



DIE ONTWIKKELING VAN 'N MEETINSTRUMENT OM
DIE TRANSIËNTE VAN ROTERENDE MASJIENE TE
MEET

deur

Daniel Stephanus van den Berg

Verhandeling ingelewer ter voldoening aan
die vereistes vir die graad

Magister Technologiae
Ingenieurswese : Elektries

in die
Fakulteit Ingenieurswese
Departement Elektriese Ingenieurswese
aan die
Technikon Vrystaat

Januarie 1998

Studieleier : Dr. G.D. Jordaan, D Tech (Eng.)

VERKLARING TEN OPSIGTE VAN SELFSTANDIGE WERK

Ek, DANIEL STEPHANUS VAN DEN BERG, verklaar hiermee dat die navorsingsprojek wat vir die verwerwing van die graad MAGISTER TECHNOLOGIAE: INGENIEURSWESE: ELEKTRIES aan die Technikon Vrystaat deur my voorgelê word, my selfstandige werk is en nie voorheen deur my of enige ander persoon ter verwerwing van enige kwalifikasie voorgelê is nie.


.....
HANDTEKENING VAN STUDENT

11/1/98
.....
DATUM

BEDANKINGS

Ek wil graag die volgende persone en instansies bedank vir hul bydrae tot die voltooiing van die projek:

My studieleier, Dr. G.D. Jordaan, vir sy hulp en leiding gedurende die verloop van die projek.

Technikon Vrystaat vir die bereidwilligheid om die projek te hanteer.

Telkom, vir die geleenthede wat hulle aan my gebied het.

My ouers vir die opvoeding en geleenthede aan my verskaf om tot op hierdie vlak te vorder.

OPSOMMING

Transduseerders kan onder andere aangewend word om defekte in roterende masjiene vroegtydig op te spoor. Indien 'n potensiële defek betyds, sonder om die masjien uitmekaar te haal, opgespoor kan word, kan die masjien op 'n stadium wanneer die instandhoudingsfasiliteite beskikbaar is, met 'n minimum verlies aan produksie herstel word. Defekte kan opgespoor word deur die snelheid en versnelling parameters van die masjien te analiseer.

Die projek behels die ontwikkeling en evaluering van 'n volledige stel sagteware, tesame met 'n hoë-resolusie meetinstrument, vir die meting van die snelheid en versnelling van 'n roterende masjien. Die hardeware van die meetinstrument bestaan uit die volgende:

- **As-enkodeerder:** 'n Hoë-resolusie inkrementele as-enkodeerder word gebruik om die draaimoment van die roterende as te bepaal. Die uitsetpuls van die as-enkodeerder word geprosesseer om die vorming van ruis te minimaliseer.
- **Seinprosesseringseenheid:** Die snelheid van die masjien word deur middel van analoog tegnieke vanaf die seinprosesseringseenheid verkry deur die uitsetpuls van die as-enkodeerder om te skakel in 'n spanning. Deur differensiasie van die snelheidsein word die oombliklike versnelling van die masjien verkry. Die uitset van die meetinstrument gaan deur 'n analoog-na-digitale omsetterkaart voordat die sein deur 'n rekenaar verwerk word.

Die sagteware ontwikkel vir die meetinstrument behels onder meer die vertoning van 'n spyskaart met die keuses wat uitgevoer kan word om die korrekte tempo van monsterring van die snelheid en versnelling van die motor te verseker. Die bemonsterde data word in grafiese formaat op die rekenaarskerm vertoon en gestoor vir verwerking.

Die korrekte werking van die meetinstrument is bevestig deur middel van praktiese evaluasie. Verskeie toetse is in hierdie verband uitgevoer en bevredigende resultate is daarmee behaal.

Besondere kennis aangaande die onderstaande is ingewin:

- Transiënt eienskappe van roterende masjiene.
- Formaat, samestelling en gebruik van transduseerders.

SUMMARY

Transducers are needed to detect defects in rotational machines. If a potential defect is detected in time, without dismantling the machine, repairs could be carried out when the maintenance facilities are available with minimum loss in production. Analysis of the speed and acceleration parameters are used to predict the existence of defects in a machine.

This project involved the development and utilisation of a comprehensive set of software in conjunction with a high resolution measuring instrument, which have the ability to measure both the speed and acceleration parameters of rotational machines. The measuring instrument hardware consists of the following:

- Shaft encoder: A high resolution incremental shaft encoder is used to determine the torque of the rotating shaft. The output pulse of the shaft encoder is processed within the encoder to minimise the generation of noise.
- Signal processing unit: The speed of the machine is obtained from the signal processing unit with the aid of analogue techniques, by converting the output pulses of the shaft encoder to a corresponding voltage. By differentiation of the speed signal, the instantaneous acceleration of the machine is obtained. The output of the instrument is digitized by an analogue-to-digital converter card before the signal is processed by a computer.

The software developed for the measuring instrument displays a menu, showing the options to be carried out to ensure the correct sampling of the speed and acceleration of the motor. The sampled data is shown graphically on the computer screen before it is saved for processing.

Correct operation of the measuring instrument has been verified by practical evaluation. A number of tests in this regard have been carried out and satisfactory results were achieved.

Particular knowledge regarding the following has been obtained:

- Transient characteristics of rotational machines.
- Format, construction and use of transducers.



INHOUD

HOOFSTUK 1	1
<i>INLEIDING</i>	
1.1 OORSIG	1
1.2 PROBLEEMSTELLING	1
1.3 DOELWIT VAN PROJEK	2
1.4 HIPOTESE	2
1.5 BELANGRIKHEID VAN PROJEK	3
1.6 METODE VAN NAVORSING	3
1.6.1 EVALUASIE VAN DIE TWEESKYF SENSOR	4
1.6.2 ALTERNATIEWE TRANSDUSEERDERS	4
1.6.3 ONTWIKKELING VAN SAGTEWARE	4
HOOFSTUK 2	5
<i>TRANSDUSEERDERS</i>	
2.1 DOELWIT VAN METING	5
2.2 TRANSIËNTE	6
2.2.1 TRANSIËNT KENMERKE	7
2.3 TRANSDUSEERDERS EN SENSORS	8
2.3.1 TRANSDUSEERDER TERMINOLOGIE	9
2.4 SPANNING-NA-FREKWENSIE EN FREKWENSIE-NA-SPANNING OMSETTERS	10
2.5 OPTIESE ELEKTRONIKA	13
2.5.1 LENSE VIR FOTO- EN LIGEMISSIEDIODES	15
2.6 MEETSTELSELS	17
2.6.1 TAGOMETER	17

2.6.2 WISKUNDIGE MODEL VIR TAGOMETERS	19
2.6.3 TWEESKYF SENSOR	20
2.6.4 AS-ENKODEERDER	24
2.7 MIKROREKENAARS EN DATAVERKRYGING.....	29
2.7.1 ANALOOG INSETTE EN UITSETTE - I/O	30
2.7.2 DIGITALE INSETTE EN UITSETTE	30
2.8 OPSOMMING.....	30

HOOFSTUK 3.....33

OPERASIONELE VERSTERKERS AS SEINPROSESSEERDERS

3.1 INLEIDING TOT OPERASIONELE VERSTERKERS	33
3.2 BASIESE OPERASIONELE VERSTERKER.....	34
3.3 INTEGREERDER	35
3.4 DIFFERENSIEERDER	38
3.5 OPERASIONELE VERSTERKER FILTERS	41
3.6 ONDERDEURLAATFILTER	43
3.7 BODEURLAATFILTER	45
3.8 NORMALISERING VAN FILTERS	46
3.9 BUTTERWORTH FILTERS	47
3.10 OPSOMMING	51

HOOFSTUK 4.....53

ONTWERP VAN DIE HARDEWARE

4.1 INLEIDING TOT DIE ONTWERP VAN DIE HARDEWARE	53
4.2 SENSOR ONTWERP	55
4.2.1 TWEESKYF, EEN LIGBRON STELSEL.....	56

4.2.2 TWEESKYF LIGBRON EN LENSSTELSEL	58
4.2.3 AS-ENKODEERDER	60
4.3 MONOSTABIELE MULTIVIBRATOR	62
4.4 SNELHEIDVERWERKINGSTROOMBAAN	64
4.5 FREKWENSIE-NA-SPANNING GEÏNTEGREERDE STROOMBAAN	70
4.6 VERSNELLINGVERWERKINGSTROOMBAAN	72
4.7 OPSOMMING	75
HOOFSTUK 5.....	77
SAGTEWARE	
5.1 INLEIDING TOT DIE ONTWERP VAN DIE SAGTEWARE.....	77
5.2 HOOFPROGRAM	77
5.2.1 INLIGTINGSUBPROGRAM	80
5.2.2 HELPSUBPROGRAM	81
5.2.3 HERSTELSUBPROGRAM	81
5.2.4 EINDE VAN PROGRAM	81
5.3 MONSTERFREKWENSIEPROGRAM	82
5.3.1 FREKWENSIE STOORSUBPROGRAM	83
5.3.2 FREKWENSIE LAAISUBPROGRAM	83
5.4 MONSTERDATAPROGRAM.....	83
5.4.1 MONSTER LAAISUBPROGRAM	84
5.5 MONSTERKANAALPROGRAM	84
5.5.1 VERIFIKASIESUBPROGRAM	85
5.5.2 MONSTERKANAAL LAAISUBPROGRAM	86
5.6 MONSTERPROGRAM	86
5.6.1 VERKRY STELSELKLOKSUBPROGRAM	87
5.6.2 ONDEBREKINGSBEHEERDE BEMONSTERINGSUBPROGRAM	88

5.6.3 VERTOON DATASUBPROGRAM	89
5.6.4 STOOR DATASUBPROGRAM	90
5.7 OPSOMMING	90
HOOFSTUK 6.....	92
<i>EVALUERING VAN DIE MEETINSTRUMENT</i>	
6.1 INLEIDING TOT DIE EVALUERING VAN DIE MEETINSTRUMENT	92
6.2 LINEARITEIT VAN DIE SNELHEIDVERWERKINGSTROOMBAAN	92
6.3 EVALUERING VAN SENSORS	94
6.3.1 TWEESKYF LIGBRON EN LENSSTELSEL	94
6.3.2 INKREMENTELE AS-ENKODEERDER	96
6.4 EVALUERING VAN SNELHEIDVERWERKINGSTROOMBAAN...	99
6.5 EVALUERING VAN DIE SAGTEWARE	102
6.6 GEVOLGTREKKING	104
HOOFSTUK 7.....	106
<i>SAMEVATTING</i>	
BYLAAG A	108
<i>BLOKDIAGRAM VAN SEINPROSESSERINGSEENHEID</i>	
BYLAAG B	111
<i>ADRES-OPSTELLING VAN PC-30 KAART</i>	

BYLAAG C	113
<i>HOOFPROGRAM</i>	
BYLAAG D	127
<i>MONSTERFREKWENSIEPROGRAM</i>	
BYLAAG E	134
<i>MONSTERDATAPROGRAM</i>	
BYLAAG F	139
<i>MONSTERKANAALPROGRAM</i>	
BYLAAG G	151
<i>MONSTERPROGRAM</i>	
BYLAAG H	166
<i>HELPSUBPROGRAM</i>	
LITERATUURLYS	170
ADDISIONELE BRONNE	174
(GERAADPLEEG, MAAR NIE NA VERWYS NIE)	

LYS VAN FIGURE

HOOFSTUK 2

FIG. 2.1 - TRANSIËNT KENMERKE	8
FIG. 2.2 - FREKWENSIE-NA-SPANNING OMSETTER.	12
FIG. 2.3 - BASIESE FOTODIODE STROOMBAAN.	14
FIG. 2.4 - FOTOTRANSISTOR KONFIGURASIES.	15
FIG. 2.5 - EENVOUDIGE LENS VOORSTELLING.	16
FIG. 2.6 - UITGEHOLDE HOUER MET FOTODETEKTOR.	17
FIG. 2.7 - BEHEERSTELSEL MET TAGOMETER TERUGVOER.	18
FIG. 2.8 - TWEESKYF SENSOR.	22
FIG. 2.9 - BORSEL TIPE ENKODEERDER.	24
FIG. 2.10 - OPTIESE ENKODEERDER.	25
FIG. 2.11 - VIER BIS BINÊRE ENKODEERDER.	25
FIG. 2.12 - TIPIESE INKREMENTALE ENKODEERDER.	26
FIG. 2.13 - TIPIESE SINUSVORMIGE UITSETGOLFFORMS.	28
FIG. 2.14 - INSET/UITSETSEINE VAN 'N SCHMITT SNELLERKRING.	28

HOOFSTUK 3

FIG. 3.1(A) - BASIESE OPERASIONELE VERSTERKER.	34
FIG. 3.1(B) - EKWIVALENTE STROOMBAAN VAN OPERASIONELE VERSTERKER.....	35
FIG. 3.2 - PASSIEWE INTEGREERDER.	35
FIG. 3.3 - PRAKTIESE OPERASIONELE VERSTERKER INTEGREERDER.	36
FIG. 3.4 - INTEGREERDER KENKROMME.....	38
FIG. 3.5 - OPERASIONELE VERSTERKER DIFFERENSIEERDER.	39
FIG. 3.6 - PRAKTIESE OPERASIONELE VERSTERKER DIFFERENSIEERDER.	40

FIG. 3.7 - DIFFERENSIEERDER FREKWENSIEKENKROMME.....	41
FIG. 3.8 - FREKWENSIE RESPONS VAN IDEALE FILTERS: (A)BODEURLAAT; (B)ONDERDEURLAAT; (C)BANDDEURLAAT.	43
FIG. 3.9 - BASIESE AKTIEWE ONDERDEURLAATFILTER.	44
FIG. 3.10 - ONDERDEURLAATFILTER MET NEGATIEWE TERUGKOPPELING. .44	
FIG. 3.11 - BASIESE AKTIEWE BODEURLAATFILTER.	45
FIG. 3.12 - BODEURLAATFILTER FREKWENSIERESPONS	45
FIG. 3.13 - BODEURLAATFILTER MET TERUGKOPPELING.....	46
FIG. 3.14 - ONDERDEURLAATFILTER (A)EERSTE-ORDE (B)TWEEDE-ORDE. 48	
FIG. 3.15 - BUTTERWORTH ONDERDEURLAATFILTER RESPONS.	49

HOOFSTUK 4

FIG. 4.1 - BLOKDIAGRAM VAN MEETSTELSEL.	53
FIG. 4.2 - BLOKDIAGRAM VAN SEINPROSESSERINGSEENHEID.	54
FIG. 4.3 - SENSORSAMESTELLING SONDER LENSE.....	56
FIG. 4.4 - SEIN VERSTERKER.	57
FIG. 4.5 - SENSORSAMESTELLING MET LENSE.....	59
FIG. 4.6 - FREKWENSIEDELER.....	62
FIG. 4.7 - MONOSTABIELE MULTIVIBRATOR.	63
FIG. 4.8 - BUTTERWORTH DERDE-ORDE ONDERDEURLAATFILTER.	65
FIG. 4.9 - LM2917 FREKWENSIE-NA-SPANNING OMSETTER.	71
FIG. 4.10 - GEMETE LINEARITEIT VAN LM2917 FREKWENSIE-NA-SPANNING OMSETTER.	71
FIG. 4.11 - BASIESE DIFFERENSIEERDER.	72
FIG. 4.12 - DIFFERENSIEERDER.	73

HOOFSTUK 5

FIG. 5.1 - STRUKTUURKAART VAN HOOFPROGRAM.	78
---	----

HOOFSTUK 6

FIG. 6.1 - LINEARITEIT VAN DIE SNELHEIDVERWERKINGSTROOMBAAN SONDER FREKWENSIEDELER.	93
FIG. 6.2 - LINEARITEIT VAN DIE SNELHEIDVERWERKINGSTROOMBAAN MET FREKWENSIEDELER.	93
FIG. 6.3 - SNELHEIDUITSET VAN TWEESKYF LIGBRON EN LENSSTELSEL. ..	95
FIG. 6.4 - VERSNELLINGUITSET VAN TWEESKYF LIGBRON EN LENSSTELSEL	96
FIG. 6.5 - SNELHEIDUITSET MET DIE AS-ENKODEERDER.	97
FIG. 6.6 - VERSNELLINGUITSET MET DIE AS-ENKODEERDER.	97
FIG. 6.7 - SNELHEIDUITSET TYDENS STOP EN VERSNELLING.	98
FIG. 6.8 - VERSNELLINGUITSET TYDENS STOP EN VERSNELLING.	99
FIG. 6.9 - TYDRESPONS MET 'N SEINGENERATOR AS INSET.	101
FIG. 6.10 - SNELHEIDUITSET VERKRY MET ONTWIKKELDE SAGTEWARE. .	103
FIG. 6.11 - VERSNELLINGUITSET VERKRY MET ONTWIKKELDE SAGTEWARE	103

BYLAAG A

FIG. A.1 - KRINGDIAGRAM VAN DIE SEINPROSESSERINGSEENHEID.	108
FIG. A.1 - (VERVOLG) KRINGDIAGRAM VAN DIE SEINPROSESSERINGSEENHEID.	109
FIG. A.2 - KRAGBRON VAN DIE SEINPROSESSERINGSEENHEID.	110

LYS VAN TABELLE

HOOFSTUK 3

TABEL 3.1 - BUTTERWORTH D EN ω_0 WAARDES.....	51
--	----

HOOFSTUK 4

TABEL 4.1 - BUTTERWORTH POOLPLASING.....	65
--	----

HOOFSTUK 5

TABEL 5.1 - DATA MONSTERS TEEN BEMONSTERINGKANALE.....	85
--	----

BYLAAG B

TABEL B.1 - POORTADRESSE VAN AD574 OMSETTER.....	111
--	-----

TABEL B.2 - ADRESSE VAN PC-30 KAART.....	112
--	-----

HOOFSTUK 1

INLEIDING

1.1 OORSIG

Tegniese vir die meting van snelhede van 'n motor behels tradisioneel die gebruik van een of ander vorm van tagometer wat die spoed direk op 'n analogeskaal vertoon. Die monitering van motors word al hoe belangriker en 'n unieke meetstelsel, om die snelheid en versnelling parameters van motors te meet, sal van groot waarde wees aangesien analisering daarvan dit moontlik maak om te voorspel of 'n spesifieke motor herstelwerk benodig - voordat ernstige skade berokken word. Deur herstelwerk vroegtydig te skeduleer kan die verlangde materiale verkry word om die herstelwerk so vinnig as moontlik te laat geskied, met 'n minimum verlies in produksie. In die uitvoering van die navorsingsprojek is sodanige snelheid en versnellingstransduseerder ontwerp en getoets.

1.2 PROBLEEMSTELLING

Rekenaars en moderne digitale tegnologie kan tot groot voordeel in die meting van snelheid en versnelling van motors gebruik word. Metodes is ondersoek waardeur hierdie tegniese in die ontwikkeling van die meetinstrument ingesluit kon word. Die uitsetseine daarvan

word in 'n rekenaar ingelees en gestoor vir moontlike verdere prosessering.

1.3 DOELWIT VAN PROJEK

Die hoofdoel van die projek was om 'n meetinstrument te ontwikkel wat die hoekversnelling en snelhede van roterende masjiene kan meet. Die stelsel is ontwikkel om op roterende masjiene met 'n maksimum rotasiesnelheid van ongeveer 10000 omwentelings per minuut gebruik te kan word. 'n Vereiste is dat 'n gedeelte van die motor-as moet uitsteek vir koppeling van die instrument. Die projek het ook die ontwikkeling van sagteware om die data vanaf die meetinstrument in die rekenaar in te lees, behels.

1.4 HIPOTESE

Na die mening van die navorser kon 'n meetinstrument, met 'n eenvoudige meganiese koppeling, wat nie die werking van die masjien beïnvloed nie, ontwikkel word. So 'n stelsel sal gleuf-optiese waarneming, of foto-sensitiewe sensors, inkorporeer en 'n persoonlike rekenaar gebruik om die meting van die transiënte in beweging van 'n roterende masjien moontlik te maak.

1.5 BELANGRIKHEID VAN PROJEK

Die meting van 'n motor se assnelheid is tradisioneel met behulp van 'n tagometer gedoen. Tagometers leen hulself tot die monitering van die snelhede van groot motors, maar die snelhede van kleiner motors mag aansienlik verminder word deur die koppeling van tagometers.

'n Alternatiewe metode behels die gebruik van 'n klein gelykstroomgenerator wat direk aan die motor gekoppel word. Die uitset van die generator kan gestel word dat dit direk met die snelheid van die motor verander en 'n voltmeter kan dan gebruik word om 'n aanduiding hiervan te gee. Hierdie rangskikking is ook net geskik vir groot motors.

Elektroniese metodes gebruik die rangskikking van sensors van een of ander tipe om die assnelheid waar te neem sonder oortollige meganiese koppeling aan die masjien. Die sensors moet verkieslik 'n hoë monstertempo moontlik maak ten einde soveel moontlik pulse per revolusie van die as te meet ter lewering van 'n akkurate uitset.

1.6 METODE VAN NAVORSING

Die projek is opgedeel in drie fases, nl.:

- Evaluasie van 'n tweeskyf transduseerder.
- Ondersoek na alternatiewe transduseerders.
- Ontwikkeling van sagteware.

1.6.1 EVALUASIE VAN DIE TWEESKYF SENSOR

Aanvanklik het die meetinstrument uit twee roterende skywe met 'n gleuf-optiese sensorsamestelling bestaan. Die uitsetpuls van 'n fototransistor word verwerk en twee uitsetseine, naamlik die snelheidsein en die versnellingsein van die motor, word verkry. Die aantal pulse ontwikkel per revolusie is gelyk aan die aantal gleuwe van die skywe, en die pulsfrekwensie is dus eweredig aan die spoed van die motor.

