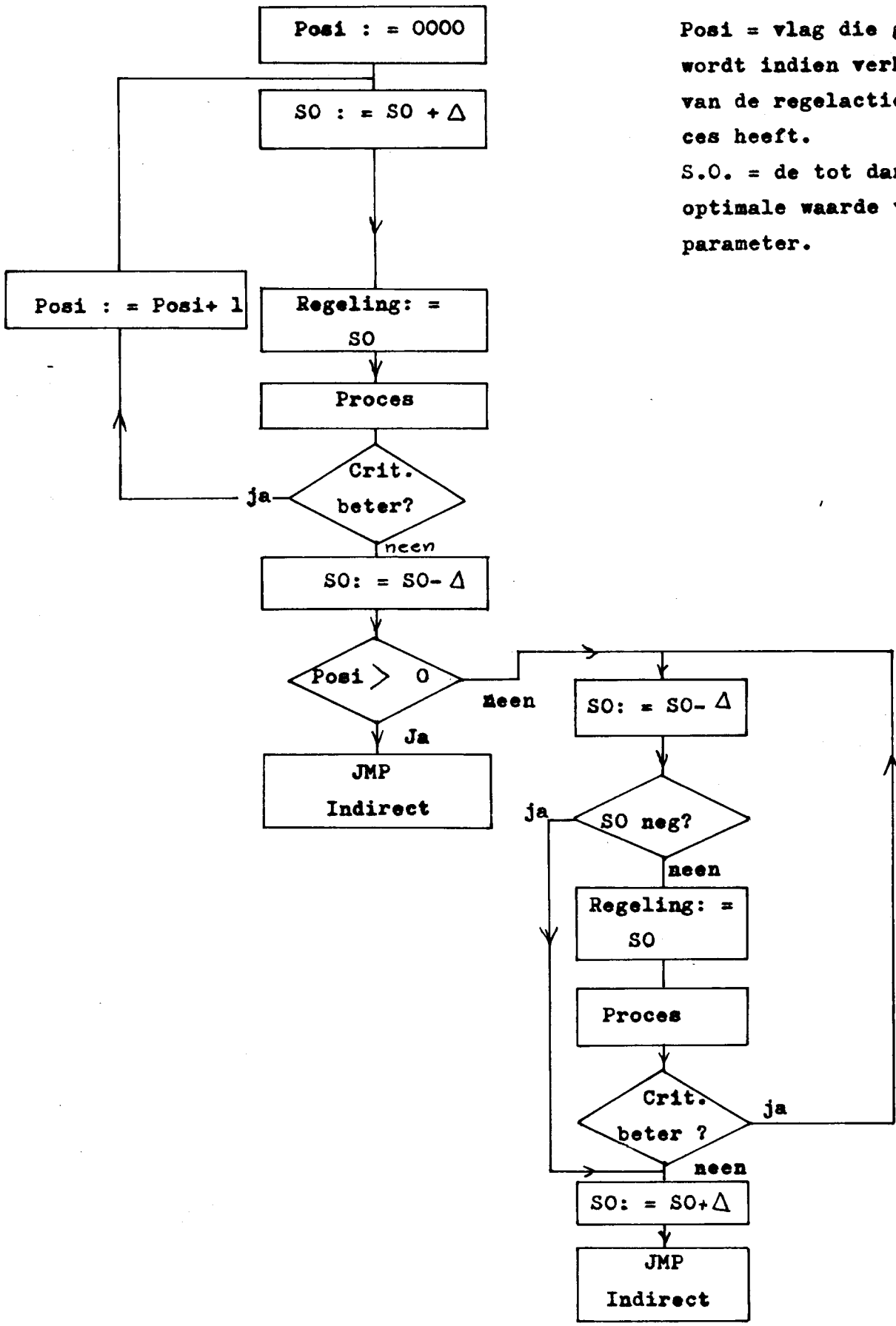
Op een gegeven ogenblik, als we voorbij het optimum zijn, zal het criterium slechter worden. S.O. wordt met Δ verlaagd om de tot dan toe beste parameter weer terug te krijgen.

Om te testen of vergroting van de regelparameter al of niet succesvol is geweest, gaan we kijken of de vlag Posi gezet is. Zo ja, dan zijn we op de goede weg geweest en hoeven we dus niet verder te kijken naar kleinere waarden van S.O. Zo nee, dan moeten we S.O. dus verkleinen en was vergroting foutief. We maken S.O. : = S.O. - Δ en beschouwen dit als regelparameter, wordt het criterium beter dan maken we S.O. nog kleiner totdat we slechter uitkomen. We zijn dan een stap te ver gegaan en moeten die terug zetten.

Op deze manieren vinden we de meest optimale instelling van de gevarieerde parameter, die hoort bij 2 bepaalde andere regelparameters.

SUBROUTINE ONBEPERKT.

Stromingsschema 1



Posi = vlag die gezet wordt indien verhogen van de regelactie succes heeft.
S.O. = de tot dan toe optimale waarde van de parameter.

3.3 Het totaalbeeld van de optimalisering.

(zie stromingsschema 2).

De verschillende regelacties worden opgeslagen in 2 registers van 12 bits. In het ene register staat M, de mantisse en in het andere register staat N, de macht. We kunnen een regelparameter dus als volgt schrijven:

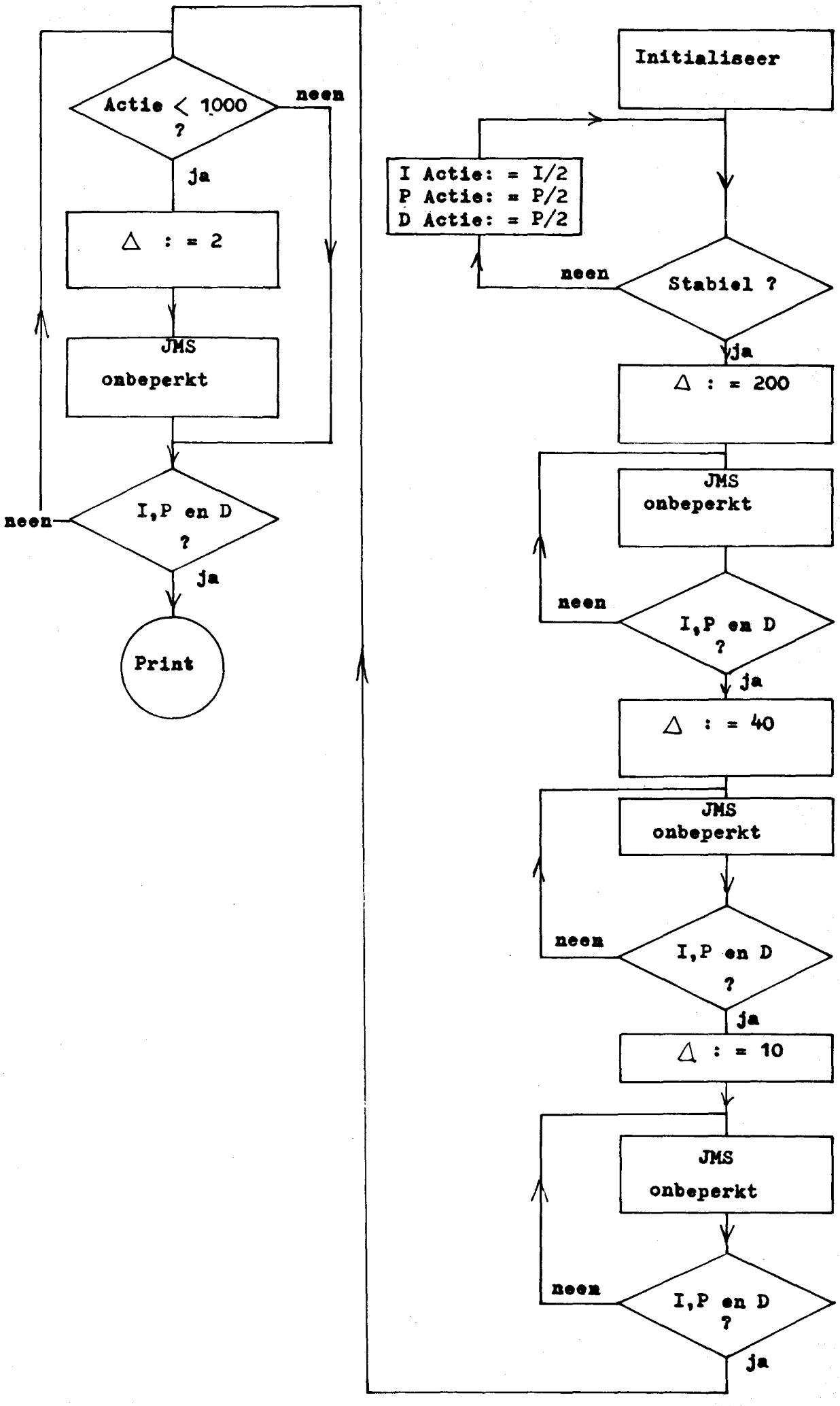
$$\text{Actie} = M \times 2^{N}$$

Als startwaarde voor de diverse regelacties nemen we steeds M = 2000 oct. terwijl we N vrij kunnen kiezen.

We gaan nu het systeem met deze startwaarden simuleren. Is dit systeem instabiel dan worden de startwaarden voor alle acties een factor 2 verkleind, net zolang tot het stabiel is geworden.

Vervolgens gaan we de I-actie variëren met $\Delta = 200$, dan de P-actie en tot slot de D-actie. Daarna maken we Δ = 40 oct. en varieren in dezelfde volgorde de 3 acties en herhalen dit procedé voor Δ = 10 oct. Indien we tot dan toe optimale actie hebben gevonden wordt er getest of er een of meerdere acties bij zijn die kleiner dan 1000 zijn geworden. Deze worden dan nog met Δ = 2 gevarieerd, waarna de resultaten worden uitgetypt.

Stromingsschema 2

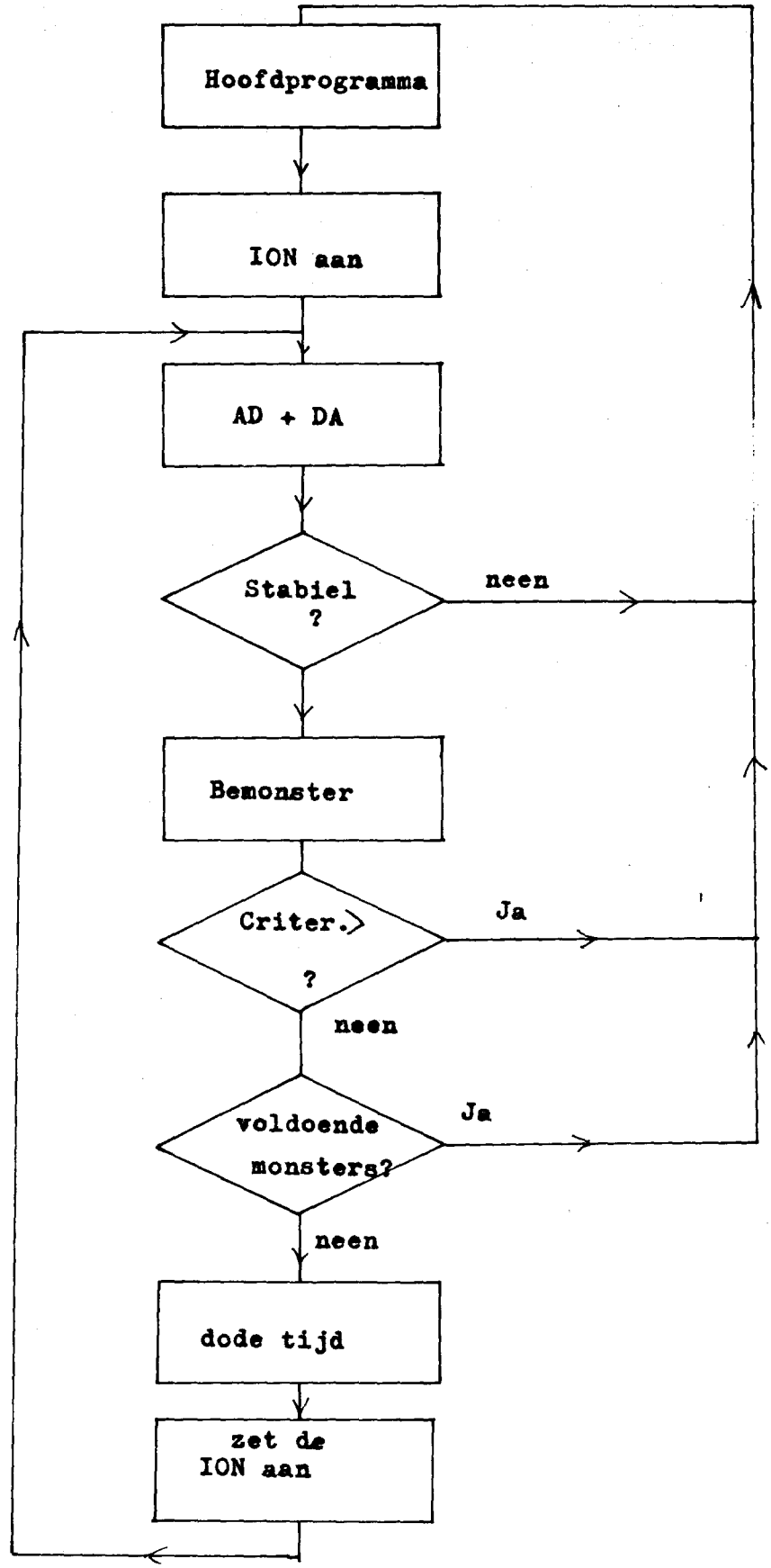


3.4 Het simulatieprogramma voor een systeem met constante regelparameters.

Het simulatieprogramma dient om te zorgen dat we het systeem met een bepaalde PR-combinatie, gedurende een van te voren vastgestelde tijd kunnen simuleren. Deze tijd (m x de monsterperiode T) is zo lang dat voor alle regelaarinstellingen die we gedurende een optimalisatie bekijken de responsie van het systeem op de storing voldoende lang nul is.

Het simulatieprogramma wordt gestart door middel van een klokpuls. Indien we met het hoofdprogramma bezig zijn hebben de klokpulsen geen interrupt tot gevolg (IOF : de interrupt-faciliteit staat af). Met behulp van de instructie ION (interrupt faciliteit staat aan) kunnen we de interrupt faciliteit aanzetten, waardoor een klokpuls het procesprogramma doet starten. Aan het eind van het simulatieprogramma wachten we op de volgende klokpuls. Dit herhalen we tot we m interrupts gehad hebben. Dan springen we terug naar het hoofdprogramma.

Het simulatieprogramma (dat is weergegeven in stromingsschema 3) start met het afgeven aan de A.R. van het signaal dat uit de dode tijd komt en met het bepalen van u*(t) met behulp van een ADO.
Nu volt een test of het systeem stabiel is, d.w.z. of het signaal u*(t) binnen de grenzen $-9\ V < u^*(t) < +9\ V$ ligt. Dit om te testen of er in de A.R. geen overflow gaat optreden. Vervolgens wordt bemonsterd d.w.z. het uitgangssignaal van de digitale regelaar wordt bepaald. Bovendien testen we of het criterium behorende bij de momentane PR-combinatie groter is geworden dan het tot dan toe optimale criterium. Is dit het geval dan wordt verdere simulatie van die PR -combinatie afgebroken.
Hierna volgt er een test of er al m monsters genomen zijn. Zo nee, dan gaat de uitkomst van storing minus regelaaruitgang naar de dode tijd subroutine, waarna de ION wordt aangezet, zodat er weer interrupten mogelijk zijn.
Zo ja, dan wordt terug gesprongen naar het hoofdprogramma.



ION Interrupt faciliteit staat aan

Het simulatieprogramma

Stromingsschema 3

IV Waarnemingen en conclusies.

4. 1 Waarnemingen.

Om te kijken of het hybride systeem betrouwbare resultaten leverde, heb ik het proces met $\tau_1 = \tau_2 = \delta = 1$ bestudeerd. Bij een 11-tal waarden van T, de monstertijd.
De waarden van T/δ die we onderzoeken zijn 1/8, ¼, ½, 1, 2, 3, 4, 5, 6, 7 en 8.
Dit is het meest interessante gebied, want kleinere waarden van doet de regeling steeds meer op continue regeling gelijken en de verbetering in criteriumwaarde die we nog kunnen bereiken is niet noemenswaard, terwijl voor grotere waarden van T/δ de optimale acties evenredig met T/δ veranderen, hetgeen ook niet meer interessant is. We hebben gewerkt met 2 verschillende criteria, het somcriterium $\sum_{i=1}^{n} |u(iT)|$ waarbij u(iT) de fout is ten tijde van de i^e bemonstering.
Verder hebben we ook het integraal criterium $\int_0^T |u(t)|\,dt$ berekend.
Uit de grafieken 1 en 2 blijkt dat er nauwelijks verschil bestaat tussen de optimale acties verkregen met het som- en met het integraalcriterium. Echter in de praktijk bleek het integraalcriterium beter reproduceerbare resultaten te geven en beter te convergeren.
De I-actie is genormeerd op δ/T, de D-actie op T/δ en het somcriterium op T/δ.
Zie voor deze normering het stageverslag van Cuypers. (Litt no. 1)

Gemiddeld duurde het 4 seconden om de optimale regelacties te bepalen. Het printen van de resultaten duurde dan nog 20 seconden.

4. 2 Conclusies.

Een van de meest opvallende verschijnselen bij de waarnemingen, is het toenemen van het verschil tussen het integraal en somcriterium naarmate T/δ kleiner wordt (grafiek 1).

Hiervoor heb ik nog geen afdoende verklaring kunnen vinden.

De optimale regelaarinstellingen bepaald met het somcriterium en het integraal criterium bleken niet veel van elkaar te verschillen.

In de grafieken 3 en 4 zijn met elkaar vergeleken de uitkomsten verkregen met het hybride systeem en de uitkomsten die door volledig digitale simulatie zijn verkregen.

Uit grafiek 3 kan geconcludeerd worden dat de uitkomsten een grote mate van overeenkomst vertonen, zowel wat de I-actie als wat het criterium betreft.

In grafiek 4 zijn vergeleken de P en D-acties die met beide methoden verkregen zijn. Hier was de overeenkomst iets minder groot.

Als algemene conclusie kunnen we stellen dat de hybride rekenmachine zoals we reeds verwachtten, een waardevol hulpmiddel is gebleken bij het bepalen van de optimale regelaarinstellingen.

V. De apparatuur die bij de koppeling werd gebruikt.

5.1 De analoge rekenmachine.

De bij de koppeling gebruikte A.R. RAT 741 heeft 23 rekenversterkers, waarvan er 8 als integrator te gebruiken zijn. De machine beschikt bovendien over 20 potentiometers en over 8 vermenigvuldigers.