1.6.2 ALTERNATIEWE TRANSDUSEERDERS

'n As-enkodeerder is ook as moontlike transduseerder getoets. Die totale massa van die transduseerder moet laag wees, ten einde oormatige fisiese belasting, en gevolglik foutiewe lesings, van die motor te vermy.

1.6.3 ONTWIKKELING VAN SAGTEWARE

Die sagteware ontwikkel maak dit moontlik om die bemonsterde data soos verkry in 'n teksformaat te stoor. Ander rekenaarprogramme, soos byvoorbeeld Quattro Pro of Excel, kan gebruik word om die data te analiseer.

HOOFSTUK 2

TRANSDUSEERDERS

2.1 DOELWIT VAN METING

Die fisiese kwantiteit van byvoorbeeld krag, druk, verplasing en temperatuur is beskikbaar in analoogvorm en kan omgeskakel word in elektriese spannings deur middel van transduseerders en versterkers [10, p. 35]. Seinprosessering van byvoorbeeld die piek, oombliklike en limietwaardes is tradisioneel met behulp van analoogkringe gedoen. Dié tipe kringe het 'n aantal nadele, soos tyd en temperatuurdryf, parameter variasies en beperkte buigsaamheid ten opsigte van aanpasbaarheid tussen verskillende metings. Komplekse seinprosessering en sein-analisering word beswaarlik in analoogvorm bewerkstellig. Deur digitale seinprosessering toe te pas word al hierdie probleme oorkom, mits die analoogseine versyfer kan word sonder om informasie te verloor.

Met die meting van meganiese kwantiteite is daar baie parameters wat oorweeg moet word. Hieronder is las, krag, druk, draaimoment, vervorming en verplasing. Vir elk van hierdie parameters is daar 'n betreklik wye reeks transduseerders beskikbaar.

Versnelling is 'n parameter wat dikwels gemeet moet word, maar waarvoor daar wynig transduseerders beskikbaar is [18, p. 24]. Ten einde foute wat kan ontstaan as gevolg van die ekstra massa van

die transduseerder te voorkom moet die massa van die versnellingstransduseerder klein wees in vergelyking met die toetsstuk waarop dit gemonteer gaan word. Daarby moet dit in staat wees om enige skokke en stremmings te weerstaan wat gedurende vervoer na, en hegting aan, die toetsstuk ondervind mag word.

2.2 TRANSIËNTE

'n Oorgangsgebeurtenis, of transiënt, is 'n kortstondige verskynsel wat slegs in 'n geringe mate voorspelbaar herhaal kan word [17, p. 41]. Die golfvorm verkry is gewoonlik uniek as gevolg van die onvermoë om die toetskondisies presies te herhaal. As gevolg hiervan kan die vaslê van 'n oorgangsgebeurtenis - wat gedoen moet word terwyl dit gebeur - vir analise, 'n uiters moeilike proses wees. In 'n enkele aftasmodus moet die oorgangsgebeurtenis dus intyds gemeet, bemonster en vasgelê word. Die bandwydte van die toetsinstrument moet wyd genoeg wees om die vinnigste verwagte oorgange vas te lê, met 'n bemonsteringstempo hoog genoeg om die benodigde tydresolusie te verseker.

Omdat tyd gewoonlik as 'n onafhanklike veranderlike in meetstelsels gebruik word, is dit belangrik om die uitsetrespons en -staat te evalueer met betrekking tot tyd, of eenvoudiger gestel: die tydrespons word bepaal [11, p. 308]. Die tydrespons van 'n stelsel word gewoonlik verdeel in twee gedeeltes; die oorgangsrespons en die bestendige staat respons. Die bestendige staat word gedefinieer as die kondisie wanneer die respons 'n konstante waarde met betrekking tot die onafhanklike veranderlike

bereik het. 'n Oorgangsrespons, of transiënt, word gedefinieer as die gedeelte van die respons wat na zero gaan as die tydsduur toeneem. Die bestendige staat respons is daardie gedeelte van die respons wat oorbly nadat die transiënt uitgesterf het.

Alle beheerstelsels het tot 'n sekere mate transiënte na 'n insetverandering, voordat die bestendige staat bereik word. Omdat elemente soos traagheid, massa en induktansie nie volledig in fisiese stelsels vermy kan word nie, kan die respons van 'n tipiese beheerstelsel nie 'n skielike verandering in die inset oombliklik volg nie, en transiënte word waargeneem. Volgens [11, p. 326] is tipiese kenmerke wat gebruik word om transiënt response te karakteriseer verbyskiet, vertragingstyd, stygtyd en vestigingstyd.

2.2.1 TRANSIËNT KENMERKE

Die volgende vier definisies gee 'n direkte indikasie van transiënt karakteristieke (sien Figuur 2.1).

1. **Maksimum verbyskiet** word gedefinieer as die grootste verskil van die uitset oor die inset gedurende die transiënt. Die hoeveelheid verbyskiet word ook gebruik as 'n indikasie van die relatiewe stabiliteit van die stelsel. Die maksimum verbyskiet word gewoonlik weergegee as 'n persentasie van die finale waarde van die traprespons.

$$\text{Persentasie verbyskiet} = \frac{\text{Maksimum verbyskiet}}{\text{Finale waarde}} \times 100\% \quad (2-1)$$

2. **Vertragingstyd, T_d** , word gedefinieer as die tyd benodig deur die traprespons om 50% van die finale waarde te bereik.
3. **Stygtyd, T_r** , word gedefinieer as die tyd benodig deur die traprespons om te styg van 10% tot 90% van die finale waarde.
4. **Vestigingstyd, T_s** , word gedefinieer as die tyd benodig vir die uitsetswaai om te verminder en binne 'n spesifieke persentasie van die finale waarde te bly. 'n Tipiese persentasie van 5% word gewoonlik gebruik.

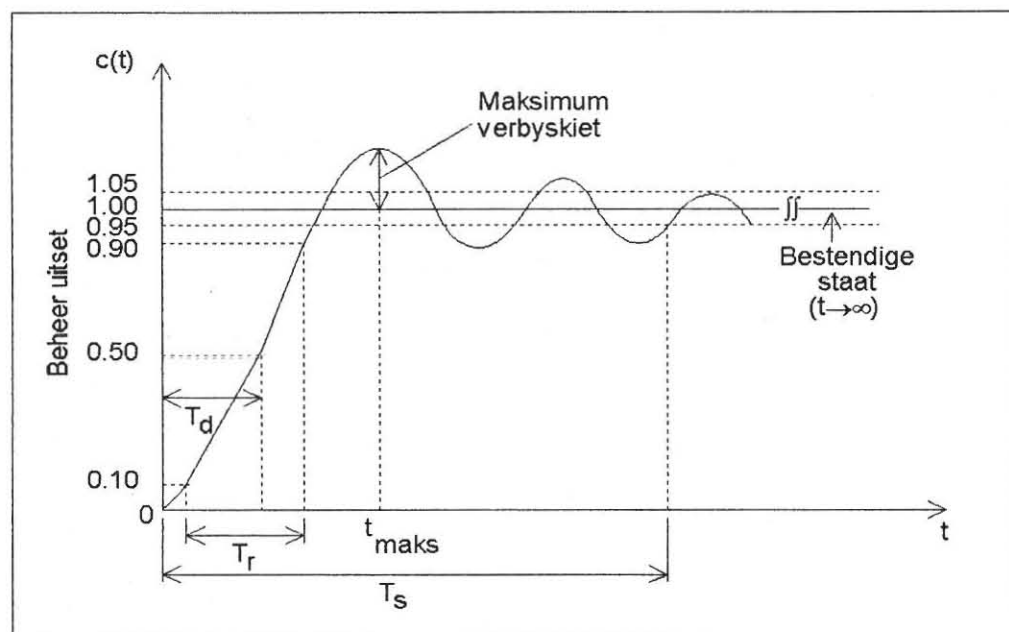


Fig. 2.1 - Transiënt kenmerke

2.3 TRANSDUSEERDERS EN SENSORS

Die woorde sensors en transduseerders word algemeen gebruik in die beskrywing van meetstelsels. 'n Sensor is daardie toestel wat veranderinge in 'n fisiese stimulus bespeur en dit verander in 'n sein wat gemeet kan word. 'n Transduseerder is 'n soortgelyke

toestel wat krag van een stelsel oordra na 'n ander in dieselfde of in 'n ander vorm [24, p. 3]. Die term transduseerder word gewoonlik gebruik om die volledige insettoestel te beskryf. Transduseerders kan ware energie-omskakelaars wees, of hulle kan 'n eksterne bron van energie benodig om 'n seinomskakeling tot gevolg te hê [3, p. 183].

Die primêre vereiste vir enige transduseerder is akkuraatheid en doeltreffendheid van seinomskakeling. Dit beteken dat die seinruisverhouding hoog moet wees, en die statiese kalibrasie nie moet verander nie en bestand moet wees teen veranderings in temperatuur en ander omgewingsfaktore. Daar moet verkieslik 'n lineêre verhouding tussen die inset en uitset wees en 'n vinnige respons word verlang. Die strek van die sensor moet by die verwagte verplasing pas, aangesien die meeste sensors tydens oorbelasting nie-lineêr word en onsensitief is as dit ver onder die volskaalse waarde gebruik word [26, p. 99]. Gebruik van 'n sensor moet beperk word tot sy maksimum of volskaalse waarde.

2.3.1 TRANSDUSEERDER TERMINOLOGIE

Die volgende terme word gebruik om die werkverrigting van transduseerders te beskryf [3, p. 183].

1. **Akkuraatheid** is die term wat gebruik word om die uitset van 'n instrument in verband te bring met die werklike waarde van die inset, met 'n gespesifiseerde standaardafwyking.

2. **Herhaalbaarheid** is die nabyheid aan mekaar van 'n groep uitsetwaardes vir 'n konstante inset onder gegewe omgewingskondisies.
3. **Resolusie** is die kleinste inkrement in die inset wat met sekerheid deur die transduseerder bespeur kan word.
4. **Sensitiwiteit** is die verhouding tussen die verandering van die uitset tot die ooreenstemmende verandering van die inset.
5. **Lineariteit** is die mate van hoe konstant die sensitiwiteit van die transduseerder is oor die strek van bruikbare waardes.
6. **Neutrale gebied** is die grootste verandering in die inset waarteen die transduseerder nie reageer nie.
7. **Histerese** is die algebraïese verskil tussen die gemiddelde uitsetfoute ooreenkomstig tot die insetwaardes, wanneer 'laasgenoemde benader word vanaf die maksimum en minimum moontlike insetwaardes.
8. **Dryf** is die verandering in die transduseerder se uitset wat nie veroorsaak word deur veranderinge in die inset nie.
9. **Zero stabiliteit** is die vermoë van die transduseerder om 'n uitset van zero te lewer wanneer die inset zero is.
10. **Monotonie** impliseer dat die uitset van die transduseerder, in respons tot 'n steeds stygende insetsein, onder geen omstandighede 'n waarde sal verminder of oorslaan nie.

2.4 SPANNING-NA-FREKWENSIE EN FREKWENSIE-NA-SPANNING OMSETTERS

Die funksie van 'n spanning-na-frekwensie omsetter is om 'n pulsreeks te lewer waarvan die frekwensie lineêr eweredig is aan

die analoog insetspanning [5, p. 624]. Sodoende verskaf die spanning-na-frekwensie omsetter 'n eenvoudige manier om analoogseine om te skakel na digitale seine. Die primêre rede vir die tipe van omskakeling is dat 'n pulsreeks met meer akkuraatheid gestuur en gekodeer kan word as 'n analoogsein, veral indien die transmissiekanaal baie lank is. Indien die spanning-na-frekwensie omsetter met 'n binêre teller en digitale uitset gekombineer word, word 'n lae koste digitale voltmeter verkry.

Die frekwensie-na-spanning omsetter voer die inverse funksie van die spanning-na-frekwensie omsetter uit. Dit aanvaar 'n wye verskeidenheid van golfvorms en lewer 'n analoogspanning wat lineêr eweredig is tot die insetfrekwensie. Frekwensie-na-spanning omsetters vind toepassing as tagometers in motor spoedbeheer en rotasiemetings. 'n Gebalanseerde spanning-na-frekwensie omsetter kan dikwels maklik verander word in 'n frekwensie-na-spanning omsetterkonfigurasie. Die insetfrekwensie word op die vergelyker toegepas en die uitsetspanning word afgelei van die operasionele versterker integreerder, met weerstand R in die terugvoerbaan, soos getoon in Figuur 2.2 [4, p. 404].

Die eenslag wipkring het 'n interne drumpelspanning van 7.5V en 'n ladingstroom van 1mA, om vergelyking (2-2) te lewer [4, p. 402].

$$T_H = \frac{7.5V}{1 mA} \times C \quad (2-2)$$

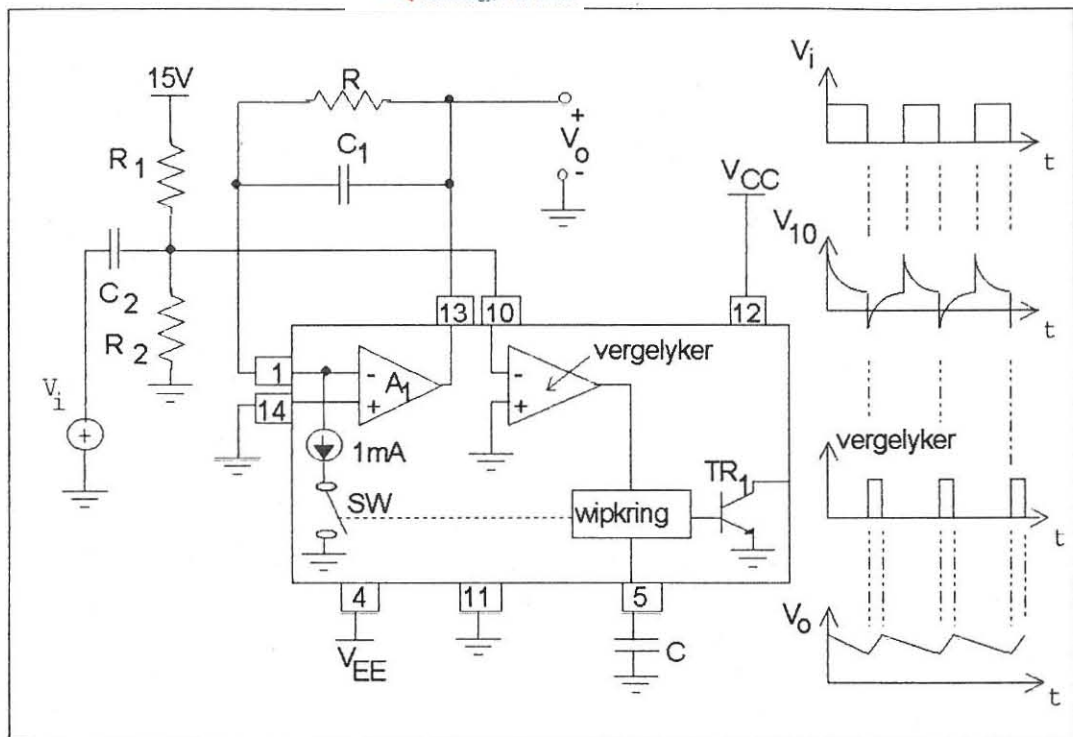


Fig. 2.2 - Frekwensie-na-spanning omsetter.

Met elke negatiewe oorgang op pen 10, word die eenslag wipkring gesneller deur die vergelyker, wat weer vir SW toemaak en sodoende 1mA uit C_1 onttrek vir tydperk T_H . As gevolg van die reeks pulse bou V_O op totdat die stroom wat uit die sommeerderversterker (A_1) getrek word presies teenoorgesteld is van die wat ingevoer word by V_O deur R . Met die insetfrekwensie gelyk aan f_{in} , is die pulsstroom $f_{in} \times 1\text{mA} \times T_H$, en die uitsetstroom is V_O/R . Deur die twee strome gelyk aan mekaar te stel en vergelyking (2-2) te gebruik volg dat,

$$V_O = 7.5(RC f_{in}) \quad (2-3)$$

wat aandui dat die uitsetspanning direk eweredig aan die insetfrekwensie is. Die insetafwykspanning van A_1 neig om die

akkuraatheid van die omsetting te benadeel teen laer waardes. Tussen opeenvolgende sluitings van SW, sal C_1 begin ontlaai deur weerstand R wat uitsetrimpelings veroorsaak. Die maksimum rimpel, $V_{R(\text{maks})}$, word verkry deur $C_1 V_{R(\text{maks})} = 1\text{mA} \times T_H$. Deur vergelyking (2-2) te gebruik volg dat,

$$V_{R(\text{maks})} = \frac{C}{C_1} \times 7.5V \quad (2-4)$$

wat aandui dat die rimpel verklein kan word deur C_1 te vergroot. As die kapasitor egter te groot is, sal die kring se respons afneem vir vinnige veranderinge in f_{in} aangesien die respons beperk word deur die tydkonstante $T = RC_1$. 'n Gepaste waarde vir C_1 is daarom 'n kompromie tussen hierdie twee opponerende vereistes.

2.5 OPTIESE ELEKTRONIKA

Optiese elektronika, in die konteks onder oorweging, kan beskou word as die studie van enige toestel wat 'n elektries-geïnduseerde optiese uitset lewer, of 'n opties-geïnduseerde elektriese uitset, sowel as die elektroniese tegnieke en stroombane wat gebruik word om die toestelle te beheer [13, p. 1]. 'n Bekende liguitstralende toestel is die ligemissiediode (LED). Dit het dieselfde elektriese karakteristieke as 'n normale diode en dit laat stroom deur in een rigting en blokkeer stroomvloei in die ander rigting. Lig word uitgestraal wanneer dit meevoorgespan is [13 p. 3]. Die liguitset vermeerder lineêr met voorwaartse stroom, maar nie-lineariteit word opgemerk by lae en hoë stroomwaardes [26, p. 84]. Ander tipes

liguitstralende toestelle is byvoorbeeld die katodestraalbuis en die laser.

Lig sensitiewe weerstande, fotovoltaiiese selle, fotodiodes en fototransistors is voorbeelde van ligsensitiewe toestelle. Fotodetektors is toestelle wat 'n verandering in elektriese karakteristieke ervaar met 'n verandering in die lig inset. 'n Fotodiode is 'n normale silikon diode wat in 'n deurskynende omhulsel gemonteer word, of die halfgeleier voegvlak is onder 'n optiese lens. Indien 'n silikon diode voegvlak teenvoorgespan word, sal die teenstroom afhang van die verligtingsintensiteit op die voegvlak (sien Figuur 2.3). Dit is ongeveer nul onder donker kondisies en vermeerder met lig intensiteit.

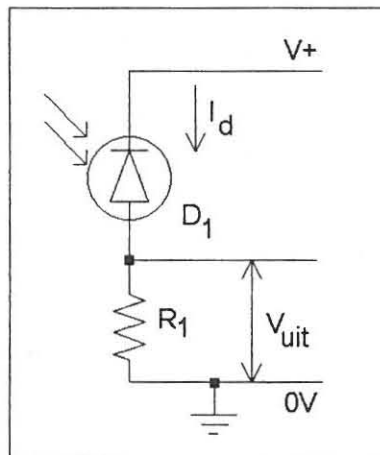


Fig. 2.3 - Basiese fotodiode stroombaan.

Fototransistors is normale silikon transistors met 'n ligsensitiewe voegvlak. Die basis-kollektor voegvlak van die transistor is teenvoorgespan in Figuur 2.4a en dit reageer dus as 'n fotodiode. Die foto-opgewekte stroom van die basis-kollektor voegvlak word direk ingevoer in die basis van die toestel.

stroomversterkingaksie van die transistor veroorsaak 'n groter uitsetstroom en R_1 veroorsaak dat die kollektorstroom 'n uitsetspanning afhangend van die invallende lig opwek [13, p. 105]. In die praktyk is die kollektor en emitter strome van die transistor basies identies, en omdat die basis 'n oopkring is, is die kring nie onderhewig aan noemenswaardige negatiewe terugvoer nie. Die fototransistor is ongeveer 10 keer meer sensitief as 'n fotodiode en kan gebruik word as 'n lig-tot-spanning omsetter deur dit in enige van die konfigurasies in Figuur 2.4 te bedraad.

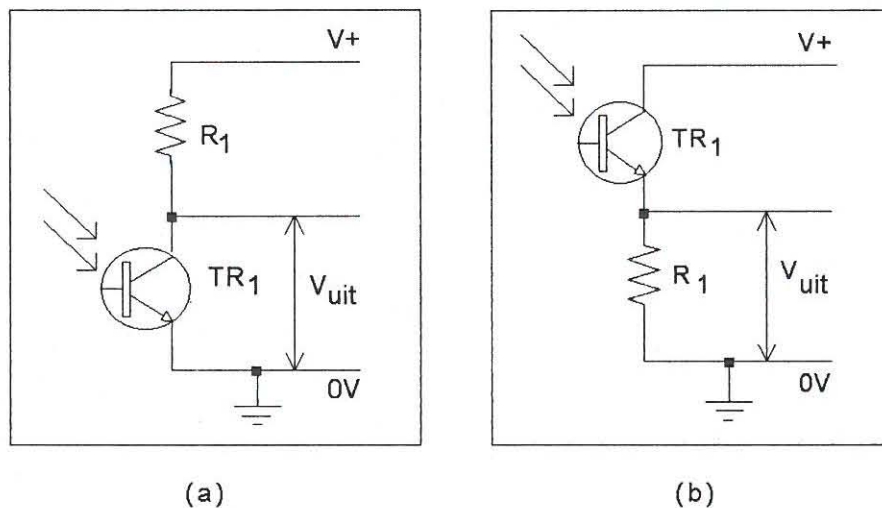
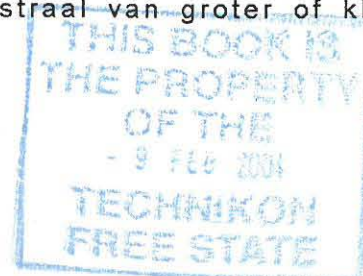


Fig. 2.4 - Fototransistor konfigurasies.