Als maat voor de snelheid van deze machine vermelden wij de mogelijke tijdconstanten van de integratoren. Deze zijn in te stellen op 1, 1/10 en 1/100 seconde.

De referentiespanning (machineeenheid) van de RAT 741 bedraagt 10 Volt.

5.2 De digitale rekenmachine.

Voor de digitale kant van het systeem werd gebruik gemaakt van de digitale rekenmachine PDP-8. Dit is een kleine processcomputer die beschikt over een kerngeheugen van 4096 woorden (4 k) van elk 12 bit. De cyclustijd van deze machine bedraagt 1,5 μ. sec, hetgeen neerkomt op een instructietijd (in machinetaal) van gemiddeld 3 μ sec.

5.3 De koppelapparatuur.

Voor de koppeling konden we beschikken over een AD-omzetter, die in 25 μ sec. van een analoog signaal een digitaal signaal van 12 bits maakte . Over een DA-omzetter, die 10 μ sec. nodig had om van een digitaal signaal (van 12 bits) een analoog signaal te maken.

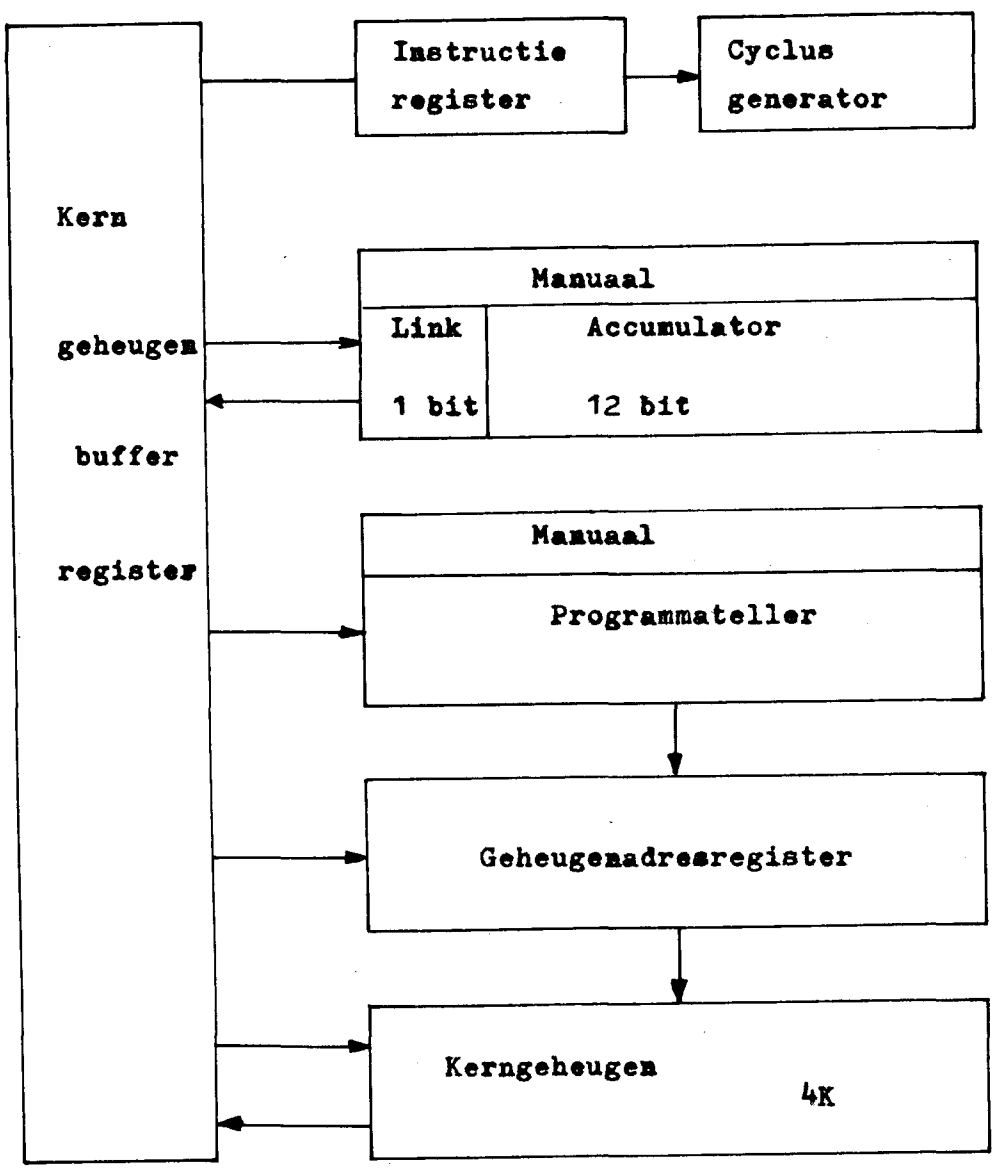
Voor de bemonstering hadden we een digitale klok tot onze beschikking die onafhankelijk van de analoge of digitale rekenmachine pulsen geeft. De tijd tussen 2 pulsen is trapsgewijze instelbaar van 0,1 msec. tot 9.999,9 msec. in stappen van 0,1 msec.

VI. De apparatuur die ten behoeve van de koppeling gebouwd werd.

6.1 Uitbreiding van de digitale rekenmachine.

6.1.1 Inleiding.

De D.R. beschikt over diverse registers die alle bepaalde functies verrichten.



Hieronder volgt een omschrijving van de functies van de diverse onderdelen:

a) Accumulator.
Het eigenlijke rekenorgaan van de machine. Via de accumulator vindt normaliter de input en output van gegevens plaats.

b) Link.
Men noemt de link ook wel het carryregister. De link dient voornamelijk ter detectie van overflow.

c) Programmateller.
Deze bepaalt de volgorde waarin de instructies doorlopen worden. De programmateller bevat het adres van de volgende instructie.

d) Geheugenadresregister.
Bevat het adres van de geheugenplaats die wordt in-of uitgelezen.

e) Manuaal.
Biedt de mogelijkheid om met de hand gegevens in de D.R. te lezen.

f) Kerngeheugen.
Opslagplaats van instructies en gegevens. Totaal 4096 woorden.

g) Geheugenbufferregister.
Alleen via dit buffer kan de uitwisseling van gegevens tussen het kerngeheugen en de rest van de apparatuur plaats vinden.

h) Instructieregister.
Een 3-bits register dat de binaire code van de instructie bevat.

i) Cyclusgenerator.
Deze bepaalt de organisatie in de tijd.

6.1.2 De indirecte methode van getaltransport.

Het kerngeheugen dient als opslagplaats van instructies en gegevens. D normale gang van zaken bij het inlezen is als volgt:

De ponsbandlezer is verbonden met de accumulator. Vanuit de accumulator gaan de gegevens via het kerngeheugenbuffer naar het kerngeheugen, alwaar ze opgeslagen worden.
Ook de AD-omzetter en de DA-omzetter zijn verbonden met de accumulator. Het getaltransport tussen kerngeheugen en uitwendige apparatuur vindt dus steeds indirect, d.w.z. via de accumulator plaats. Dit is nadelig, want de accumulator is het centrale rekenorgaan van de D.R.

Indien er een interrupt komt die zegt dat we de ADO moeten inlezen of de DAO moeten uitlezen, zullen we dus steeds de inhoud van de accumulator moeten redden, daarna de beschikbare gegevens in-of uitlezen en tot slot weer de oorspronkelijke toestand herstellen.

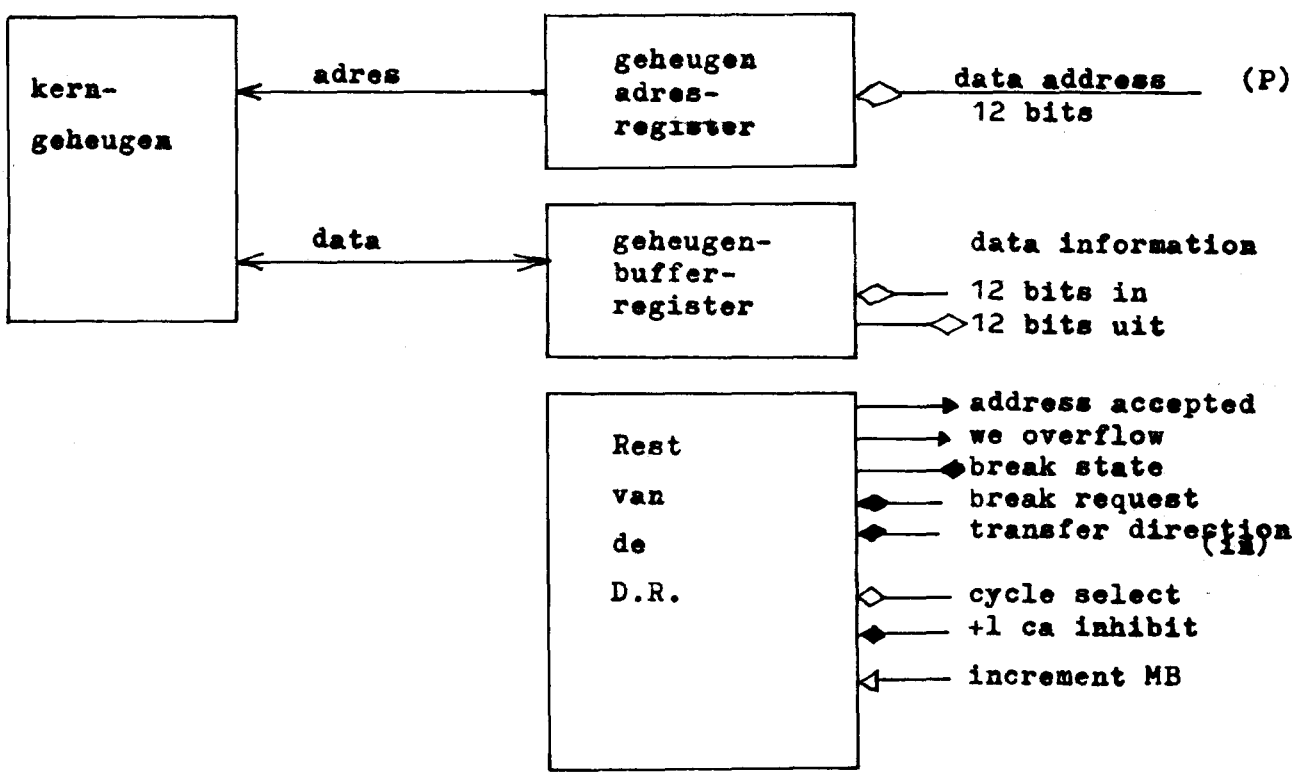
Gelukkig biedt de PDP-8 een mogelijkheid om in-en uitlezen te versnellen, n.l. data break.

6.1.3 De directe methode van getaltransport.

De essentie van data break is de volgende:

Het lopende programma wordt gedurende 3 cycli stopgezet en zonder dat de inhoud van enig register aangetast wordt, vindt er getaltransport plaats, waarna het lopende programma weer normaal verder uitgevoerd wordt.

Om een breakcyclus te genereren moet de D.R. diverse signalen toegediend krijgen, (ingaande pijlen) terwijl hij zelf ook verschillende signalen afgeeft (uitgaande pijlen). Dit is in onderstaand schema geschetst.



Bij de data break worden de ADO en DAO rechtstreeks aan het kerngeheugenbuffer gekoppeld. Indien wij dan een Breakverzoek indienen, zorgt de D.R. dat er getaltransport plaats vindt. Daartoe worden bij elk Breakverzoek door de cyclusgenerator achtereenvolgens de volgende cycli gegenereerd:

1) Word Count cyclus.

Het getal dat op een door ons gespecificeerd (even) geheugenadres P staat, wordt met een verhoogd. Treedt er overflow op dan geeft de D.R. een zogenaamde WC-Overflowpuls (de Word Count Overflow kan gebruikt worden om het aantal Breaks te tellen).

2) Current Address cyclus.