2.5.1 LENSE VIR FOTO- EN LIGEMISSIEDIODES

Fotodiodes en fototransistors gebruik lense om 'n inkomende divergerende of 'n breë ligbundel te fokus op 'n klein sensitiewe gebied. Netso kan die lig van 'n klein uitstralingsgebied gefokus word deur middel van 'n lens in 'n straal van groter of kleiner divergensie.



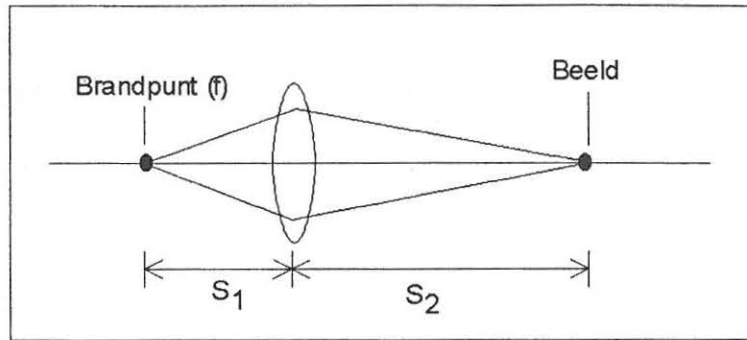


Fig. 2.5 - Eenvoudige lens voorstelling.

Die posisie van die beeld (S_2) word gegee deur die lensformule, en deur die definisies in Figuur 2.5 te volg is:

$$\frac{1}{S_2} = \frac{1}{f} - \frac{1}{S_1} \quad (2-5)$$

Indien 'n bron (S_1) op die brandpunt, f , van die lens geplaas word, word 'n parallelle of 'n gekollimeerde ligstraal verkry wat korrespondeer met $S_2 = \infty$. Aangesien die ligemissiediode se uitstralingsgebied nie 'n punt is nie, alhoewel klein, sal die straal in 'n mate divergeer selfs met die beste moontlike plasing [26, p. 88].

Fotodetektors met lense word dikwels gemonteer in 'n uitgeholde houer wat verspreide lig sal blokkeer (sien Figuur 2.6). Dit is belangrik om die effek van algemene kamerbeligting te verminder. In die meeste optiese stelsels is sensitiwiteit 'n kleiner probleem as agtergrondsteurings aangesien die versterker wins baie hoog gemaak kan word.

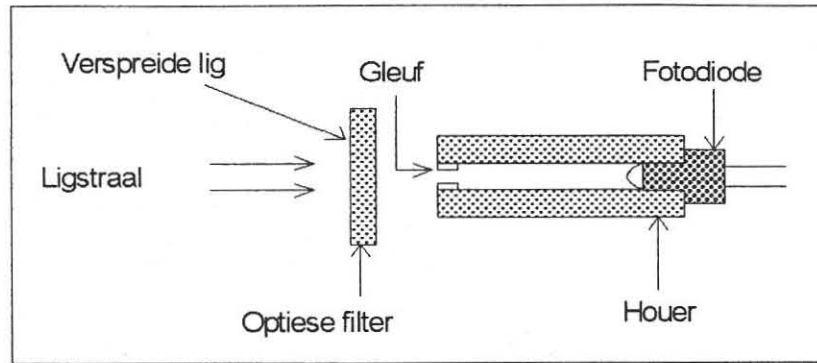


Fig. 2.6 - Uitgeholde houer met fotodetektor.

2.6 MEETSTELSELS

Die meting van 'n motor se assnelheid word tradisioneel gedoen met behulp van 'n tagometer, waar die snelheid direk op 'n analoge skaal weergegee word. Versnelling is 'n parameter wat soms benodig word, maar daar is slegs enkele transduseerders wat dié parameter kan verskaf.

Die rotasie-beweging van 'n liggaam kan gedefinieer word as die beweging rondom 'n vaste as. Die veranderlikes wat gewoonlik gebruik word om die rotasie-beweging te beskryf is draaimoment; hoekversnelling, α ; hoeksnelheid, ω ; en hoekverplasing, θ [11, p. 142].

2.6.1 TAGOMETER

Tagometers is elektromeganiese toestelle wat meganiese energie omskakel in elektriese energie. Die toestel werk hoofsaaklik as

'n generator wat 'n analoog uitsetsein, gewoonlik 'n spanning, lewer wat direk eweredig is aan die hoeksnelheid [9, p. 37]. Tagometers wat in beheerstelsels gebruik word se uitset is gewoonlik 'n gelykstroomspanning wat gebruik kan word om die assnelheid van 'n motor weer te gee, of om 'n snelheid-terugvoersein te verskaf vir spoedbeheer of vir stabilisering [6, p. 126]. Die blokdiagram van 'n tipiese snelheidbeheerstelsel word in Figuur 2.7 getoon.

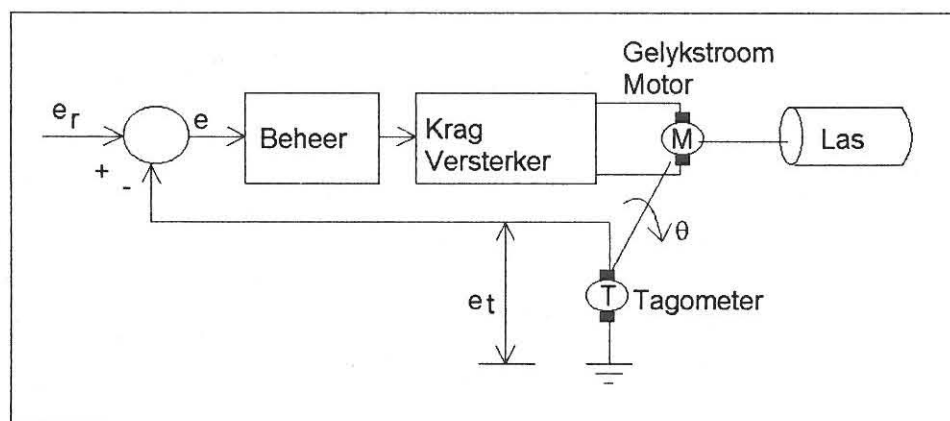


Fig. 2.7 - Beheerstelsel met tagometer terugvoer.

Die tagometer se uitset (e_t) word met 'n verwysingspanning (e_r) vergelyk. Die verwysingspanning stel die verlangde snelheid van die stelsel voor. Die verskil tussen die twee seine (e) word versterk en toegepas op die motor beheerstelsel sodat die verlangde snelheid bereik kan word.

Die mees tradisionele gebruik van 'n tagometer is egter om 'n visuele snelheidslesing van 'n roterende as te verskaf. Tagometers wat vir hierdie doel gebruik word, word gewoonlik direk aan 'n voltmeter verbind wat in revolusies per minuut gekalibreer is. Gewoonlik het tagometers 'n ysterkern rotor terwyl

die magnetiese veld deur 'n permanente magneet voorsien word. Die wikkelings op die rotor word aan die kommutatorsegmente gekoppel, en die uitsetspanning word geneem vanaf 'n paar borsels wat op die kommutatorsegmente druk [11, p. 170]. Aangesien die borsels en die kommutator onderhoud benodig en 'n beperkte leeftyd het, is daar ook 'n borsellose konfigurasie beskikbaar. In hierdie geval word kommutasie verkry deur middel van optiese of magnetiese enkodeerders.

2.6.2 WISKUNDIGE MODEL VIR TAGOMETERS

Die basiese karakteristiek van 'n tagometer is dat die uitsetspanning proporsioneel tot die rotorsnelheid is. Die dinamika van 'n tagometer kan dus voorgestel word deur die vergelyking,

$$e_t(t) = K_t \frac{d\theta(t)}{dt} = K_t \omega(t) \quad (2-6)$$

waar $e_t(t)$ die uitsetspanning is, $\theta(t)$ die rotor verplasing in radiale, $\omega(t)$ die rotor snelheid in rad/sek, en K_t die tagometerkonstante in V/rad/sek. Die waarde van K_t word gewoonlik gegee as 'n katalogus parameter in volts per 1000 rpm (V/krpm). 'n Tipiese waarde vir K_t is 6 V/krpm. Vergelyking (2-6) ignoreer die rimpelfrekwensie in die tagometer se uitsetspanning. Hierdie rimpelfrekwensie word veroorsaak deur die borsels wanneer hulle van een kommutasiesegment na die volgende oorgaan, deur

anker-eksentrisiteit of enige ander hoë frekwensie bron, soos elektromagnetiese induksie.

Die oordragfunksie van 'n tagometer word verkry deur die Laplace transforme te neem aan beide kante van vergelyking (2-6). Die volgende vergelyking word verkry,

$$\frac{E_t(s)}{\Theta(s)} = K_t s \quad (2-7)$$

waar $\Theta(s)$ die rotor verplasing is en beskou word as die inset, en die spanning $E_t(s)$ as die uitset [11, p. 172].

2.6.3 TWEESKYF SENSOR

Dit is 'n komplekse probleem om 'n spanning te verkry wat direk eweredig aan die versnelling van die as is, met weglaatbare fase vervorming en lae ruis inhoud. Hierdie versnellingseine is van belang in die studie van transiënt-gedrag in alle tipes roterende masjiene. Gelykstroom tagometers is tradisioneel gebruik, maar genereer ruis of vervorming waarvan die frekwensie direk eweredig aan die spoed is. Die ruis kan nie teen lae snelhede uitgefilter word nie, aangesien die verlangde sein vervorm sal word.

As 'n nie-deurskynende skyf, met deurskynende gleuwe op die rand van die skyf, gemonteer word op die as wat nagevors word, sal 'n ligstraal, met 'n wydte kleiner as die afstand tussen twee

opeenvolgende gleuwe, onderbreek word wanneer die skyf roteer en die gleuwe deur die ligstraal se pad beweeg. Indien die uitgestraalde lig op 'n fotoëlektriese toestel val, sal 'n reeks elektriese pulse opgewek word - teen 'n tempo eweredig aan die spoed van rotasie. Die pulse kan gebruik word om 'n analoge snelheidsein te verteenwoordig. Die fundamentele rimpelfrekwensie sal gelyk wees aan die herhalings tempo van die pulse en soos in die geval van die gelykstroom tagometer val dit na zero saam met die spoed. Die skyf moet deur een gleuftoestand roteer tussen elke transmissie van bruikbare inligting en dit kan 'n aansienlike hoek wees.

Die probleem is oorkom deur die gebruik van 'n tweede skyf, identies tot die eerste, sodat beide skywe die ligstraal onderbreek, soos getoon in Figuur 2.8. Dit is belangrik dat die wydte van die ligstraal nou groter moet wees as een gleuftoestand sodat al die gleuf samevallings teen enige hoekposisie opgetel kan word. Die tweede skyf word aangedryf teen 'n konstante snelheid in die teenoorgestelde rigting as die eerste [21, p. 1981].

Neem aan dat elke skyf 180 gleuwe het en dat die konstante spoed skyf teen 3000 omwentelings per minuut aangedryf word. Wanneer die veranderlike spoed skyf stilstaan is die puls frekwensie 9kHz. Indien die veranderlike spoed skyf versnel word tot 3000 omwentelings per minuut is die frekwensie 18kHz. Na prosessering bestaan die snelheidsein uit 'n gelykstroom waarde eweredig aan die snelheid van die veranderlike spoed skyf. Die minimum rimpelfrekwensie is nou bo die benodigde bandwydte en kan maklik verwyder word deur filtrering. Die

minimum frekwensie van transmissie van inligting oor die posisie van die veranderlike spoed skyf is verhoog van zero tot 9kHz.

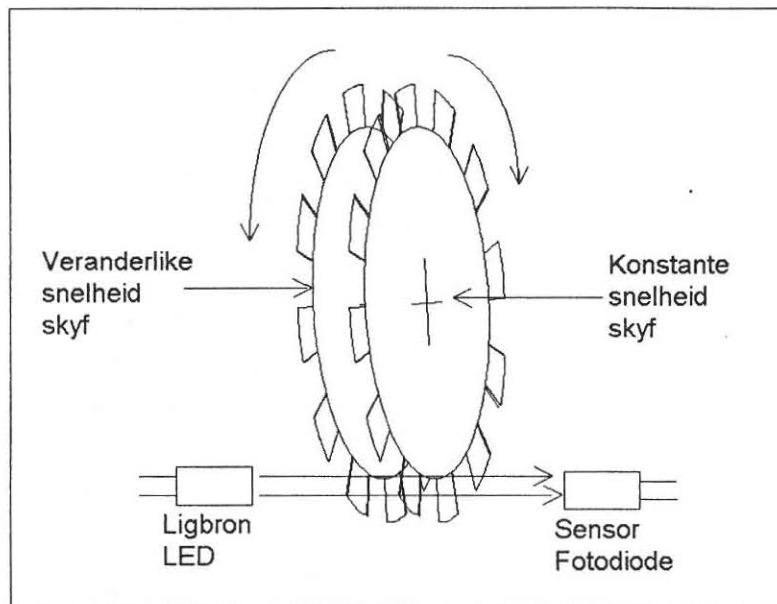


Fig. 2.8 - Tweeskyf sensor.

Alhoewel die toestel daarin slaag om die inherente rimpelfrekwensie te verskuif na 'n geskikte waarde, sodat die verwydering daarvan geen probleem is nie, is dit onvermydelik dat dit ruis sal veroorsaak as gevolg van hoekfoute in die plasing van die gleuwe.

'n Tweede tipe ruis word veroorsaak deur die eksentriese montering van die skyf. Vir 'n skyf wat roteer teen 3000 omwentelings per minuut sal hierdie ruis 'n frekwensie van 50Hz hê. Die frekwensie van 50Hz word verkry deurdat die skyf 3000 omwentelings per minuut maak met een afwyking vir elke omwenteling as gevolg van die eksentriese montering van die skyf. Wanneer beide skywe roteer sal die ruis as gevolg van die twee tipes foute saamsmelt en 'n gekompliseerde ruisspektrum

lewer. Addisionele ruiskomponente kan deur meganiese vibrasie en foutiewe konstante spoed rotasie veroorsaak word.

Daar is 'n metode om van die belangrikste ruiskomponente te verminder sonder om die bandwydte te beïnvloed. Die metode behels die gebruik van meer ligsensitiewe selle en bykomstige kringe. Byvoorbeeld, ruis veroorsaak deur skyf eksentrisiteit, onewe harmonieke en die fundamenteel van die kumulatiewe gleufposisie foute, kan teoreties uitgeskakel word deur die byvoeging van 'n tweede optiese stelsel. Die stelsel word 180° vanaf die eerste geplaas en lewer 'n tweede stel pulse van spanning en tyd-produkte. Die komponente van ruis sal uitkanselleer in die sommering van die twee stelle pulse. Twee verdere optiese stelsels kan bygevoeg word, 90° weg van die eerste stel, sodat die ewe harmonieke ook uitgekanselleer word.

Dit is onvermydelik dat daar ruiskomponente binne en bokant die frekwensiestrek van belang sal oorbly. Die ruis kan deur sorgvuldige konstruksie beperk word, sodat wanneer die sein aan laeruis elektroniese kringe verbind word, die toestel lae ruis, wye bandwydte snelheid- en versnellingseine lewer. Hoe laer die inherente ruis opgewek in die stelsel, hoe beter sal die seinruisverhouding vir 'n gegewe bandwydte wees.

'n Ander tipe ruis is die lae frekwensie veranderings in die sensitiwiteit van die snelheid en versnellingsuitsette. Die foute kan veroorsaak word deur veranderings in snelheid van die konstante spoed skyf of deur dryf in die elektroniese kringe. Indien die skyf aangedryf word deur 'n sinchrone reluktansie of 'n histerese motor wat gevoer word vanaf die 50Hz toevoer, is die

snelheid stabiliteit afhanklik van die frekwensie stabiliteit van die toevoer stelsel, en na verwagting dus redelik stabiel.

2.6.4 AS-ENKODEERDER

Digitale enkodeerders word hoofsaaklik gebruik om 'n absolute waarde van die posisie van 'n roterende as in digitale vorm te verkry [3, p. 197]. 'n Skyf of drom met 'n digitaal gekodeerde patroon word geheg aan die as en die posisie word vasgestel deur borsels of foto-elektroniese transduseerders, soos getoon in Figuur 2.9.

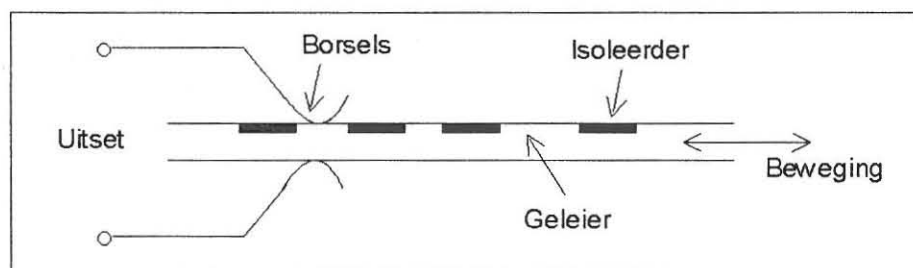


Fig. 2.9 - Borsel tipe enkodeerder.

In 'n optiese enkodeerder bestaan die skyf uit 'n deursigtige laag met ondeursigtige segmente op die oppervlak of uit 'n weerkaatsende laag met nie-weerkaatsende segmente. Wanneer 'n ondeursigtige segment in die eerste geval tussen die ligbron en die sensor is, word die ligstraal onderbreek. Wanneer die ondeursigtige segment nie tussen die bron en die sensor is nie, word die straal voltooi. In die tweede geval word lig geweerkaats tussen die ligbron en die sensor, wat die straal voltooi en onderbreek, (sien Figuur 2.10) [1, p. 64].

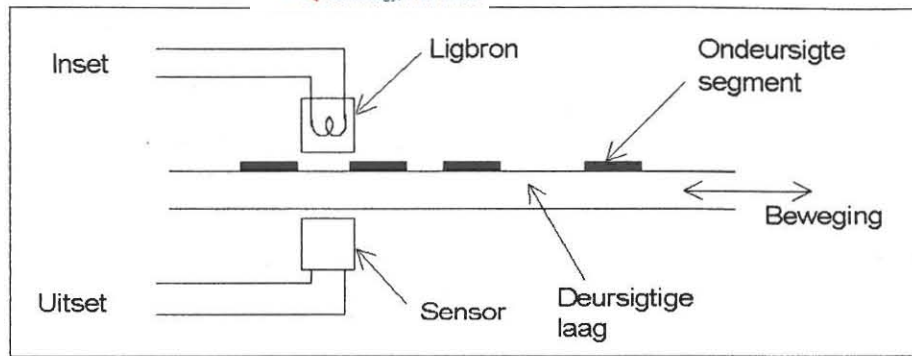


Fig. 2.10 - Optiese enkodeerder.

Die enkodeerder skyf in Figuur 2.11 bestaan uit verskeie segmente waarvan party deurskynend is en die ander ondeurskynend. Dit vorm 'n 4 bis binêre kode wat in elektriese vorm omgeskakel kan word deur vier bron-sensor foto-elektroniese samestellings te gebruik. In die praktyk gebruik optiese enkodeerders kodes soos byvoorbeeld die tien syfer Gray sikliese kode wat 1024 tellings per revolusie gee - met 'n akkuraatheid van ± 1 telling. Hierdie tipe enkodeerders, wat 'n digitale uitsetsein lewer, staan bekend as absolute enkodeerders [11, p. 172]. Absolute enkodeerders word gewoonlik gebruik waar daar moontlik data verliese kan wees as gevolg van kragonderbrekings.

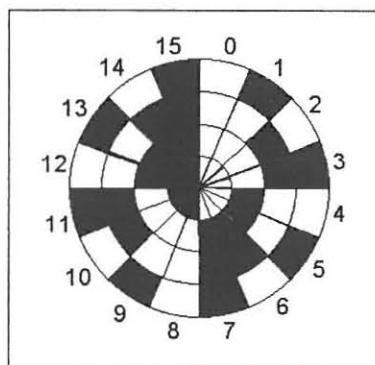


Fig. 2.11 - Vier bis binêre enkodeerder.

'n Inkrementele enkodeerder lewer 'n puls vir elke inkrement van resoluksie, maar maak nie 'n onderskeid tussen die verskillende inkremente nie. Optiese inkrementale as-enkodeerders het die volgende eienskappe [9, p. 7].

1. Die uitsetseine word verkry deur optiese metodes.
2. Die uitsetseine is inkrementaal en gee nie 'n absolute aanduiding van die hoekposisie van die transduseerder se as nie.
3. Die transduseerder is van die roterende tipe met uitsetseine afhanklik van die hoekposisie van die transduseerder se as.

'n Tipiese inkrementale enkodeerder het vier basiese onderdele; 'n ligbron, 'n roterende skyf, 'n stilstaande masker en 'n sensor, soos voorgestel in Figuur 2.12 [11, p. 172].

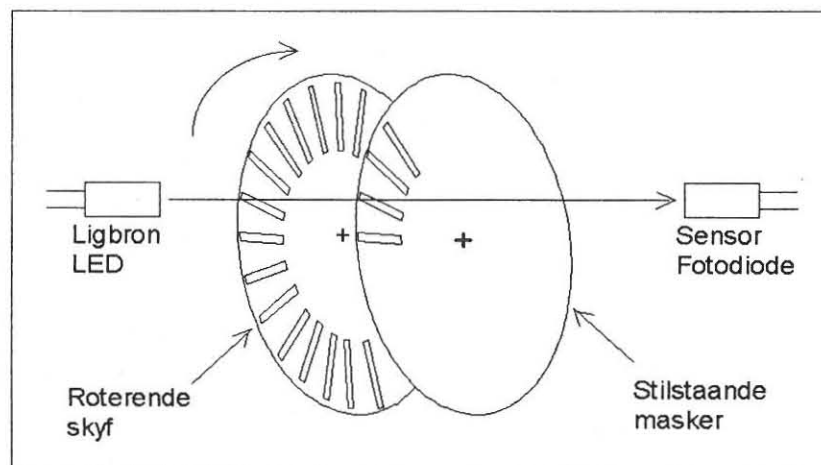


Fig. 2.12 - Tipiese inkrementale enkodeerder.

Die skyf bestaan uit alternerende deursigtige en ondeursigtige segmente. Enige paar van die segmente stel 'n inkrementale

periode voor. Die masker word gebruik om die ligstraal tussen die ligbron en die foto sensor, wat agter die masker geplaas is, deur te laat of te blokkeer. Vir enkodeerders wat ongeveer 1000 pulse per omwenteling lewer, word 'n veelvoudige gleuf masker gebruik vir maksimalisering van die sluiters lig ontvang.

Moderne optiese as-enkodeerders bestaan uit 'n dun, liggewig skyf wat gegleuf is op die omtrek en wat aan die naaf van die enkodeerder vasgeheg is, en kan maklik aan 'n roterende as gekoppel word [3, p. 197]. Die posisie van die gleuwe word bepaal deur die optiese bron en sensorsamestelling. Die aantal pulse opgewek per revolusie is gelyk aan die aantal gleuwe. Drie van die bron-sensorsamestellings word gebruik, twee word so geplaas dat hulle uitsette 90° uit fase tot mekaar is en die derde word opgestel relatief tot 'n addisionele gleuf op die skyf om 'n verwysingspuls met elke omwenteling te lewer.

Deur die lei- en volgrande van die twee pulsreekse te tel, word 'n viervoudige vermeerdering in die hoekresolusie verkry. Wanneer saam met mikroprosesseerders gebruik, word die verwysingspuls gebruik om die stelsel te inisialiseer, die tellers te herstel en as sinchronisasiepuls. Die rigting van rotasie word bepaal deur monitering van die relatiewe fasevoorloop of fasenaloop van die twee pulsreekse deur 'n rigtings sensitiewe logikakring.

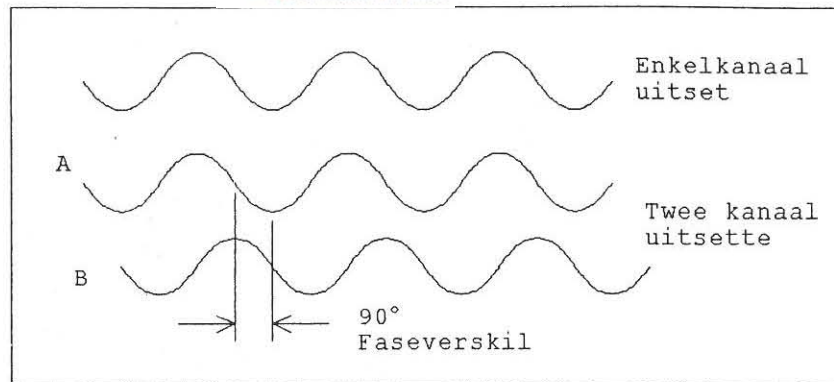


Fig. 2.13 - Tipiese sinusvormige uitsetgolfvorme.

Figuur 2.13 toon 'n enkelkanaal uitset en die kwadratuur uitsette met sinusvormige golfvorme. Die golfvorme van die sensoruitsette is normaalweg driehoekig of sinusvormig afhangend van die resoluëie benodig. 'n Schmitt snellerkring word gebruik om 'n veranderende spanning om te skakel in 'n vierkantsgolf wat aanpasbaar met digitale logika is (sien Figuur 2.14).

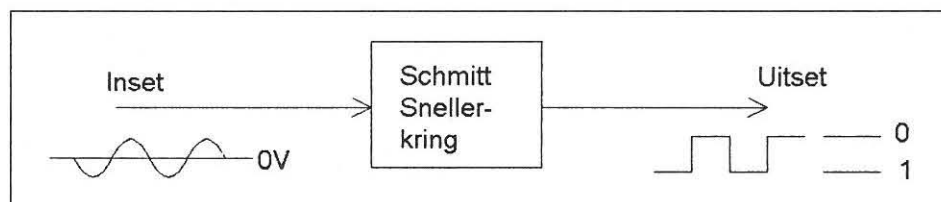


Fig. 2.14 - Inset/uitsetseine van 'n Schmitt snellerkring.

Die elektriese uitset wat direk van die optiese stelsel van die transduseerder verkry word is klein (tipies ongeveer 40mV wgk) en moet dus voor gebruik versterk word. Dit is ook ongewens om lae spannings oor 'n lang afstand te stuur as gevolg van moontlike steurings of ander probleme. Die klein seine word in die transduseerder versterk voordat dit uitgebring word by die transduseerder se terminale. Transistor-transistor-logika (TTL)

aanpasbare uitsetspannings word gelewer om die moontlikheid van moontlike steurings van nabygeleë elektriese kringe te minimaliseer:

2.7 MIKROREKENAARS EN DATAVERKRYGING

Persoonlike rekenaars maak dit moontlik om voordeel te trek uit die buigsaamheid, krag en doeltreffendheid van dataverkryging en beheer deur middel van rekenaars. Die TTL uitsetseine van die sensor eenheid kan nie direk in 'n rekenaar ingelees word nie en 'n analoog-na-digitale omsetterkaart (bv. PC-30C) word benodig sodat die seine in 'n digitale formaat omgeskakel kan word. Nadat die seine omgeskakel is na digitale seine kan dit in die rekenaar ingelees word vir verwerking.

Die PC-30C kaart is 'n lae koste, hoë akkuraatheid kaart vir IBM PC/XT/AT en aanpasbare rekenaars en word gebruik om die analoog uitsette van die enkodeerderstelsel in die rekenaar in te lees vir verwerking. Die PC-30D is ontwerp vir die gebruik met 'n IBM AT of 386/486 of Pentium persoonlike rekenaar. Die PC-30C se maksimum frekwensie van analoog-na-digitale omset is 100kHz en die van die PC-30D is 200kHz. Kenmerke van die kaarte is dat dit 16 analoog insetkanale het, twee 12-bis digitaal-na-analoog omsetters, twee 8-bis digitaal-na-analoog omsetters en 24 digitale inset en uitset lyne besit.

2.7.1 ANALOOG INSETTE EN UITSETTE - I/O

Die PC-30 kaart het 16 analoog insette met selekteerbare strekke en hoë insetimpedansies. Die strekke is vanaf -5V tot +5V; -10V tot +10V en 0V tot +10V. Data kan oorgedra word vanaf die kaart na 'n rekenaar deur geprogrammeerde I/O, onderbrekingsgedrewe I/O of direkte geheue-toegang. Die omsettingstempo kan gestel word deur 'n interne klokverdeler wat deur 'n kristal beheerde ossillator gedryf word, of van 'n eksterne klok inset. Die interne klok kan geprogrammeer word in die strek van 0.001Hz (in die geval van 'n PC-30D kaart) tot 200kHz.

2.7.2 DIGITALE INSETTE EN UITSETTE

Die kaart het 24 digitale inset en uitset lyne wat in drie poorte gerangskik is. Elke poort kan gebruik word vir inset of uitset en is TTL aanpasbaar. Poort C kan gebruik word vir aansluitbevestiging vir poorte A en B. 'n Onverbonde 16-bis tydhouer/teller is ook beskikbaar. Dit kan gebruik word om frekwensies of pulse op te wek, gebeurlikhede te tel, of frekwensie of pulsperiodes te meet.

2.8 OPSOMMING

Alle beheer- en meetstelsels het tot 'n sekere mate transiënte voordat die bestendige staat bereik word. Transiënte is

kortstondige verskynsels en word gedefinieer as daardie gedeelte van die respons wat na zero gaan as die tydsduur toeneem. Transiënte is as gevolg van traagheid, massa en induktansie wat nie volledig in fisiese stelsels vermy kan word nie. Sensors en transduseerders is toestelle wat veranderings in 'n fisiese stimulus bespeur en dit verander in 'n sein wat gemeet en geanaliseer kan word. Frekwensie-na-spanning omsetters lewer 'n uitsetspanning wat eweredig is aan die insetfrekwensie. Die uitsette van sensors en frekwensie-na-spanning omsetters kan met behulp van 'n PC-30 kaart in 'n rekenaar ingelees word vir verwerking.

Tagometers is elektromeganiese toestelle wat meganiese energie omskakel in elektriese energie, en verskaf soms 'n visuele snelheidslesing van 'n roterende as. Tagometers genereer ruis as gevolg van die rimpelfrekwensie wat deur die borsels veroorsaak word wanneer hulle van een kommutasiesegment na die volgende oorgaan. Die probleem kan oorkom word deur 'n tweeskyf sensor te gebruik. Die een skyf word teen 'n konstante snelheid in die teenoorgestelde rigting as die tweede aangedryf. Ruis kan in hierdie stelsel uitgefiltreer word. Indien meer as een optiese stelsel gebruik word, kan die ruis komponente verminder word sonder om die bandwydte te beïnvloed.

Daar is twee tipes as-encodeerders beskikbaar, nl. absolute as-encodeerders en inkrementele as-encodeerders. Absolute as-encodeerders gee 'n digitale uitsetsein van die posisie van 'n roterende as terwyl inkrementele encodeerders 'n puls lewer vir elke inkrement van revolusie.

Die PC-30C kaart is geskik vir gebruik met IBM PC/XT/AT en aanpasbare rekenaars. Die kaart besit 16 analoog insetkanale, twee 12-bis digitaal-na-analoog omsetters, twee 8-bis digitaal-na-analoog omsetters en 24 digitale inset en uitset lyne. Die PC-30 kaart word gebruik om die analoog uitsette van die enkodeerder in die rekenaar in te lees as digitale seine vir verwerking en analisering.

HOOFSTUK 3

OPERASIONELE VERSTERKERS AS SEINPROSESSEERDERS

3.1 INLEIDING TOT OPERASIONELE VERSTERKERS

Die uitset van die sensoreenheid, soos beskryf in Paragrafe 2.6.3 en 2.6.4, bestaan uit 'n pulsreeks met variërende frekwensie. Die variërende frekwensie kan egter nie in 'n rekenaar ingelees word nie en 'n frekwensie-na-spanning omsetter word benodig. Die doel van die omsetter is om die variërende frekwensie om te skakel in 'n eenheid wat die rekenaar kan verstaan, byvoorbeeld 'n variërende spanning. Vir die omskakelingsproses word van filters, integreerders en differensieerders gebruik gemaak. Dit behels dikwels 'n operasionele versterker vir verwerking van die uitsetseine van die sensoreenheid.

Van al die halfgeleier toestelle wat die afgelope jare in die elektroniese wêreld bekend gestel is, is die operasionele versterker een van die belangrikste en veelsydigste. Die operasionele versterker is 'n hoë wins gelykstroom differensiaalversterker met 'n hoë inset- en lae uitsetimpedansie [12, p. 119].

Passiewe komponente, soos weerstande, kapasitors en induktors, kan in seinprosesseringstroombane gebruik word. Met passiewe stroombane word 'n sekere gedeelte van die sein egter verloor of verbruik, terwyl aktiewe stroombane die amplitude van die insetsein

behou of versterk. Operasionele versterker stroombane is goedkoper, makliker om te gebruik, meer effektief, benodig minder eksterne komponente, benodig kleiner spasies en is baie ligter as soortgelyke stroombane wat diskrete komponente gebruik.

3.2 BASIESE OPERASIONELE VERSTERKER

Die skematiese diagram van 'n operasionele versterker word in Figuur 3.1a getoon en die ekwivalente stroombaan daarvan in Figuur 3.1b [14, p. 501]. Operasionele versterkers het 'n differensiële inset met spannings V_1 en V_2 wat onderskeidelik op die omkeer en nie-omkeer insetterminale toegepas word.

Die uitsetspanning V_o van 'n ideale differensiële inset operasionele versterker is eweredig aan die verskil in spannings tussen die twee seinbronne, soos aangedui deur

$$V_o = A_o (V_2 - V_1) \quad (3-1)$$

waar A_o die ooplus wins is [26, p. 11].

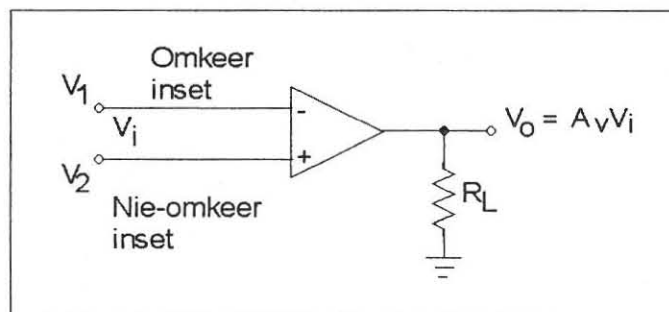


Fig. 3.1(a) - Basiese operasionele versterker.

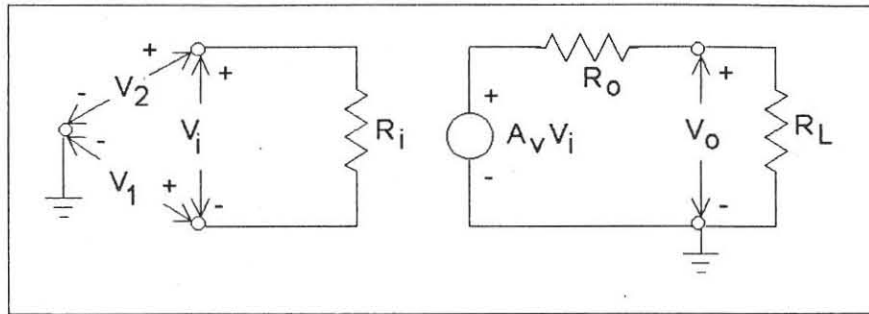


Fig. 3.1(b) - Ekwivalente stroombaan van operasionele versterker.

3.3 INTEGRERDER

Die integreerder is 'n kring wat die wiskundige bewerking van integrasie uitvoer en lewer 'n uitset wat eweredig is aan die integraal van die insetsein oor tyd [2, p. 223]. Die integreerder word gebruik in berekenings, seinprosessering en seinopwekking toepassings en maak gewoonlik gebruik van 'n operasionele versterker in die omkeer konfigurasie [23, p. 213].

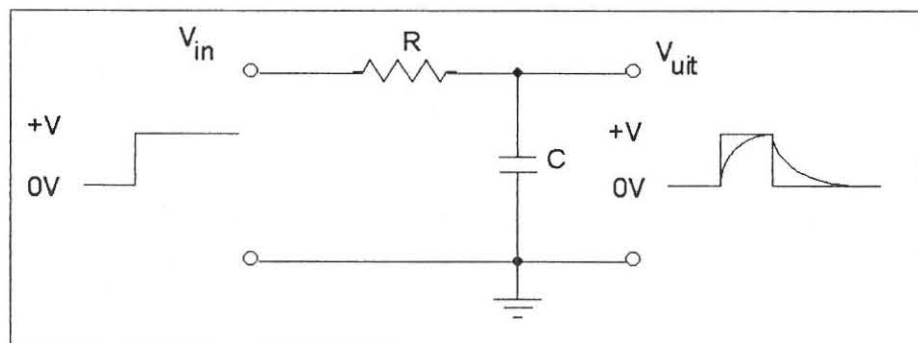


Fig. 3.2 - Passiewe integreerder.

Figuur 3.2 toon 'n basiese passiewe RC integreerder waar die uitset oor die kapasitor geneem word. Indien die insetspanning konstant

positief bly word 'n uitsetspanning verkry wat opwaarts swaai [26, p. 18]. Die spanning oor die kapasitor neem 'n sekere tydskuur om te styg en is afhanklik van die RC tydkonstante. Aangesien die stroom verminder namate die kapasitor laai, sal die spanning oor die kapasitor styg teen 'n eksponensiële tempo, soos getoon word deur die golfvorm in Figuur 3.2 [7, p. 63]. Wanneer die insetpuls na zero val ontlai die kapasitor teen 'n eksponensiële tempo. Die insetspanning word verdeel oor die weerstand en die kapasitor. As gevolg van verliese deur die kring is die uitsetspanning kleiner as die insetspanning. Die uitsetspanning is ongeveer gelyk aan die integraal van die insetsein mits die insetpulsduur korter as een tydkonstante ($T=RC$) is.

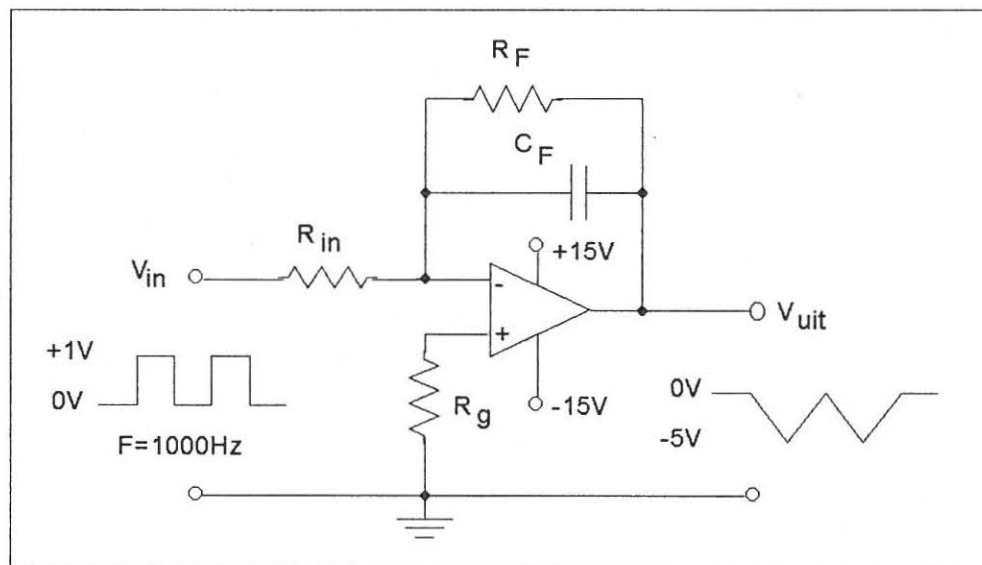


Fig. 3.3 - Praktiese operasionele versterker integreerder.

Figuur 3.3 toon 'n praktiese operasionele versterker integreerder. In die integreerder is kapasitor C_F die terugvoerelement. Die terugvoerstrom moet gelyk wees aan die insetstroom en daarom voorsien die operasionele versterker 'n lineêr stygende

uitsetspanning. Wanneer die insetpuls val na zero, is die dalende uitsetspanning ook lineêr. Die totale resultaat is dus 'n driehoekgolf uitsetspanning. Aangesien die omkeer inset gebruik word, sal die uitset golfvorm in 'n negatiewe rigting gaan ten opsigte van 'n positiewe insetpuls. Indien 'n terugvoerweerstand, R_F , in parallel met die terugvoerkapasitor geplaas word, word uitset versadiging voorkom en 'n praktiese integreerder verkry met verminderde ruis en beter stabiliteit. Die afsnyfrekwensie, f_C , van die integreerder kan met behulp van vergelyking (3-2) bepaal word.

$$f_C = \frac{1}{2\pi R_F C_F} \text{ Hz} \quad (3-2)$$

Indien die insetfrekwensie laer as f_C is, sal die kring ophou om soos 'n integreerder te werk en sal dit soos 'n omkeerversterker met 'n spanningswins, wat deur die verhouding $\frac{R_F}{R_{IN}}$ bepaal word, funksioneer. Die integreerder frekwensiekenkromme in Figuur 3.4 toon die gebiede aan waar die kring as 'n integreerder werk, en waar die operasionele versterker as 'n omkeer versterker werk. Hierdie gebiede word deur die afsnyfrekwensie, f_C , gedefinieer.

Die kapasitiewe reaktansie, X_C , verander met frekwensie. X_C vergroot teen lae frekwensies wat veroorsaak dat minder van die sein teruggevoer word en die uitsetspanning verhoog. Indien die frekwensie verhoog verklein X_C en meer van die sein word teruggevoer en die uitsetspanning verminder. Aangesien X_C verander met frekwensie reageer die integreerder soos 'n onderdeurlaatfilter [8, p. 54]. Die kriteria vir 'n goeie integreerder

is dat die laagste insetfrekwensies bo die -3dB punt, f_c in Figuur 3.4, moet wees [5, p. 39].

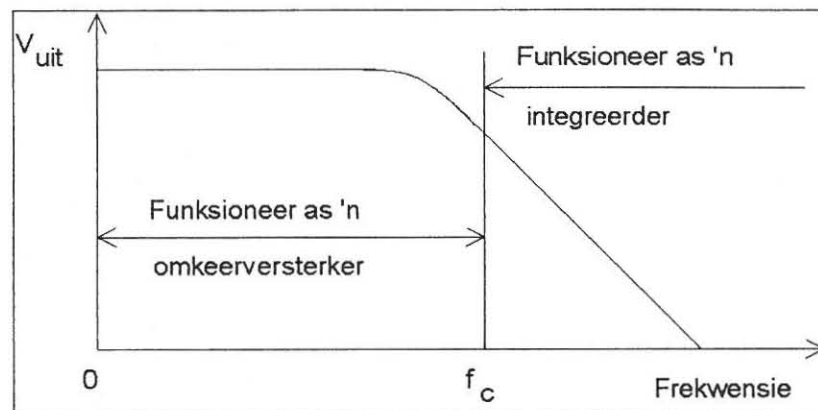


Fig. 3.4 - Integreerder kenkromme.