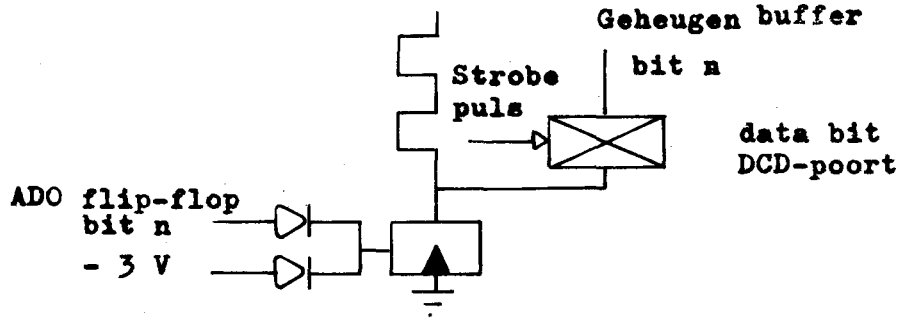
Het getal dat op de geheugenplaats P + 1 staat wordt al of niet met 1 verhoogd. (P + 1 bevat dus het adres waar de gegevens in-of uitgelezen worden).

C De Break cyclus.

Het eigenlijke transport van getallen vindt plaats.

6.1.4 In- en uitlezen van gegevens.

De manier waarop we een enkele bit in de D.R. inlezen m.b.v. data break is in de volgende figuur geschetst.

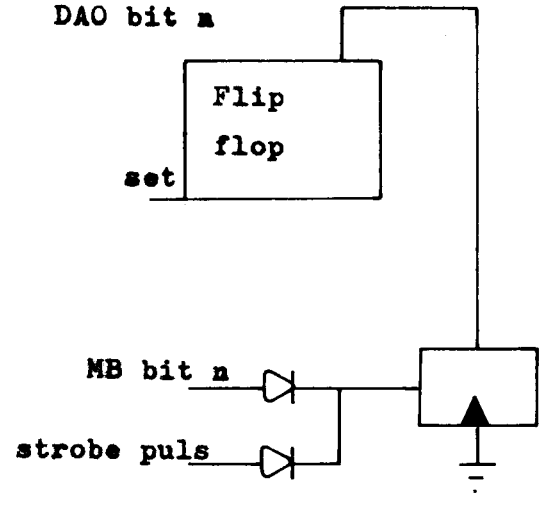


De n-de bit van de ADO flip-flop is verbonden met een ingang van een NAND-poort, terwijl de andere ingang continu verbonden is met een niveau van - 3 Volt. (Hier is rekening gehouden met de mogelijkheid om eventueel nog andere gegevens m.b.v. data break in te lezen).

Stel bit n = 0. Op de 0-output staat 0 Volt. We verbinden de 1-output van de met de bovenste ingang van de NAND-poort, de uitgang van de poort is dan negatief en de strobe puls komt hier door de data bit DCD poort, waardoor er een nul in het geheugenbuffer blijft staan. Is de 1-output negatief (1-stand van de flip-flop) dan komt er een 1 in het geheugenbuffer.

De data bit DCD-poort is een stekkersbus in de rekenmachine. Het transport van de data vanaf deze bus naar het geheugenbuffer wordt door de D.R. zelf verzorgd.

Bij het uitlezen van de gegevens gaan we als volgt te werk.



Door middel van een setpuls zetten we de flip-flop in de 1-stand. Is bit n van het kerngeheugenbuffer (MB) een 1 (0 Volt) dan zal de door ons gegenereerde strobe puls geen effect hebben op de 1-stand van de flip-flop, waardoor we dus een 1 hebben ingelezen.
Bevat bit 1 echter een 0 (-3 Volt) dan wordt de uitgang van de NAND-poort positief tengevolge van de negatieve strobe puls en de flip-flop wordt in de 0-stand gezet.

6.1.5 Een dode tijd subroutine met behulp van de directe methode.

Het is mogelijk de D.R. te verbieden tijdens de current address cyclus de inhoud van de geheugenplaats P + 1 met 1 te verhogen (CA Inhibit). Hierdoor kunnen we zeer eenvoudig een dode tijd genereren. Daardoor moeten we als volgt te werk gaan:

We zorgen dat alleen de DAO de kans krijgt het current address met 1 te verhogen. Indien we er bovendien nog voor zorgen dat de ADO break steeds later komt dan de DAO-break, dan wordt de ADO ingelezen op de plaats waar we met de DAO hebben uitgelezen, dus:

geheugenplaats	N	1) DAO uit	2) ADO in
	N + 1	3) DAO uit	4) ADO in
	N + 2	5) DAO uit	6) ADO in
		etc.	

De dode tijd die we hiermee genereren is gelijk aan het aantal gebruikte geheugenplaatsen maal de monstertijd, namelijk na 2 m breaks (ADO + DAO breaks) kunnen we een WC-Overflowpuls krijgen, die

interruptprogramma initialiseert.

Dit zorgt ervoor dat we weer op geheugenplaats N beginnen uit te lezen; wat we m bemonsteringen eerder hebben ingelezen. Dit interrupt programma is uiteraard veel korter dan de normale dode tijd subroutine.

Indien we dus een data break systeem bouwen, hebben we naast het voordeel van sneller in- en uitlezen nog een extra voordeel omdat het maken van een dode tijd veel eenvoudiger is.

6.2 Sturing.

6.2.1 Algemeen.

Het is mogelijk met de analoge rekenmachine repeterend te rekenen (d.w.z. hij kan zelfstandig een onbeperkt aantal malen van rekenen naar I(nitial) C(ontrol) en van IC naar rekenen schakelen) en op die manier de D.R. te sturen.
De IC-tijd is in te stellen op 0,1 sec.en op 1 sec. De rekentijd is continu regelbaar tussen o,1 sec. en 110 sec.
Gedurende de rekenperiode wordt een analoog proces met een bepaalde regelaar geregeld. Bij deze PR-combinatie wordt tevens het criterium bepaald.
In de IC-stand wordt een van de regelparameters gewijzigd om gedurende de volgende rekenperiode met deze combinatie te regelen.
De taakverdeling tussen de A.R. en de D.R. is nog eens weergegeven in onderstaande tabel.

Stand	A.R.	D.R.
Rekenen	Processimulatie	Bemonsteren etc.
I.C.	Condensatoren initialiseren	Nieuwe regelacties bepalen

De analoge rekenmachine staat 100 msec. in IC terwijl de digitale rekenmachine voor zijn pauzeprogramma slechts 1 msec. nodig heeft.
Deze discrepantie qua tijd hebben we opgeheven door de A.R. te laten sturen door de D.R.