3.4 DIFFERENSIEERDER

'n Differensieerder is funksioneel die omgekeerde van 'n integreerder. 'n Differensieerder is 'n toestel wat die wiskundige funksie van differensiasie uitvoer en voorsien 'n uitsetspanning wat eweredig is aan die helling van die insetspanning. Toepassings van differensieerders is onder andere om lei- en volgrande van 'n reghoekgolf te identifiseer [2, p. 244]. By aktiewe differensieerders is die insetkomponent 'n kapasitor en die terugvoercomponent 'n weerstand.

Wanneer die insetspanning verander sal die kapasitor laai en ontlai. As gevolg van die effektiewe grond op die operasionele versterker inset sal die stroom deur die terugvoerweerstand vloei en 'n spanning opwek. Deur 'n operasionele versterker as differensieerder te gebruik kan die uitsetspanning, wat uit fase met

die insetspanning is, gelyk aan, of groter as die insetspanning gemaak word [7, p. 65]. Slegs veranderinge in die insetsein sal stroomvloei deur die insetkapasitor en die terugkoppelweerstand lewer (as gevolg van die kapasitor wat gelykstroom blokkeer). Die uitset van die differensieerder is gelyk aan die spanningsval oor die terugkoppelweerstand en is eweredig aan die negatiewe afgeleide van die insetsein. 'n Operasionele versterker differensieerder word in Figuur 3.5 getoon.

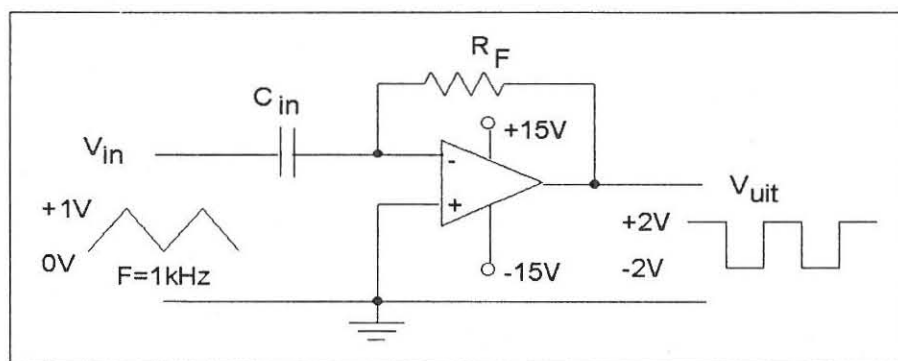


Fig. 3.5 - Operasionele versterker differensieerder.

Die nadeel van hierdie differensieerderkring is dat die wins van die operasionele versterker deur die verhouding R_F/X_C bepaal word. Die reaktansie van die kapasitor C_{in} neem teen hoë frekwensies af tot 'n lae waarde, wat veroorsaak dat die operasionele versterker dan 'n buitengewone hoë wins het. Die versterker kan tot versadiging gedryf word, wat tot ongewenste vervorming van die uitsetsein sal lei. Daarbenewens is hierdie kring gevoelig vir hoë frekwensie ruis.

Om dit in 'n meer praktiese kring te bowe te kom, word, soos aangetoon in Figuur 3.6, 'n weerstand R_2 in serie met die insetkapasitor C_1 geplaas om hoë frekwensie wins tot die verhouding R_1/R_2 te beperk [2, p. 248]. Die waarde van R_2 word

tipies ongeveer tien keer kleiner as die waarde van weerstand R_1 gemaak.

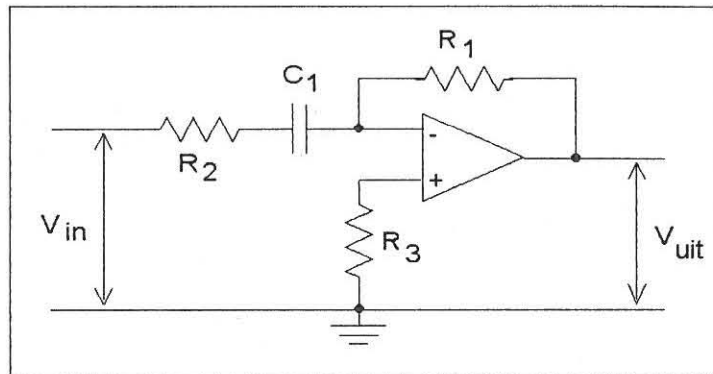


Fig. 3.6 - Praktiese operasionele versterker differensieerder.

Die differensieerder frekwensierespons kenkromme in Figuur 3.7 toon die gebiede waar die kring as 'n differensieerder funksioneer en waar die operasionele versterker as 'n omkeer versterker funksioneer. Hierdie gebiede word deur die afsnyfrekwensie f_c gedefinieer.

$$f_c = \frac{1}{2\pi R_1 C_1} \text{ Hz} \quad (3-3)$$

In praktiese differensieerder kringe word 'n verdere weerstand, R_3 (sien Figuur 3.6), met dieselfde waarde as weerstand R_1 , in die kring ingesluit om die uitset verskilspanning te balanseer.

Die voordeel van 'n operasionele versterker differensieerder relatief tot 'n RC differensieerder is dat die uitsetpuls van 'n lae impedansie bron afkomstig is. Die skerp snellerpuls wat deur die differensieerder gelewer word is baie bruikbaar om ander kringe

aan te dryf. Wanneer die insetsein verhoog in frekwensie sal X_C van die insetkapasitor daal en die uitsetsein sal verhoog. Sodoende kan die differensieerder ook gebruik word as 'n bodeurlaatfilter [8, p. 54].

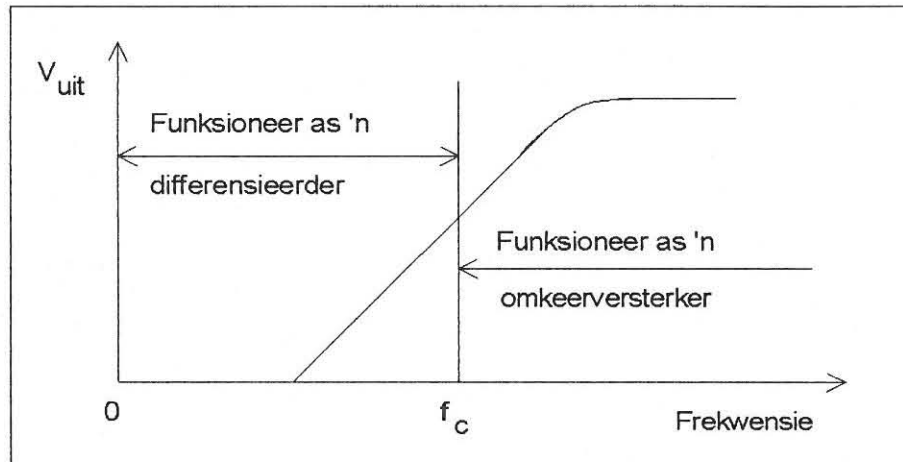


Fig. 3.7 - Differensieerder frekwensiekenkromme.

3.5 OPERASIONELE VERSTERKER FILTERS

Aktiewe filters bevat versterkers wat die ontwerp van 'n wye streek van oordragfunksies toelaat [19, p. 520]. Die byvoeging van 'n versterker tot die passiewe kring veroorsaak 'n oordragfunksie wat die vorm het van 'n verhouding van twee polinome. Daar is 'n aantal voordele wat geassosieer kan word met die gebruik van aktiewe netwerke teenoor passiewe netwerke, naamlik lae koste, kaskadering en wins [19, p. 525]. Tesame met die voordele is daar egter beperkings, soos byvoorbeeld die noodsaaklikheid van toevoerspannings, seinlimiete en frekwensiegrense.

Daar is twee hoof benaderings tot hoër orde aktiewe filter sintese:

1. Die kaskade benadering, waar die verlangde respons verkry word deur kaskadering van tweede en moontlik selfs eerste-orde filter stadiums.
2. Die RLC leersimulasie benadering, waar die verlangde vereistes verkry word deur gebruik te maak van aktiewe impedansie-omsetters, byvoorbeeld frekwensie afhanklike negatiewe weerstande, om 'n passiewe RLC filter te simuleer.

Die kaskade metode is egter die eenvoudigste en gewildste metode. Hierdie benadering genereer die verlangde oordragfunksie as 'n produk van die tweede en moontlik die eerste-orde oordragfunksies. Te danke aan die lae uitsetimpedansie van elke seksie is daar geen belasting tussen die verskillende stadiums nie. Elke seksie kan gesien word as geïsoleer van die ander en kan afsonderlik gestel word indien nodig [4, p. 146].

Die primêre voordeel van aktiewe filters is hul grootte en massa vir lae frekwensie toepassings en hul betroubaarheid [23, p. 284]. Figuur 3.8 toon die respons van ideale bodeurlaat, onderdeurlaat en banddeurlaat filters. 'n Ideale bodeurlaatfilter sal alle frekwensies bokant 'n afsnyfrekwensie f_c sonder verswakking deurlaat en laat geen frekwensies onder f_c deur nie. 'n Onderdeurlaatfilter doen die teenoorgestelde, en 'n banddeurlaatfilter laat slegs frekwensies binne $\Delta f_0/2$ vanaf 'n middelfrekwensie, f_0 , toe om deur te gaan. Nie-ideale (praktiese) filters het nie so 'n skerp oorgang by die afsnyfrekwensie nie. Die respons van goed ontwerpte realiseerbare filters neig egter na die ideale respons [26, p. 167].

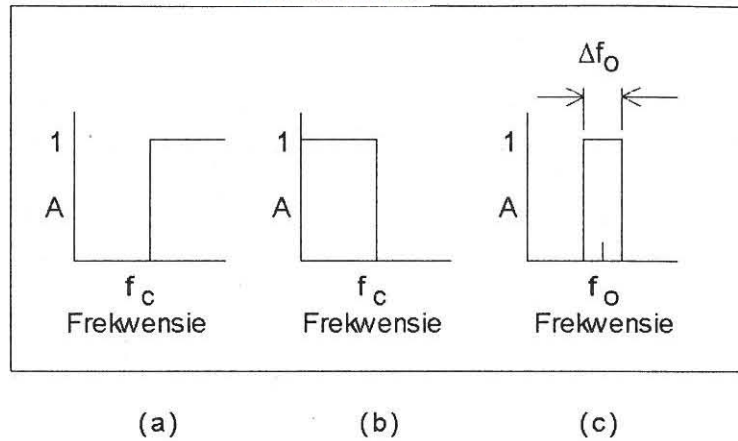


Fig. 3.8 - Frekwensie respons van ideale filters: (a)Bodeurlaat; (b)Onderdeurlaat; (c)Banddeurlaat.

3.6 ONDERDEURLAATFILTER

Onderdeurlaatfilters verrig 'n belangrike funksie by analoog-na-digitaal en digitaal-na-analoog omsetting [4, p. 156]. Die frekwensiestrek van die insetsein vir 'n analoog-na-digitale omsetter moet tot onder die helfte van die monsterfrekwensie begrens word. Die uitsetsein van 'n digitaal-na-analoog omsetter moet ook afgevlak word om die effek van diskrete kwantisering te voorkom. Beide take word bereik deur die gebruik van onderdeurlaatfilters wat genoegsame verswakking bied teen minder as die helfte van die monsterfrekwensie.

Figuur 3.9 toon die basiese aktiewe onderdeurlaatfilter. Die operasionele versterker word as 'n hoë insetimpedansie toestel met 'n wins van een gebruik. Die tipe filter staan as 'n eerste-orde filter bekend. Indien 'n skerper afsnyding vereis word, moet 'n hoër orde struktuur gebruik word.

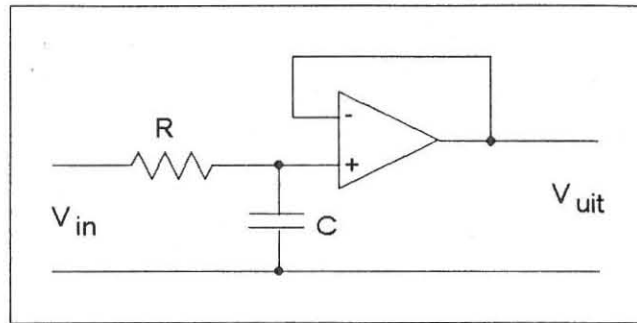


Fig. 3.9 - Basiese aktiewe onderdeurlaatfilter.

In die eenvoudige negatiewe terugkoppelkring van Figuur 3.10 neem die impedansie van die terugkoppelkomponent af namate die frekwensie toeneem. Die geslote wins van die kring is dus groter teen lae frekwensies. Die afsnyfrekwensie vir die filter word gegee deur,

$$f_c = \frac{1}{2\pi RC} \text{ Hz} \quad (3-4)$$

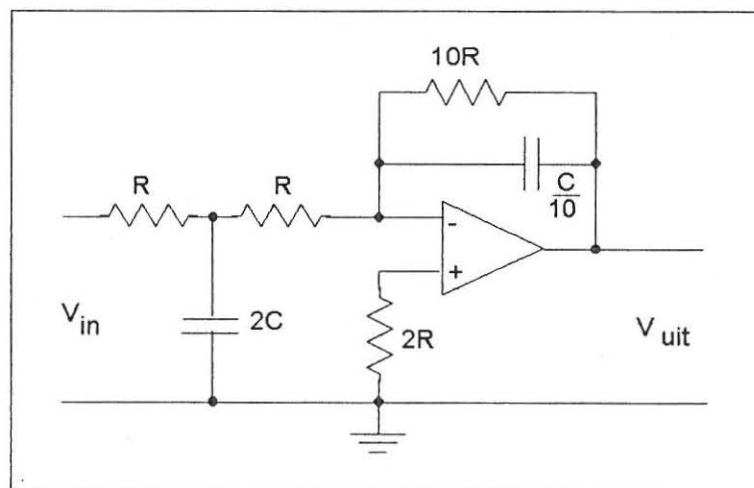


Fig. 3.10 - Onderdeurlaatfilter met negatiewe terugkoppeling.

Die afsnyhelling van die onderdeurlaatfilter word gewoonlik in terme van die dB/oktaaf verswakingsfaktor uitgedruk (sien Figuur 3.15). Die afsnypunt word geneem op die punt waar die uitset tot 'n waarde gelyk aan die halfkragpunt daal, d.w.s. -3dB [12, p. 143].

3.7 BODEURLAATFILTER

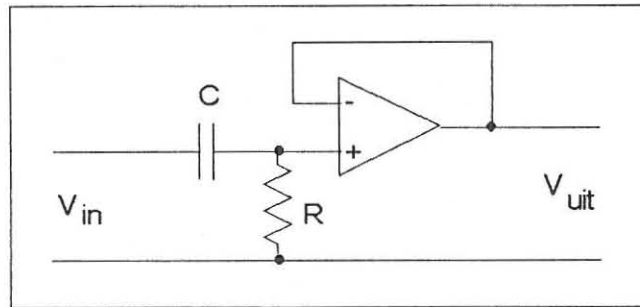


Fig. 3.11 - Basiese aktiewe bodeurlaatfilter.

Figuur 3.11 toon die basiese aktiewe bodeurlaatfilter en die operasionele versterker wat as 'n hoë insetimpedansietoestel met 'n wins van een gebruik word. Die frekwensierespons van 'n bodeurlaatfilter word in Figuur 3.12 getoon.

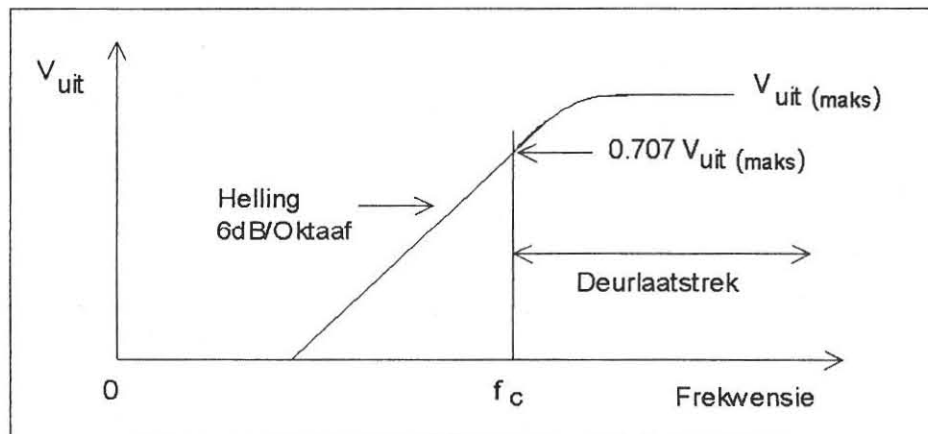


Fig. 3.12 - Bodeurlaatfilter frekwensierespons .

'n Aktiewe bodeurlaatfilter met terugkoppeling word in Figuur 3.13 getoon. Die reaktansie van die kapasitors neem af namate die frekwensie toeneem en die geslote lus wins van die kring word verhoog teen hoë frekwensies. 'n Weerstand word in serie met die insetleiding geplaas om die wins teen baie hoë frekwensies te beperk. Die afsnyfrekwensie word gegee deur vergelyking 3-4.

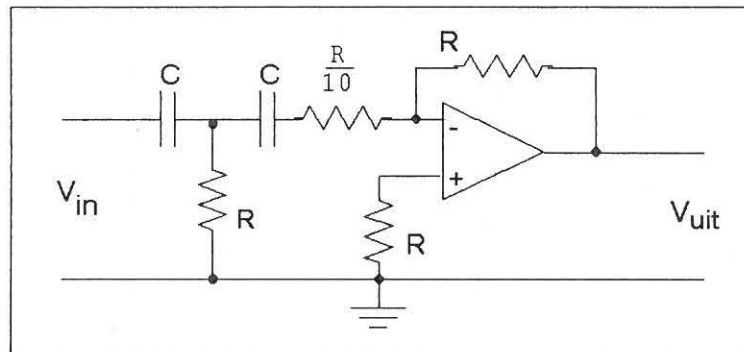


Fig. 3.13 - Bodeurlaatfilter met terugkoppeling.

3.8 NORMALISERING VAN FILTERS

Die basis vir normalisering van filters is die feit dat 'n gegewe filterrespons verskuif kan word na 'n ander frekwensiestrek deur die reaktiewe elemente te deel met 'n frekwensieskaalfaktor (FSF) [25, p. 2-1]. Die FSF is die verhouding van 'n verwysingsfrekwensie van die verlangde respons tot die ooreenkomstige verwysingsfrekwensie van die gegewe filter. Gewoonlik word die -3dB punte gekies as die verwysingsfrekwensie van 'n onderdeurlaat en 'n bodeurlaatfilter. Die FSF kan soos volg uitgedruk word:

$$FSF = \frac{\text{Verlangde verwysingsfrekwensie}}{\text{Bestaande verwysingsfrekwensie}} \quad (3-5)$$

Die FSF is 'n dimensielose waarde, dus moet die teller en die noemer in uitdrukking (3-5) dieselfde eenhede besit, gewoonlik radiale per sekonde.

3.9 BUTTERWORTH FILTERS

Die Butterworth benadering lewer 'n klas van filter wat 'n matige verswakkingshelling en aanvaarbare transiënt karakteristieke besit [25, p. 2-35]. Die elementwaardes is prakties en nie so krities soos in die meeste ander filtertipes nie. Die afronding van die frekwensieweergawe in die afsnygebied maak die tipe filters ongewens waar 'n skerp afsnyfrekwensie benodig word. Dikwels word daar na Butterworth filters verwys as maksimale gelyk filters, aangesien die oordragfunksie so gekies word dat die responskurwe so gelyk as moontlik binne die deurlaatband van die filter is [19, p. 550]. Die algemene uitdrukking vir die respons van die onderdeurlaatfilter is as volg:

$$|H| = \frac{1}{\sqrt{1 + (f/f_c)^{2n}}} \quad (3-6)$$

waar $|H|$ die grootte van die oordragfunksie is, n die orde van die filter, en f_c die afsnyfrekwensie (wat die -3dB frekwensie voorstel). Indien die afronding gespesifiseer word in dB per dekade, word die

benodigde orde van die filter verkry deur die afronding te deel met 20 [19, p. 551].

Die geskikste manier om hoër orde filters te ontwerp is om die funksie op te breek in tweede en derde-orde stadiums. Figuur 3.14a toon 'n eerste-orde onderdeurlaatfilter en Figuur 3.14b toon 'n tweede-orde onderdeurlaatfilter. Opeenvolgende stappe kan gekombineer word om die verlangde respons te lewer [22, p. 177]. Kaskadering is moontlik met operasionele versterkers as gevolg van die isolasie tussen opeenvolgende stadiums [19, p. 555].

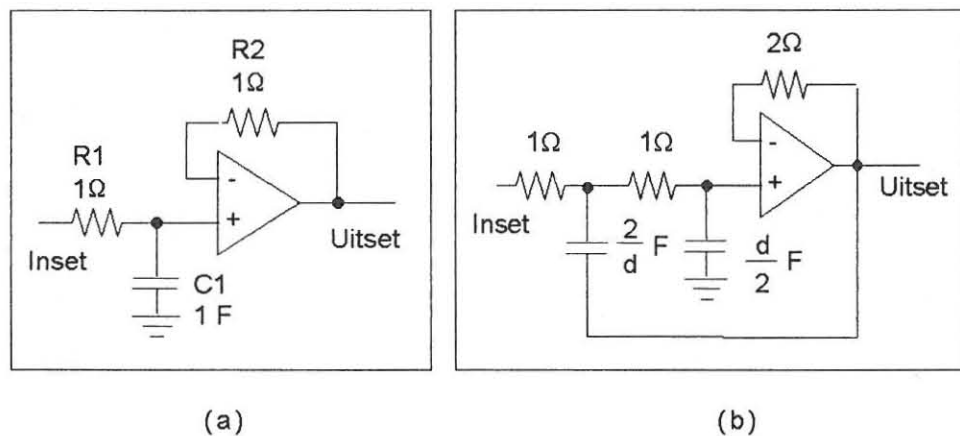


Fig. 3.14 - Onderdeurlaatfilter (a)Eerste-orde (b)Tweede-orde.

As die orde van die filter toeneem, verbeter die gelykheid van die deurlaatband van Butterworth filters. Die respons bo die afsnypunt is monotonies soos die kenkromme vir 'n Butterworth onderdeurlaatfilter in Figuur 3.15 toon. Dit wil sê dat die verswakking konstant toeneem met 'n vermeerdering in frekwensie [16, p. 16]. Daar is nie duidelike kenmerke wat die einde van die oorgangsband en die begin van die sperband definieer nie. Die

ontwerper van die filter moet die frekwensie ω_s definieer waar die verswakking A_s tot 'n sekere waarde moet afneem.

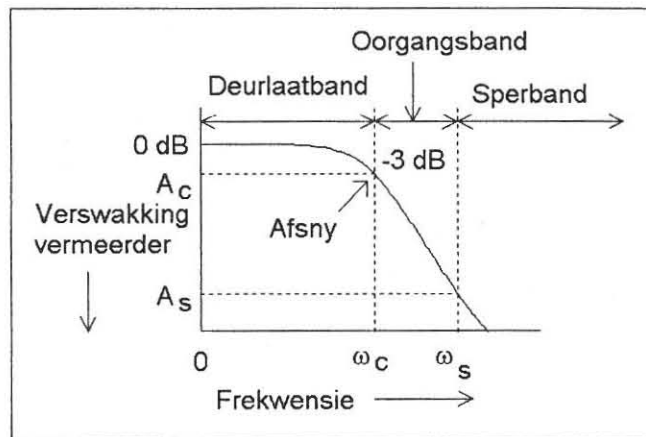


Fig. 3.15 - Butterworth onderdeurlaatfilter respons.

Filters wat die Butterworth benadering gebruik is 'n kompromie tussen die helling van verswakking en die aanvangsgelykheid in die deurlaatband. Die verswakking, A , van 'n Butterworth onderdeurlaatfilter word gegee deur

$$A = 10 \log \left[1 + \left(\frac{\omega}{\omega_c} \right)^{2n} \right] \text{ dB} \quad (3-7)$$

waar ω die hoekfrekwensie in rad/s is waarby die gespesifiseerde verswakking bereken word, ω_c is die afsnyfrekwensie in rad/s en n is die orde van die filter - wat gelyk is aan die aantal elemente in die filter. Vergelyking (3-7) kan gebruik word om die verswakking by enige frekwensie te bereken, onder of bo die afsnyfrekwensie. Vergelyking (3-8) is 'n alternatiewe metode om die benodigde orde van 'n Butterworth filter te bereken.

$$n = \frac{\log [10^{A/10} - 1]}{2 \log \left(\frac{\omega}{\omega_c} \right)} \quad (3-8)$$

Die veranderlikes het dieselfde betekenis as in vergelyking (3-7).

Die afsnyfrekwensie van eerste-orde seksies kan gelyk aan enige verlangde waarde gestel word. By die tweede-orde seksies kan die dempingsfaktor, d , en ω_0 verander word [16, p. 76]. Dempingsfaktor is die neiging tot ossillasie van 'n aktiewe filter. Praktiese dempingsfaktor waardes is in die strek vanaf 2 tot 0, met zero demping die waarde vir ossillasie. Dempingsfaktor is die omgekeerde van Q , waar Q gebruik word om die bandwydte van 'n tweede-orde banddeurlaatseksie te bepaal. Soos wat katalogusse met genormaliseerde komponentwaardes vir passiewe filters beskikbaar is, kan die waardes vir d en ω_0 ook getabelleer word vir aktiewe filters.

Tabel 3.1 is 'n katalogus vir die Butterworth benadering vir $n = 2$ tot 10 [16, p. 76]. Die eerste-orde seksie van alle onewe orde filters is konstant met $d = 2$ en $\omega_0 = 1$. Vir die tweede-orde seksie verander slegs die waarde van d aangesien ω_0 gelyk is aan 1 vir alle ordes. Hierdie tabel beskryf die Butterworth filter vir alle tipe toepassings.

Tabel 3.1 Butterworth d en ω_0 waardes.

Filter	Orde	d	ω_0
B02	02	1.4142	1.0000
B03	03	1.0000 2.0000	1.0000 1.0000
B04	04	0.7654 1.8478	1.0000 1.0000
B05	05	0.6180 1.6180 2.0000	1.0000 1.0000 1.0000
B06	06	0.5175 1.4142 1.9319	1.0000 1.0000 1.0000
B07	07	0.4450 1.2470 1.8019 2.0000	1.0000 1.0000 1.0000 1.0000
B08	08	0.3902 1.1111 1.6629 1.9616	1.0000 1.0000 1.0000 1.0000
B09	09	0.3473 1.0000 1.5321 1.8794 2.0000	1.0000 1.0000 1.0000 1.0000 1.0000
B10	10	0.3129 0.9080 1.4142 1.7820 1.9754	1.0000 1.0000 1.0000 1.0000 1.0000



3.10 OPSOMMING

Die ideale operasionele versterker lewer 'n uitset wat eweredig is aan die verskil tussen die twee seine wat toegepas word op die insetterminale, en lewer teoreties 'n uitset van zero indien twee identiese seine gelyktydig op albei terminale toegepas word.

Integreerders word effektief gebruik vir berekeninge en seinprosesseringstoepassings. Die uitset van die integreerder is



eweredig tot die integraal oor tyd van die insetsein. Die kriterium vir 'n integreerder is dat die laagste seinfrekwensie bo die -3dB punt moet wees, andersins sal dit soos 'n onderdeurlaatfilter reageer. Differensieerders is toestelle wat die wiskundige funksie van differensiasie uitvoer en voorsien 'n uitsetspanning wat eweredig is aan die helling van die insetspanning. By 'n differensieerder moet die hoogste frekwensie onder die -3dB punt wees anders werk dit soos 'n bodeurlaatfilter.

Onderdeurlaatfilters laat frekwensies onder die afsnyfrekwensie deur en blokkeer alle ander frekwensies, terwyl 'n bodeurlaatfilter die teenoorgestelde doen. Die afsnypunt word geneem waar die uitset tot onder die halfkragpunt, -3dB, daal. Die verlangde respons word verkry deur die kaskadering van eerste en tweede-orde filter stadiums. Daar is weinig belasting tussen die verskillende stadiums as gevolg van die lae uitsetimpedansie en elke seksie is sodoende geïsoleer van mekaar en kan afsonderlik gestel word.

Die frekwensieskaalfaktor kan gebruik word om 'n gegewe filter se respons te verskuif na 'n ander frekwensiestrek deur die reaktiewe elemente van die filter te deel met die verlangde faktor. Chebyshev filters het 'n skerp afsnypunt maar veroorsaak rimpels in die deurlaatstrek, terwyl Butterworth filters se respons in die deurlaatstrek gelyk is en nie so 'n skerp afsnypunt besit nie.

HOOFSTUK 4

ONTWERP VAN DIE HARDEWARE

4.1 INLEIDING TOT DIE ONTWERP VAN DIE HARDEWARE

Die ontwerp van 'n elektroniese instrument bestaan hoofsaaklik uit die samestelling van bewese stroombane en toestelle om 'n spesifieke doelwit te bereik. Die doelwit van 'n elektroniese instrument is die meting en vertoning van 'n fisiese parameter. Hierdie hoofstuk beskryf die ontwerp van die hardeware van die meetinstrument om transiënte van roterende masjiene te meet. Figuur 4.1 toon 'n blokdiagram van die meetstelsel soos ontwikkel. Die ontwerp van die eerste twee blokke word in hierdie hoofstuk bespreek.

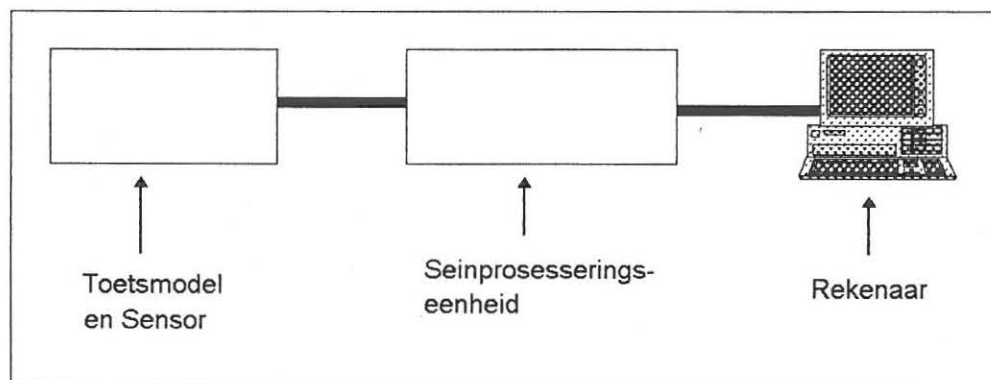


Fig. 4.1 - Blokdiagram van meetstelsel.

Die eerste blok in die blokdiagram bestaan uit die motor of toestel wat getoets gaan word, asook die sensoreenheid. Twee tipes

sensoreenhede word bespreek, naamlik die tweeskyf stelsel en die as-enkodeerder. 'n Vereiste vir die twee tipes sensors is dat hul uitsette soortgelyk moet wees sodat dieselfde seinprosesseringseenheid vir beide stelsels gebruik kan word. Die uitsetseine van die sensors word vervolgens in die seinprosesseringseenheid verwerk. Figuur 4.2 toon 'n blokdigram van die sensors en seinprosesseringseenheid.

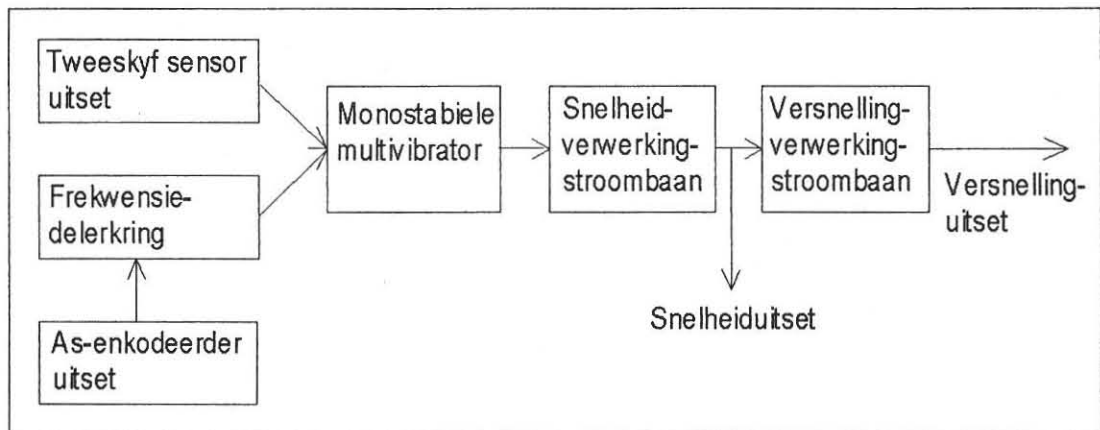


Fig. 4.2 - Blokdigram van seinprosesseringseenheid.

Die seinprosesseringseenheid bestaan uit die volgende drie verwerkingskringe:

- Monostabiele multivibrator (sien Paragraaf 4.3).
- Snelheidsverwerkingsstroombaan (sien Paragraaf 4.4).
- Versnellingsverwerkingsstroombaan (sien Paragraaf 4.6).

Die uitsetsein van die seinprosesseringseenheid word met behulp van 'n analoog-na-digitaal omsetter in 'n rekenaar ingelees. Die rekenaar en analoog-na-digitaal omsetterkaart se ontwerp gaan nie bespreek word nie aangesien dit 'n losstaande eenheid is en nie as deel van die projek herontwerp is nie.

4.2 SENSOR ONTWERP

Eerstens is die tweeskyf sensor ontwerp oorweeg. Die stelsel bestaan uit twee identiese 219mm ondeursigtige perspex skywe met 100 gate in elk. Die gate is 14mm vanaf die rand van die skyf geboor, is 2.9mm groot en 2.9mm van mekaar af. Die skywe word op twee verskillende motors gemonteer. Die twee motors kan in dieselfde rigting, of in teenoorgestelde rigtings roteer. Die verwysingsmotor is nodig sodat die minimum transmissiefrekwensie van inligting oor die posisie van die veranderlike spoed skyf verhoog kan word vanaf zero tot die frekwensie van die verwysingsmotor. Byvoorbeeld, indien die verwysingsmotor teen 3000 omwentelings per minuut roteer en daar is 100 gate in die skyf sal die standaard pulsfrekwensie 5kHz wees. Indien die twee motors in teenoorgestelde rigtings roteer vermeerder die aantal gleuf samevallings en indien die motor onder evaluering teen 1500 omwentelinge per minuut roteer, verhoog die pulsfrekwensie tot 7.5kHz.

Deur middel van filtrering word die veranderlike spoed skyf se pulsfrekwensie vanaf die frekwensie van die verwysingsskyf geskei, dit wil sê slegs die frekwensie vanaf 5kHz tot 7.5kHz word verkry. Die minimum transmissiefrekwensie van die veranderlike spoed skyf is dus verhoog vanaf 0Hz tot 5kHz.

Die snelheid van die verwysingsmotor is nie baie belangrik nie, maar indien die twee motors in dieselfde rigting roteer moet die verwysingsmotor se snelheid hoër wees as die van die veranderlike spoed skyf, aangesien die relatiewe pulsfrekwensies van mekaar

afgetrek word. Byvoorbeeld, indien die veranderlike spoed motor teen 1500 omwentelinge per minuut roteer sal die pulsfrekwensie verlaag vanaf 5kHz tot 2.5kHz.

4.2.1 TWEESKYF, EEN LIGBRON STELSEL

Die samestelling in Figuur 4.3 is aanvanklik gebruik. Aan die een kant van die skywe is 'n ligbron geplaas en aan die anderkant 'n fototransistor. Aanvanklik is 'n LED as ligbron gebruik. 'n Belangrike vereiste vir die stelsel is dat die twee skywe so na as moontlik aan mekaar geplaas moet word en presies teenoor mekaar opgestel moet word. Die ligstraal vanaf die ligbron na die fototransistor word meganies deur die beweging van die twee skywe onderbreek.

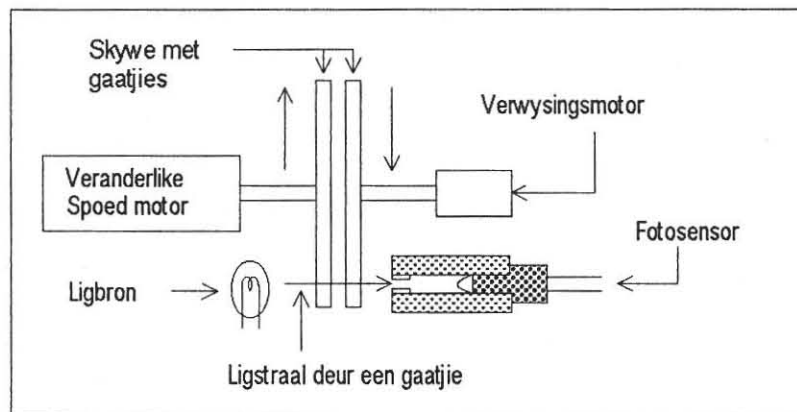


Fig. 4.3 - Sensorsamestelling sonder lense.

Die uitsetsein van die fototransistor Q_2 in Figuur 4.4 is nie baie groot nie (ongeveer 0.67V). 'n Gemeenskaplike emitter versterkerkring word gebruik om die uitsetsein tot ongeveer 5V te

versterk. Die versterkerkring is nodig aangesien Q_2 in Figuur 4.4 se uitsetspanning te laag is vir die snellering van die monostabiele multivibrator (sien Paragraaf 4.3).

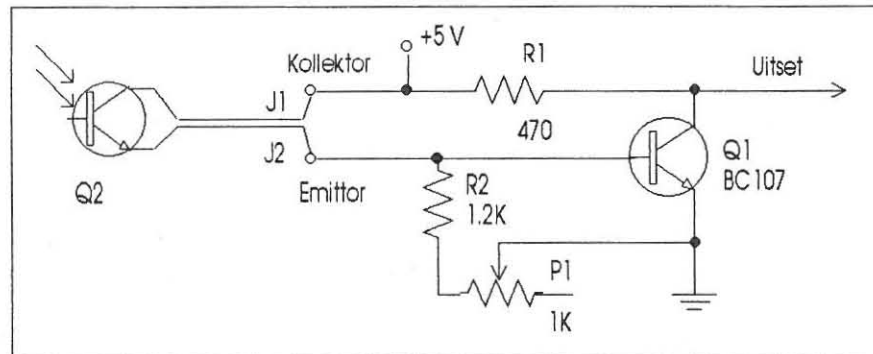


Fig. 4.4 - Sein versterker.

Die uitsetsein wat by transistor Q_1 se kollektor verkry word is 'n reghoekpuls waarvan die frekwensie direk eweredig is aan die snelheid van die twee skywe. Namate die snelheid van die veranderlike spoed skyf verander sal die frekwensie van die uitsetsein ook verander.

Die ligemissiediode in die ligbron van die sensoresamestelling is met 'n gloeilamp wat 'n sterker lig uitstraal vervang, sodat die ligveranderings wat by die fototransistor plaasvind duideliker bespeur kon word. Die gloeilamp is ook eksperimenteel vervang met 'n laser, maar daar was nie 'n merkwaardige verskil tussen die fototransistor uitset as gevolg van die gloeilamp en die laser nie. Die grootste nadeel van die samestelling is dat die ligstraal slegs deur een gaatjie skyn. Indien beide skywe stilstaan met twee gaatjies oor mekaar sal die ligstraal op die fototransistor val.

'n Probleem ontstaan wanneer die twee skywe begin roteer. Indien die een skyf 'n entjie kloksgewys beweeg sal die ander skyf 'n entjie antikloksgewys beweeg, wat veroorsaak dat daar nie meer twee gaatjies is wat regoor mekaar staan nie. Sodoende word geen lig deurgelaat na die fototransistor nie en die ligstraal is onderbreek. Dit het ook gebeur dat twee gaatjies ooreenkom maar nie op die oorspronklike posisie nie en dus het die ligstraal die fokuspunt van die fototransistor gemis. Die ligstraal val slegs op die fototransistor indien twee gaatjies ooreenkom presies op die punt waar die stelsel in die rustoestand opgestel is. Dus, in hierdie geval gaan daar van die pulse verlore. Deur gebruik te maak van kollimeerlense is die probleem van pulse wat verlore gaan oorkom.

4.2.2 TWEESKYF LIGBRON EN LENSSTELSEL

As gevolg van pulse wat verlore gaan in die sensoreenheid wat in Paragraaf 4.2.1 bespreek is, is ongewenste ruis in die sensoreenheid opgewek. Die opwekking van ruis in die sensoreenheid kan in 'n groot mate voorkom word deur van lense gebruik te maak. Die lense soos gebruik was 35mm in deursnee en is 25mm van mekaar af geplaas. Die lense se brandpunt was 30mm vanaf die lens se middelpunt.

Soos getoon in Figuur 4.5 is 'n gloeilamp op die brandpunt van 'n lens geplaas en 'n parallelle ligstraal is opgewek wat deur meer as een gaatjie van die skywe skyn. Aan die anderkant van die twee skywe gaan die ligstrale wat deur die gaatjies skyn weer

deur 'n lens wat al die strale kombineer en fokus. 'n Fototransistor, in 'n houer gemonteer om agtergrond beligting te blokkeer, is op die brandpunt van die tweede lens geplaas. Al is die gaatjies dus nie op dieselfde plek oor mekaar binne die ligbundel van die kollimeerstelsel nie, word die ligstraal steeds gefokus op die fototransistor en gaan geen pulse verlore nie. Sodoende word die opwekking van ruis deur die sensoreenheid beperk.

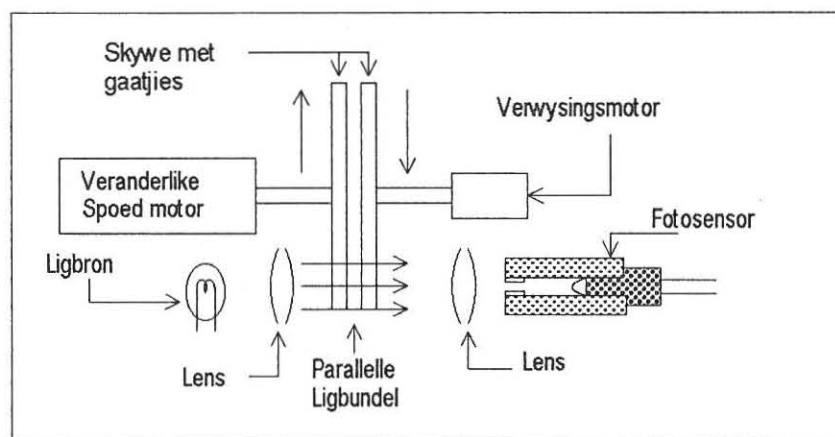


Fig. 4.5 - Sensorsamestelling met lense.

Dieselfde versterkerkring soos in Figuur 4.4 getoon, is saam met die sensor gebruik om die uitsetsein van Q_2 te versterk. Hierdie sensoreenheid samestelling om pulse vanaf die motors te verkry is akkuraat, maar die opstelling van die toetsapparaat is 'n tydrowende proses aangesien die twee skywe presies teenoor mekaar opgestel moet wees. Baie tyd word aan die opstelling van die skywe spandeer om die opwekking van ruis as gevolg van eksentriese montering te voorkom. Daarna moet verseker word dat die gloeilamp en die fototransistor op die brandpunte van die lense is voordat daar begin kan word met die neem van enige lesings.