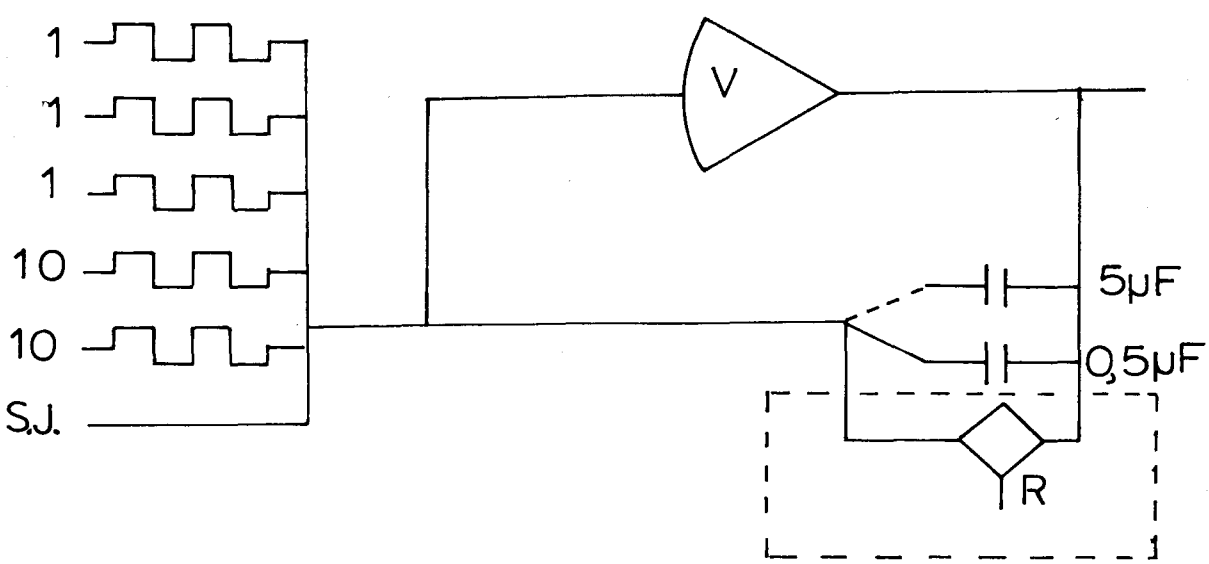
6.2.2 Sturing van de AR door de D.R.

De A.R. biedt de mogelijkheid tot sturing van buitenaf.

Hiervoor zijn op het programmeerveld bussen aangebracht. Deze sturing geschiedt electro-mechanisch dus zeer traag.

Om de snelheid van de D.R. helemaal uit te buiten is besloten om elektronische schakelaars te gebruiken. Deze worden als volgt toegepast: De A.R. staat continu te rekenen, echter na beëindiging van het rekenprogramma van de D.R. of indien de D.R. merkt dat de criteriumwaarde gedurende het rekenen al groter is geworden dan aan het eind van de vorige rekenslag, wordt de condensator die in de analoge integrator zit voor korte tijd kortgesloten en ontladen door de elektronische schakelaar die parallel aan de condensator zit. Hierbij bleek een tijd van 1,17 msec. voldoende om de twee condensatoren van 0,5μF, die nodig zijn voor de simulatie van een tweede orde proces te ontladen.

De integrator met elektronische schakelaar is in onderstaande figuur nog eens geschetst. De sturing van de elektronische schakelaar geschiedt vanuit het programma.



Elektronische schakelaar.

6.2.3 Adviezen wat betreft de apparatuur.

Het analoge rekencentrum, dat door onze groep beheerd wordt, beschikt sinds kort over een snelle analoge rekenmachine, de EAI 680.

De kleinst mogelijke tijdconstante van de integratoren die op deze machine te verwezenlijken is, bedraagt 10 μ sec.

De I.C.-tijd van de integratoren bedraagt 30 μ sec.

Verder is er voor de PDP-8 een elektronische vermenigvuldiger besteld. Vermenigvuldigen gaat dan tien keer zo kort duren als momenteel het geval is.

Door beide verbeteringen te combineren zullen we uitkomen op een hybride rekenmachine die nauwkeuriger en ongeveer een factor 5 sneller is.

Litteratuurlijst.

1) Cuypers, J.G.M.
Het dynamische gedrag van enkele processen met dominante looptijd, die intermitterend of continu worden geregeld.
(Stageverslag).

2) Cuypers, J.G.M.
Studies over de invloed van kwantisatie op de regeling van processen.
(Afstudeerverslag).

3) Hermeling, G.M.W.
Enige aspecten van fouten en correctiemogelijkheden bij hybride systemen.
(Stageverslag).

4) Nagel, A.L.
Optimalisatie van intermitterende regelsystemen met behulp van dynamische criteria.
(Proefschrift).

5) Sipman, W.H.M.
Optimalisatie van bemonsterde systemen.
(Afstudeerverslag).

6) Telefunken
Transistierter Tischanalogrechner RAT 740
(Beschreibung).

7) The Digital Logic Handbook
1967 Edition.

8) Small Computer Handbook
1967 Edition.

I-Actie

Crit.