4.2.3 AS-ENKODEERDER

'n Inkrementele as-enkodeerder met drie uitsette is ook in die navorsing gebruik. Die eerste twee uitsette, A en B, lewer elk 1000 pulse per omwenteling wat 90° uit fase is, terwyl die derde uitset 'n verwysingspuls met elke omwenteling lewer [20, p. 712]. Die enkodeerder benodig 'n toevoerspanning van tussen +12V en +24V. Die uitsetsein van die as-enkodeerder is 'n TTL aanpasbare reghoekgolf waarvan die frekwensie direk eweredig is aan die rotasiesnelheid van die as-enkodeerder. Die uitsetsein kan direk gebruik word om die monostabiele multivibrator aan te dryf (sien Paragraaf 4.3).

Opstelling van die toetsmodel wat die enkodeerder gebruik is baie eenvoudig en vinnig. Slegs die as van die as-enkodeerder en dié van die motor moet aan mekaar verbind word. Heelwat tyd word bespaar - relatief tot die prosedure soos bespreek in punt 4.2.2 - om die sensor te monteer en dit kan maklik van een motor na 'n ander geskuif word.

Die uitsetsein van die as-enkodeerder word gekoppel aan die stroombaan soos in Figuur 4.6 getoon. Die hoofdoel van die stroombaan is om die frekwensie van die as-enkodeerder uitsetsein te verminder. Die vermindering van frekwensie van die as-enkodeerdersein is nodig ten einde dieselfde verwerkingskring as vir die tweeskyf sensor te kan gebruik. Die uitsetspannings van die seinprosesseringseenheid mag maksimaal tussen -10V en +10V wissel. Indien die insetfrekwensie te hoog is word hierdie

waardes baie vinnig bereik en daarom is begrensing van die uitsetfrequentie van die as-encodeerder nodig.

Die insetfrequentie gemeet by die monostabiele multivibrator van die sensor, waar afkapping intree sonder die delerkring, is 8kHz, wat voldoende is vir die tweeskyf sensor. Hierdie frequentie is onvoldoende vir die as-encodeerder en daarom word die uitsetfrequentie daarvan verminder deur dit met vyf te deel. Nadat die frekwensiedeler in die kring gekoppel is, was die hoogste uitsetfrequentie van die as-encodeerder gemeet 26kHz, voor die frekwensiedeler. Die gemete uitsetspanning van die snelheidverwerking-stroombaan was 5.35V en geen afkapping het voorgekom nie.

Die werking van die frekwensiedeler in Figuur 4.6 is as volg:

1. Die uitset van die as-encodeerder gaan deur die zenerdiode afkapperkring, D_1 en D_2 , om te verseker dat die amplitude van die puls nie groter as die TTL standaard is nie. Indien die sein nie begrens word nie kan die geïntegreerde kringe beskadig word.
2. Na die zenerdiodes gaan die uitsetsein deur die delerkring waar die frequentie van die uitsetsein deur vyf gedeel word. Die uitset van die delerkring kan nou aan die monostabiele multivibrator kring verbind word (sien Paragraaf 4.3).

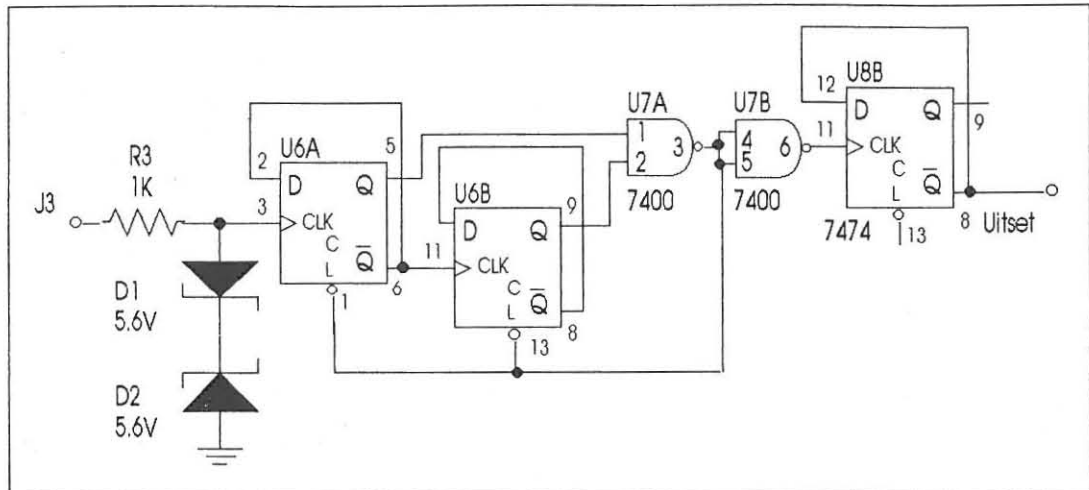


Fig. 4.6 - Frekwensiedeler.

4.3 MONOSTABIELE MULTIVIBRATOR

Die uitset van die frekwensieverdeler sneller 'n monostabiele multivibrator. 'n Monostabiele multivibrator is stabiel in een staat maar onstabiel in die ander. Die kring moet gesneller word om 'n verandering by die Q uitset te veroorsaak. In reaksie op die snellerpuls gaan die Q uitset hoog vanaf die stabiele staat na die onstabiele staat en die \bar{Q} uitset laag. Die uitsette bly in hierdie toestand vir 'n vooraf bepaalde tydperk voordat dit terugkeer na hul oorspronklike stand.

Figuur 4.7 toon die monostabiele kring wat in die navorsing gebruik is. Die kapasitor C_1 , tesame met die weerstand R_4 , bepaal die pulswydte. Die tydskuur wat die sein in die onstabiele staat sal verkeer, word as volg bepaal:

$$\begin{aligned}
 T &= 0.7RC \\
 T &= 0.7 \times 22\text{k}\Omega \times 1\text{nF} \\
 T &= 15.4\mu\text{s}
 \end{aligned}
 \tag{4-1}$$

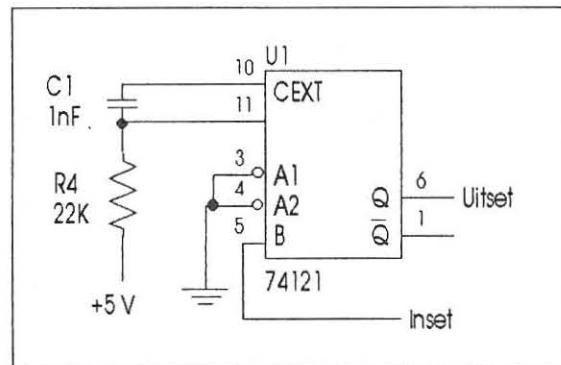


Fig. 4.7 - Monostabiele multivibrator.

Die monostabiele multivibrator lewer een reghoekige puls vir elke klokpuls. Die kring kan nie weer gesneller word terwyl dit in die onstabiele staat is nie, aangesien die insetsein vanaf die sensoreenheid eers moet daal tot onder 'n drumpelspanning om af te skakel en dan weer moet styg tot 'n drumpelspanning om snellering te veroorsaak. Deurdat A₁ en A₂ gegrond is word snellering veroorsaak indien B verander vanaf grond na 'n positiewe waarde. Die monostabiele kring lewer reghoekpulsse waarvan die amplitude ongeveer 3.5V is oor die hele frekwensiestrek van werking. Die uitset is dus 'n reeks pulse met 'n konstante duurre maar verskillende frekwensies, afhangend van die rotasiesnelheid van die toetsmotor. Dus, inderwaarheid 'n variërende dienssiklus uitset, wat varieer met snelheid.

4.4 SNELHEIDVERWERKINGSTROOMBAAN

Die snelheidverwerkingstroombaan is die tweede gedeelte van die seinprosesseringseenheid (sien Figuur 4.2). In hierdie gedeelte van die stroombaan word die veranderende frekwensie omgeskakel in 'n veranderende spanning wat die rotasiesnelheid van die motor voorstel. Die omskakeling word bewerkstellig deur die pulse van die veranderende frekwensie sein te integreer. Sodoende word 'n variërende uitsetspanning eweredig aan 'n variërende insetfrekwensie verkry. 'n Belangrike vereiste by die ontwerp van die integreerder is dat die laagste frekwensie wat geïntegreer moet word, hoër as die afsnyfrekwensie van die onderdeurlaatfilter moet wees. Indien hierdie vereiste nie nagekom word nie, sal die uitsetspanning verkry nie die korrekte rotasiesnelheid van die motor voorstel nie.

Frekwensierespons is een van die belangrikste spesifikasies by die keuse van filters. Frekwensierespons van 'n filter kan uitgedruk word as 'n verhouding van twee polinome in s waar $s = j\omega$ ($j = \sqrt{-1}$ en ω die frekwensie in radiale per sekonde, $2\pi f$) en word na verwys as die oordragfunksie [25, p. 1-1]. Wiskundig kan dit as volg uitgedruk word :

$$T(s) = \frac{E_L}{E_s} = \frac{N(s)}{D(s)} \quad (4-2)$$

Die wortels van die noemer polinoom $D(s)$ word pole genoem en die wortels van die teller polinoom $N(s)$ word zeros genoem. Die absis α is die reële komponent van die wortels en die ordinaat β is die

imaginêre gedeelte. Aktiewe filters kan direk vanaf die pole en zeros ontwerp word deur gebruik te maak van Tabel 4.1.

Tabel 4.1 - Butterworth poolplasing [25, p. 11-1].

Orde n	Reële gedeelte - α	Imaginêre gedeelte $\pm j\beta$
2	0.7071	0.7071
3	0.5000 1.0000	0.8660
4	0.9239 0.3827	0.3827 0.9239
5	0.8090 0.3090 1.0000	0.5878 0.9511

Vir navorsingsdoeleindes is daar besluit op 'n aktiewe Butterworth onderdeurlaatfilter aangesien die filter se uitset maksimaal gelyk is tot by zero frekwensie.

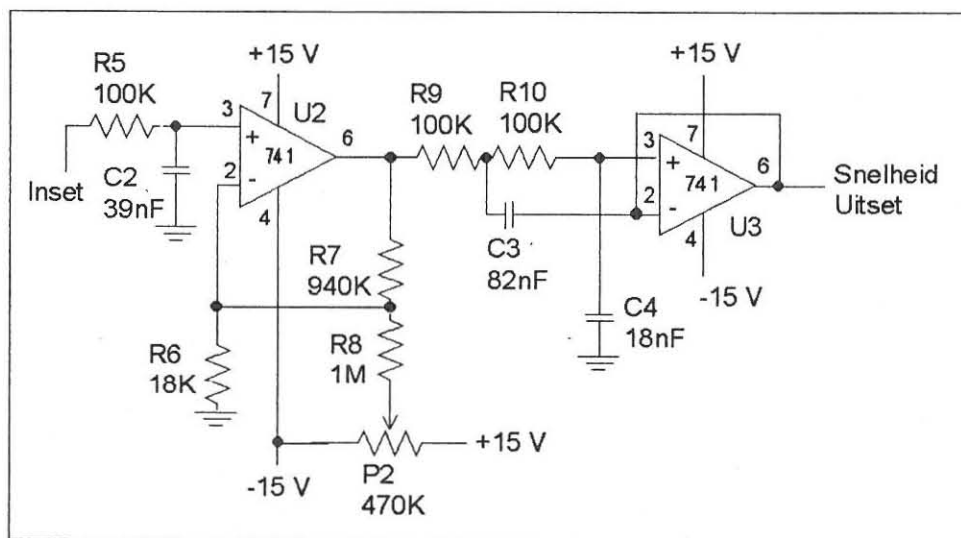


Fig. 4.8 - Butterworth derde-orde onderdeurlaatfilter.

Die uitsetfrequentie van die monostabiele multivibrator gaan deur 'n derde-orde Butterworth onderdeurlaatfilter met 'n -3dB afsnypunt by 40Hz. Hierdie frekwensie waarde is gekies om moontlike 50Hz rimpels van die toevoerspanning wat in die stelsel kan voorkom te blokkeer. Die filter kan in twee dele verdeel word, naamlik 'n eerste-orde filter en 'n tweede-orde filter, soos getoon in Figuur 4.8.

Die inset na die onderdeurlaatfilter word op die nie-omkeer terminaal van U_2 in Figuur 4.8 verbind. 'n Versterkingswinst van 52 word deur die filter gelewer. Die wins is tot 52 beperk om te voorkom dat die filter tot in sy versadigingsgebied gedryf word met die uitsetsein vanaf die tweeskyf sensor stelsel. Om hierdie rede is dit ook noodsaaklik dat die frekwensie van die uitsetsein vanaf die as-enkodeerder eers verminder moet word voordat dit aan die snelheidverwerkingstroombaan gekoppel word. Die $n = 2$ seksie besit eenheidswinst en die totale wins word dus deur die $n = 1$ seksie gelewer. Die waarde van R_6 is willekeurig gekies as 18k en R_7 is soos volg bereken:

$$A = \frac{R_7}{R_6}$$
$$52 \times 18k = R_7 \quad (4-3)$$
$$940k = R_7$$

Die poolplasings van 'n genormaliseerde $n = 3$ Butterworth onderdeurlaatfilter word vanaf Tabel 4.1 verkry.

Komplekse pool : $\alpha = 0.5000$ en $\beta = 0.8660$

Reële pool : $\alpha_0 = 1.0000$

Die genormaliseerde kapasitorwaardes vir die $n = 2$ seksie is soos volg [25, p. 3-20]:

$$C_3 = \frac{1}{\alpha} = \frac{1}{0.5} = 2F \quad (4-4)$$

$$C_4 = \frac{\alpha}{\alpha^2 + \beta^2} = \frac{0.5}{0.5^2 + 0.866^2} = 0.5F \quad (4-5)$$

en vir die $n = 1$ seksie is die genormaliseerde kapasitorwaarde soos volg:

$$C_2 = \frac{1}{\alpha_0} = \frac{1}{1.0} = 1F \quad (4-6)$$

Deur die reaktiewe elemente te deel met die frekwensieskaalfaktor, kan 'n filter se respons na 'n ander frekwensiestrek geskuif word.

$$\begin{aligned} \text{FSF} &= \frac{\text{Verlangde verwysingsfrekwensie}}{\text{Bestaande verwysingsfrekwensie}} \\ \text{FSF} &= \frac{2\pi 40 \text{ rad/s}}{1 \text{ rad/s}} \\ \text{FSF} &= 251,33 \end{aligned} \quad (4-7)$$

Die impedansieskaleringsfaktor is 'n waarde wat gekies kan word en in die geval is dit gekies as 10^5 . Die waarde van die impedansieskaleringsfaktor is so gekies dat praktiese kapasitorwaardes verkry kan word. Deur van die FSF waarde en die impedansieskaleringsfaktor gebruik te maak kan die gedenormaliseerde kapasitorwaardes bereken word [25, p. 2-4]. Die kapasitorwaardes vir die $n = 2$ seksie is soos volg:

$$C_3' = \frac{C_3}{FSF \times Z} = \frac{2}{251.33 \times 10^5} = 79\text{nF} \quad (4-8)$$

$$C_4' = \frac{C_4}{FSF \times Z} = \frac{0.5}{251.33 \times 10^5} = 19.8\text{nF}$$

en vir die $n = 1$ seksie

$$C_2' = \frac{C_2}{FSF \times Z} = \frac{1}{251.33 \times 10^5} = 39\text{nF}$$

Die berekende kapasitorwaardes is afgerond na die naaste beskikbare kommersiële kapasitorwaardes, en is soos volg:

$$C_3' = 82\text{nF}$$

$$C_4' = 18\text{nF}$$

$$C_2' = 39\text{nF}$$

Die weerstandwaardes in die genormaliseerde filterkring is gelyk aan 1Ω . Die weerstande word vermenigvuldig met die impedansieskaleringsfaktor, Z , wat veroorsaak dat die weerstande dieselfde weerstand van $Z\Omega$ sal besit. Aangesien $Z = 10^5$ sal die weerstandwaardes in die genormaliseerde filterkring gelyk wees aan $100\text{k}\Omega$, d.w.s. R_5 , R_9 en $R_{10} = 1\Omega \times 10^5 = 100\text{k}\Omega$. Die waarde van Z kan onafhanklik vir elke filterseksie gekies word en hoef nie dieselfde te wees nie, aangesien die individuele seksies van mekaar geïsoleer is deur middel van operasionele versterkers. Die FSF waarde moet egter vir elke seksie dieselfde wees.

Deur die kapasitorwaarde (C_2') en die weerstandwaarde (R_5) in vergelyking (3-4) te vervang, kan die afsnypunt van die onderdeurlaatfilter ontwerp bereken word.

$$f_c = \frac{1}{2\pi R_5 C_2'} \text{ Hz} \quad (4-9)$$

$$f_c = \frac{1}{2\pi 100\text{K} \times 39\text{nF}}$$

$$f_c = 40.81 \text{ Hz}$$

Vergelyking (4-9) bevestig dat die afsnypunt van die onderdeurlaatfilter ver onder 50Hz is en moontlike rimpels in die toevoerspanning blokkeer sal word. Die onderdeurlaatfilter soos ontwerp voldoen dus aan die vereiste kriteria van die ontwerp.

Deur die eerste-orde filterseksie met die versterkingswins eerste te plaas word die sein eers versterk, wat sodoende die seinruisverhouding verbeter en dit is dus moontlik om rimpels wat in die sein teenwoordig kan wees, makliker uit te filter. Die uitsetsein wat verkry word verteenwoordig die snelheidkurwe van die motor.

Die minimum rotasiesnelheid wat deur die snelheidverwerkingstroombaan gemeet kan word, met die as-enkodeerder sensoreenheid, is soos volg:

$$\begin{aligned} \text{As - enkodeerder uitsetfrequentie} &= \text{Afsnyfrequentie} \times \text{FSF} & (4-10) \\ &= 40\text{Hz} \times 5 \\ &= 200\text{Hz} \end{aligned}$$

Die as-encodeerder lewer 1000 pulse/omwenteling.

'n 200Hz sein word gelewer deur 200 pulse/sekonde.

$$\begin{aligned} \text{Dit verteenwoordig} \quad & \frac{200 \text{ pulse/sekonde}}{1000 \text{ pulse/omwenteling}} & (4-11) \\ & = 0,2 \text{ omwentelinge/sekonde van die encodeerder} \end{aligned}$$

$$\begin{aligned} 0,2 \text{ omwentelinge/sekonde} &= 0,2 \times 60 \\ &= 12\text{rpm} \end{aligned}$$

'n Minimum uitsetfrekwensie van 200Hz verteenwoordig dus 'n minimum meetbare rotasiesnelheid van 12rpm.

4.5 FREKWENSIE-NA-SPANNING GEÏNTEGREERDE STROOMBAAN

'n LM2917 frekwensie-na-spanning omsetter geïntegreerde stroombaan is getoets in die navorsing aangesien dit dieselfde funksie uitvoer as die snelheidverwerkingstroombaan soos beskryf in Paragraaf 4.4. Dit lewer dus 'n uitsetspanning wat varieer in ooreenstemming met die insetfrekwensie. Die stroombaan in Figuur 4.9 is vir eksperimentele doeleindes gebruik. Eksperimente wat met die stroombaan gedoen is, het getoon dat die frekwensie-na-spanning omsetter lineêr is vanaf 0Hz tot ongeveer 8kHz.

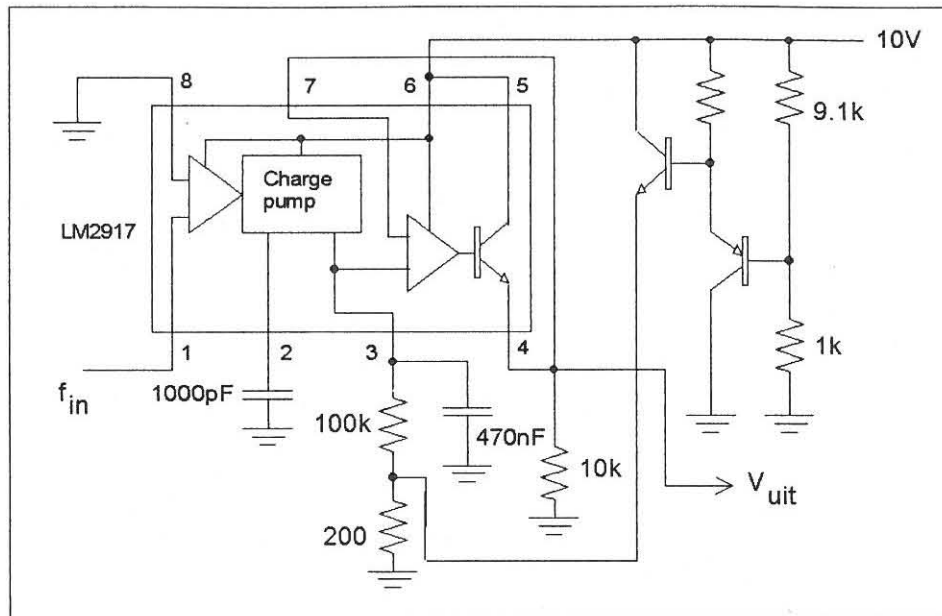


Fig. 4.9 - LM2917 frekwensie-na-spanning omsetter [15, p. 28].

Figuur 4.10 toon die lineariteit wat gemeet is met die stroombaan tot by 8kHz, en hoe dit begin afplat na 8kHz.