⊙ = ∫

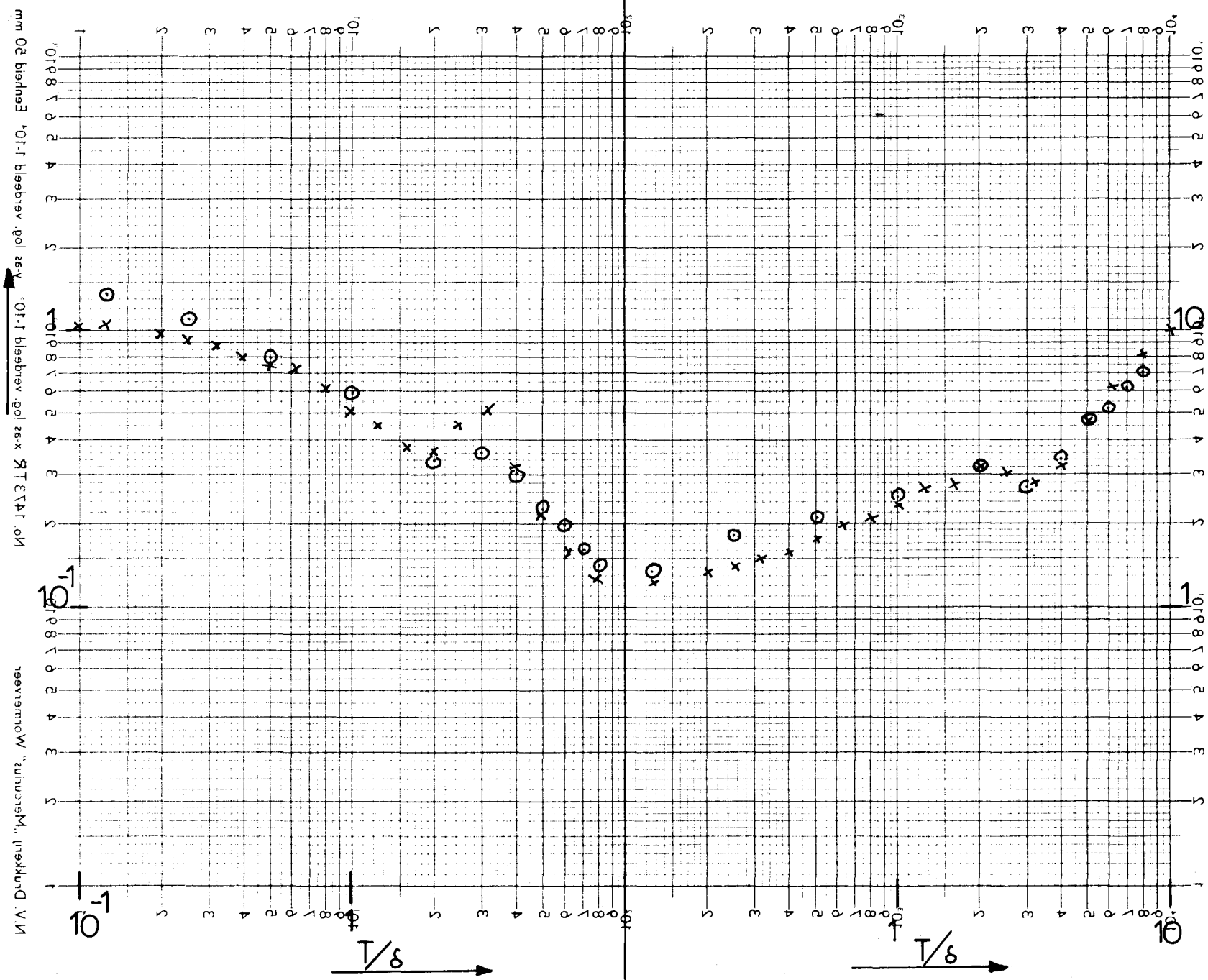
× = Σ

T/6

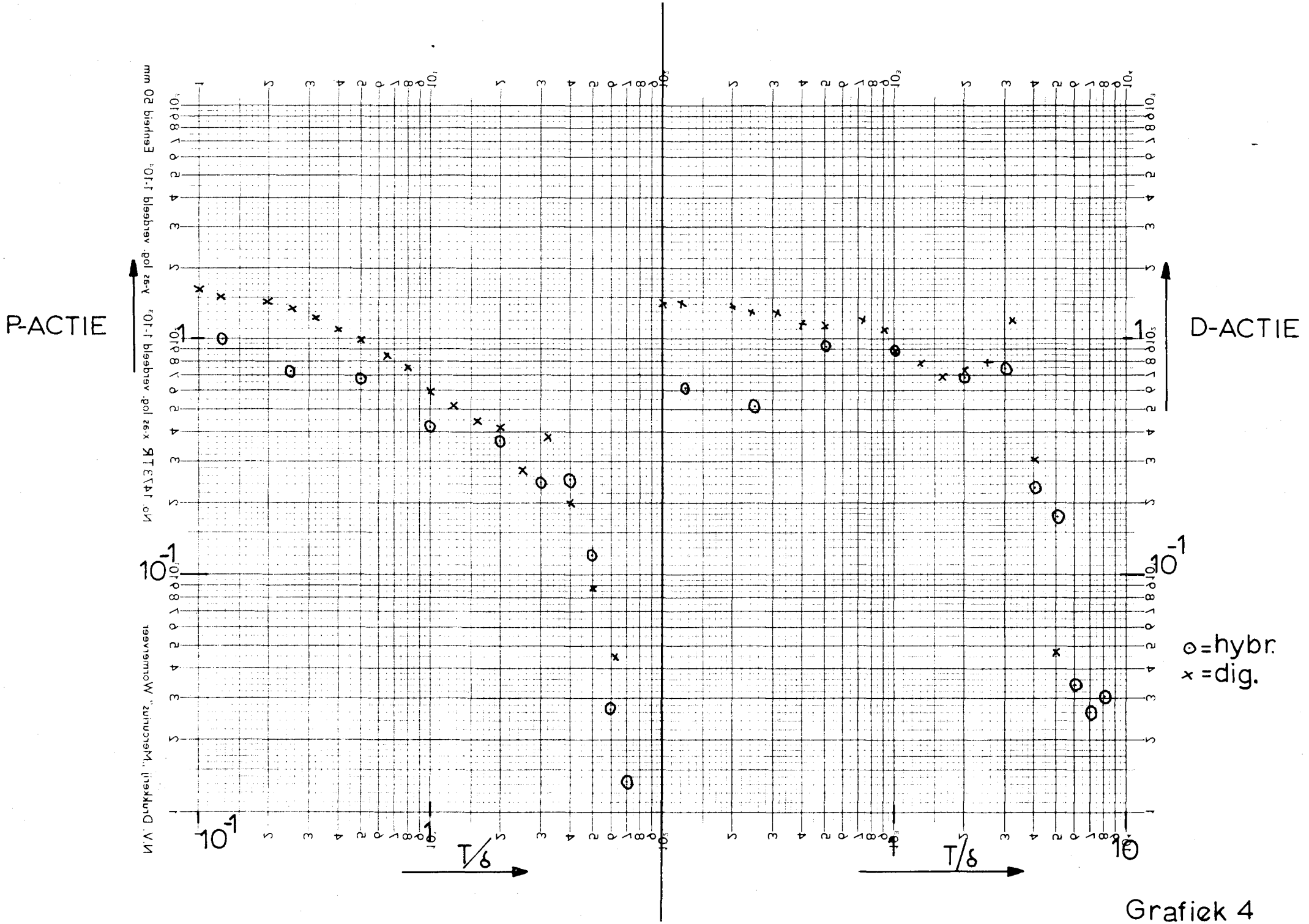
T/6

Grafiek 1

Grafiek 2



Grafiek 3



Grafiek 4