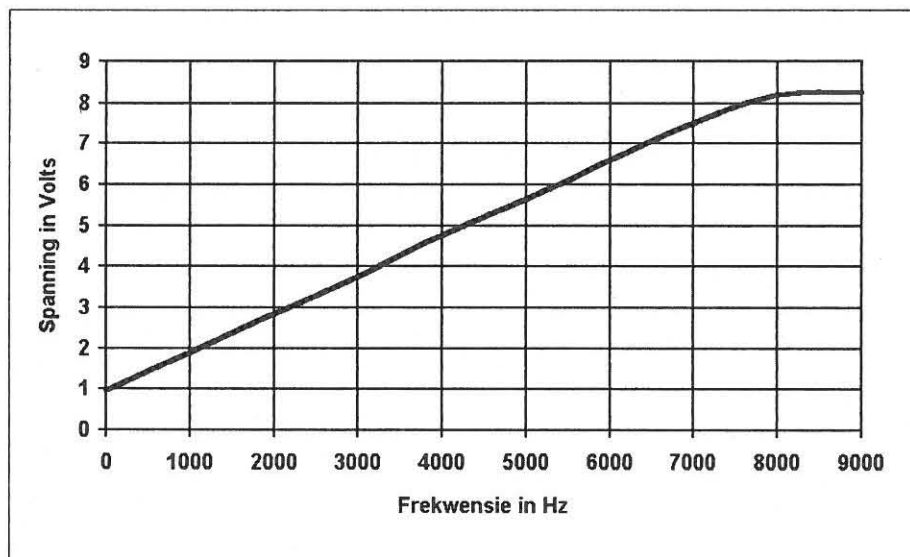


Fig. 4.10 - Gemete lineariteit van LM2917 frekwensie-na-spanning omsetter.

'n Probleem wat egter met die stelsel ondervind is, is die feit dat die uitsetsein nie intyd saam met die insetsein verander nie. Die

reaksietyd van die omsetter was te lank en sodoende het baie van die informasie wat verlang word verlore gegaan. Dus was die frekwensie-na-spanning omsetter nie geskik vir die navorsingsdoeleindes nie en is daar teen die gebruik daarvan besluit.

4.6 VERSNELLINGVERWERKINGSTROOMBAAN

Om die versnelling van die roterende motor te verkry moet die snelheidsein gedifferensieer word. Die spanning wat hieruit verkry word stel die versnelling van die motor voor. Die basiese kring word in Figuur 4.11 getoon.

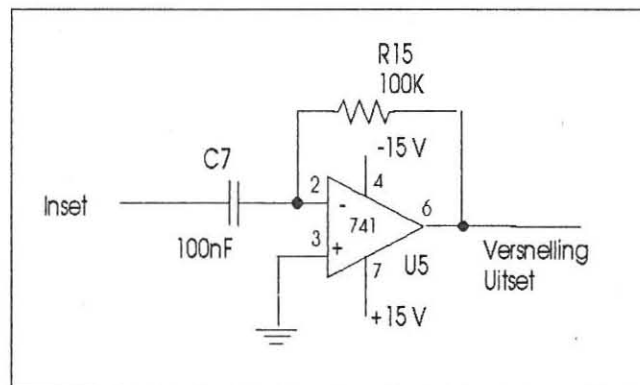


Fig. 4.11 - Basiese differensieerder.

Die uitsetspanning vir hierdie kring word gegee deur -

$$V_{uit} = -R_{15}C_7 \frac{dV_{in}}{dt} \quad (4-12)$$

waar die hoeveelheid $\frac{dV_{in}}{dt}$ die verandering in insetspanning is wat oor 'n spesifieke tydverloop plaasvind. Die uitset van die

differensieerder is gelyk aan die spanningsval oor die terugkoppelweerstand R_{15} - en is eweredig aan die negatiewe afgeleide van die insetsein.

Die nadeel van die differensieerderkring is dat die wins van die operasionele versterker deur die verhouding $\frac{R_{15}}{X_{C7}}$ bepaal word. Die kring is ook gevoelig vir hoë frekwensie ruis, en om dit te bowe te kom, is die kring in Figuur 4.11 drasties aangepas. Figuur 4.12 toon die finale differensieerder wat ontwerp en gebruik is in die navorsing.

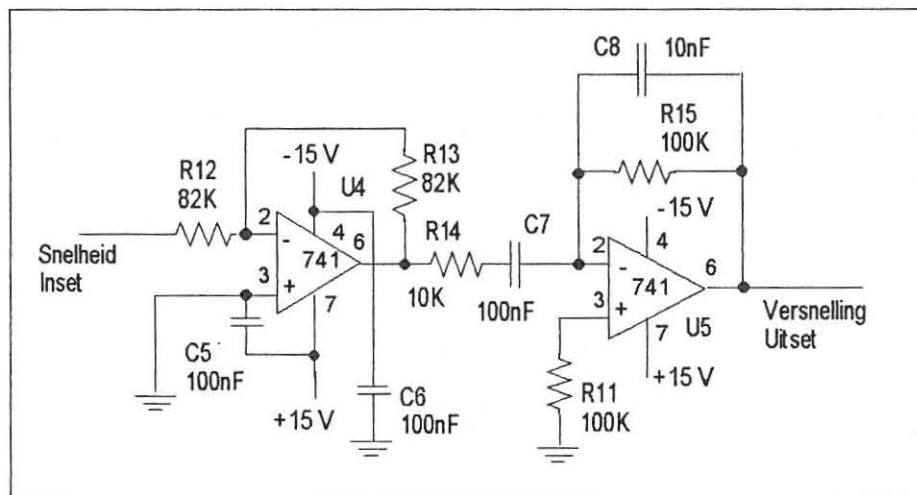


Fig. 4.12 - Differensieerder.

'n Weerstand, R_{14} , is in serie met die insetkapasitor C_7 geplaas. Op hierdie wyse word die hoë frekwensie wins tot die verhouding $\frac{R_{15}}{R_{14}}$ beperk. Die waarde van weerstand R_{14} word ongeveer tien keer kleiner as die van weerstand R_{15} gemaak. 'n Verdere weerstand, R_{11} , met dieselfde waarde as weerstand R_{15} , word in die kring ingesluit om die insetverskilspanning te balanseer. Om die hoë

frekwensie wins verder te beperk word 'n klein kapasitor, C_8 , in parallel met weerstand R_{15} verbind.

Om te bevestig dat die differensieerder as 'n differensieerder sal funksioneer word die afsnyfrekwensie van die kring bereken as:

$$f_c = \frac{1}{2\pi R_{14} C_7} \quad (4-13)$$

$$f_c = \frac{1}{2\pi \times 10 \times 10^3 \times 100 \times 10^{-9}}$$

$$f_c = 159\text{Hz}$$

Die insetsein na die differensieerder kom vanaf die snelheidverwerkingstroombaan, wat basies 'n onderdeurlaatfilter met 'n afsnyfrekwensie van 40Hz is. Die insetsein is dus ver onder die berekende 159Hz en dus sal die kring as 'n differensieerder funksioneer. Die uitsetsein van die differensieerder is 180° uit fase met die insetsein en stel die versnellingsein van die motor onder toets voor.

Aangesien die differensieerder se uitset 180° uit fase is met die inset, gaan die insetsein, wat vanaf die snelheidverwerkingstroombaan kom, eers deur 'n omkeer eenheidswins versterker (U_4 in Figuur 4.12). Die rede vir die stap is om die uitset van die versnellingverwerkingstroombaan in fase te bring met die uitset van die snelheidverwerkingstroombaan.

Eenheidswins versterkers het 'n hoë insetimpedansie en 'n lae uitsetimpedansie, wat gebruik word om daaropvolgende stadiums

aan te dryf. Nadat die uitsetsein van die eenheidswins versterker deur die differensieerder is, is die uitset verkry weer in fase met die uitset van die snelheidverwerkingstroombaan. Sodoende word 'n snelheid en 'n versnellingsein, wat in fase met mekaar is, van die motor onder toets verkry en kan variasies in versnelling met variasies in snelheid duidelik waargeneem word.

Figuur A.1 in Bylae A toon die volledige kringdiagram van die seinprosesseringseenheid wat uit die delerkring (Paragraaf 4.2.3), monostabiele multivibrator (Paragraaf 4.3), snelheidverwerkingskring (Paragraaf 4.4) en versnellingverwerkingstroombaan (Paragraaf 4.6), bestaan. Die twee uitsette verkry vanaf die seinprosesseringseenheid word met behulp van die PC-30 analogoog-na-digitaal omsetterkaart in 'n rekenaar ingelees waar die resultaat op die rekenaarskerm vertoon word, en gestoor kan word vir latere verwysing of verwerking van data.

Figuur A.2 in Bylae A toon die stroombaan van die kragbron wat in die seinprosesseringseenheid gebruik word. Die kragbron het drie uitsette, nl. +15V, -15V en +5V. Die $\pm 15V$ word vir die seinprosesseringseenheid en die as-enkodeerder gebruik en die +5V word vir die seinversterker van die tweeskyf sensorstelsel gebruik.

4.7 OPSOMMING

Wanneer die tweeskyf metode as die sensoreenheid vir die toetsinstrument gebruik word, word 'n kollimeerstelsel gebruik om

te verseker dat die ligstraal wat gebruik word maksimaal deur die gaatjies in die skywe gaan. Sodoende sal die lig oor meer as een gat skyn en op een punt aan die anderkant van die skywe gefokus word en geen pulse sal verlore gaan nie. Die metode is egter 'n tydrowende proses as gevolg van al die voorsorgmaatreëls wat getref moet word met die opstelling van die apparaat ter verkryging van suksesvolle metings.

Die as-enkodeerder metode is 'n baie eenvoudiger proses aangesien dit maklik aan die toetsmotor gekoppel kan word en 'n uitsetpuls lewer wat geen versterking nodig het nie.

Die uitsetpulse van die as-enkodeerder word geïntegreer om 'n snelheidsein te lewer. Die frekwensie van die pulse is direk eweredig aan die rotasiesnelheid van die motor onder toets. Die snelheidsein word gedifferensieer wat die versnellingsein lewer. Beide die seine kan met behulp van 'n analoog-na-digitaal omsetterkaart in 'n rekenaar ingelees en vertoon word.

HOOFSTUK 5

SAGTEWARE

5.1 INLEIDING TOT DIE ONTWERP VAN DIE SAGTEWARE

Al die sagteware is geskryf deur gebruik te maak van Turbo Pascal 7.0 en Turbovision¹. Die sagteware wat tydens die projek ontwikkel is bestaan hoofsaaklik uit vyf programme nl.:

- Hoofprogram
- Monsterfrekwensieprogram
- Monsterdataprogram
- Monsterkanaalprogram
- Monsterprogram

Elk van hierdie programme word vervolgens bespreek.

5.2 HOOFPROGRAM

Die hoofprogram voorsien 'n spyskaart vanwaar verskillende opsies geselekteer kan word, byvoorbeeld:

- * Om die kanaal (of kanale) wat bemonster moet word te selekteer.

¹ Turbo Pascal is 'n handelsmerk van Borland International.

- * Om te spesifiseer hoeveel monsters geneem moet word.
- * Om die monsterfrekwensie te spesifiseer.
- * Om die monsterring te aktiveer en die resultaat in 'n grafiese formaat op die rekenarskerm te vertoon.

'n Uitdruk van die program kan in Bylae C gesien word terwyl Figuur 5.1 die basiese uitleg van die hoofprogram toon.

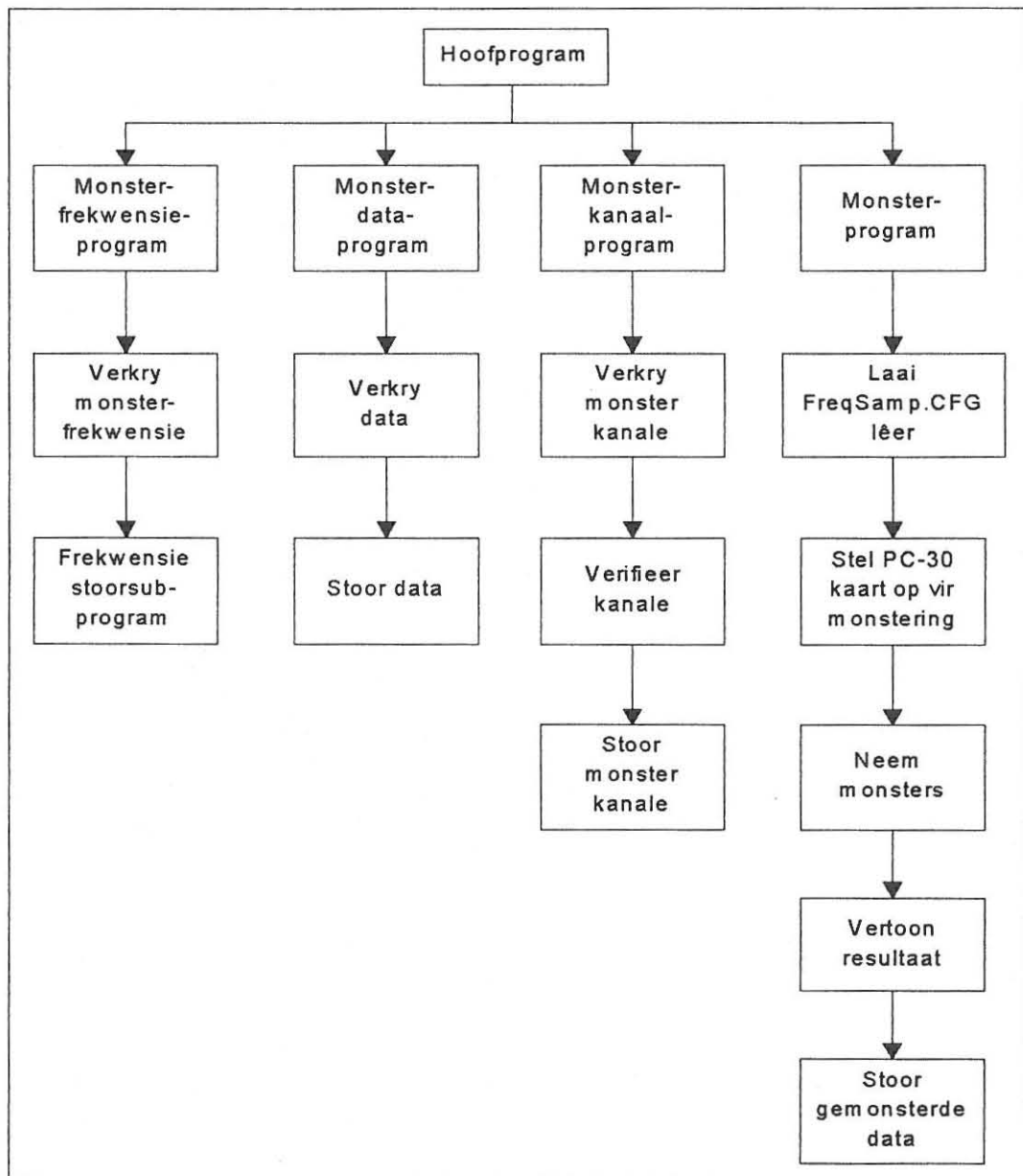


Fig. 5.1 - Struktuurkaart van hoofprogram.

Die verskillende eenhede wat deur die program gebruik word, word in die begin van die hoofprogram gedefinieer. Die verskillende keuses word uitgevoer deur gebruik te maak van die muis, wysersleutels of die kortpadsleutels. Die volgende subprogramme is deel van die hoofprogram:

- Inligtingsubprogram
- Helpsubprogram
- Herstelsubprogram
- Einde van program

Die volgende is 'n pseudokode lys van die hoofprogram soos ontwikkel.

1. Verkry die klokfrekwensie van die stelsel (sien Paragraaf 5.6.1).
2. Vertoon spyskaart.
 - 2.1. Inligting oor die program benodig?
 - 2.1.1. Ja - Gaan na inligtingsubprogram (sien Paragraaf 5.2.1).
 - 2.1.2. Anders - Gaan na 2.2.
 - 2.2. Hulp benodig?
 - 2.2.1. Ja - Gaan na helpsubprogram (sien Paragraaf 5.2.2).
 - 2.2.2. Anders - Gaan na 2.3.
 - 2.3. Vertoon monsterfrekwensies?
 - 2.3.1. Ja - Gaan na monsterfrekwensieprogram (sien Paragraaf 5.3).
 - 2.3.2. Anders - Gaan na 2.4.
 - 2.4. Vertoon monsterring parameters?
 - 2.4.1. Ja - Gaan na monsterdataprogram (sien Paragraaf 5.4).
 - 2.4.2. Anders - Gaan na 2.5.

2.5. Vertoon kanale om te bemonster?

2.5.1. Ja - Gaan na monsterkanaalprogram (sien Paragraaf 5.5).

2.5.2. Anders - Gaan na 2.6.

2.6. Herstel keuses?

2.6.1. Ja - Gaan na herstelsubprogram (sien Paragraaf 5.2.3).

2.6.2. Anders - Gaan na 2.7.

2.7. Begin bemonstering?

2.7.1. Ja - Gaan na monsterprogram (sien Paragraaf 5.6).

2.7.2. Anders - Gaan na 2.8.

2.8. Stoor data?

2.8.1. Ja - Gaan na stoordatasubprogram (sien Paragraaf 5.6.4).

2.8.2. Anders - Gaan na 2.9.

2.9. Stop?

2.9.1. Ja - Einde van program.

2.9.2. Anders - Gaan na 2.

5.2.1 INLIGTINGSUBPROGRAM

Die inligtingsubprogram vertoon die volgende inligting omtrent die program op die rekenaar se skerm:

- die naam van die program,
- die weergawe van die program,
- die kopiereg,
- wanneer die program geskryf is,
- die programmeerder se naam.

5.2.2 HELPSUBPROGRAM

Indien die help opsie geselekteer word, lees die subprogram 'n tekslêer vanaf die harde- of sagteskyf wat in gebruik is, en word die hulp inligting na die rekenaar se skerm geskryf. Die inligting wat vertoon word verduidelik vir die operateur die funksies van die onderskeie opsies wat geselekteer kan word. Bylae H is 'n uitdruk van die help tekslêer. 'n Kort beskrywing, of wenk, van 'n opsie word in die statuslyn vertoon wanneer die operateur van die monsterprogram die wyser oor die verskillende opsies in die spyskaart beweeg.

5.2.3 HERSTELSUBPROGRAM

Hierdie program stel al die opsies terug na hul oorspronklike staat, naamlik die neem van agt monsters teen 1Hz op kanaal 0. Dit word alleenlik gedoen nadat die operateur bevestig het dat die waardes teruggestel moet word na hul oorspronklike staat.

5.2.4 EINDE VAN PROGRAM

Die subroetine beëindig die werking van die monsterprogram en 'n Dos-promp word verkry.



5.3 MONSTERFREKWENSIEPROGRAM

Die doel van hierdie program is om die verskillende monsterfrekwensies te vertoon wat deur die operateur geselekteer kan word wanneer die program vanaf die spyskaart geaktiveer word. Slegs een frekwensie kan uit die lys van nege moontlikhede (wat wissel vanaf 1Hz tot 20kHz) gekies word. 'n Geselekteerde monsterfrekwensie sal behou word totdat die operateur die stelling verander. Bylae D is 'n uitdruk van hierdie program. Die volgende subprogramme is deel van die monsterfrekwensieprogram:

- Frekwensie stoorsubprogram
- Frekwensie laaisubprogram

Die monsterfrekwensieprogram word as volg opgestel:

1. Behou vorige monsterfrekwensie?

1.1. Ja - Gaan na frekwensie laaisubprogram (sien Paragraaf 5.3.2).

1.2. Anders - Gaan na 2.

2. Vertoon monsterfrekwensies.

3. Nuwe keuse?

3.1. Ja - Gaan na monsterfrekwensieprogram (sien Paragraaf 5.3).

3.2. Stoor keuse?

3.2.1. Ja - Gaan na frekwensie stoorsubprogram (sien Paragraaf 5.3.1).

3.2.2. Anders - Gaan na 4.

4. Stop?

4.1. Ja - Einde van program.

4.2. Anders - Gaan na 2.

5.3.1 FREKWENSIE STOORSUBPROGRAM

'n Lêer met die naam FreqSamp.CFG word oopgemaak waarna die monsterinligting na die lêer geskryf word. Die lêer kan op die harde- of sagteskyf van die rekenaar wees. Die data wat in die lêer gestoor word is slegs dit wat benodig word om die monstering te spesifiseer ten opsigte van frekwensie, ens. en is nie die fisies bemonsterde data nie.

5.3.2 FREKWENSIE LAAISUBPROGRAM

Die doel van hierdie program is om die FreqSamp.CFG lêer vanaf 'n harde- of sagteskyf in die geheue van die rekenaar in te lees en vanaf die lêer te bepaal wat die vorige monsterfrekwensie was. Hierdie frekwensie sal behou word totdat die operateur dit verander. Indien die lêer nie gevind word nie, sal die monsterfrekwensie na die oorspronklike waarde gestel word (1Hz).

5.4 MONSTERDATAPROGRAM

Die program vertoon die moontlike aantal monsters wat van elke kanaal geneem kan word. 'n Waarde tussen 'n minimum van 8 en 'n

maksimum van 4096 monsters kan gekies word. Wanneer 'n keuse uitgevoer is, sal die waarde behou word vir toekomstige gebruik totdat dit deur die operateur verander word. Die verkose aantal monsters word gestoor deur die frekwensie stoorsubprogram wat in Paragraaf 5.3.1 bespreek is. Bylae E is 'n uitdruk van die monsterdataprogram.

5.4.1 MONSTER LAAISUBPROGRAM

Die monster laaisubprogram is 'n subprogram van die monsterdataprogram. Vanaf die FreqSamp.CFG lêer word bepaal wat die laaste opsie was van hoeveel monsters geneem moes word. Die waarde sal behou word vir toekomstige gebruik totdat die operateur dit verander. Indien die lêer nie gevind word nie, sal die monster hoeveelheid na die oorspronklike waarde (8 monsters) gestel word.

5.5 MONSTERKANAALPROGRAM

Die program vertoon die sestien moontlike kanale wat bemonster kan word. Enige hoeveelheid kanale (tussen 1 en 16) kan gekies word. Nadat die keuse van kanale wat bemonster moet word uitgevoer is, sal die keuse behou word vir toekomstige gebruik totdat dit deur die operateur verander word. Die spesifieke kanale wat bemonster moet word, word gestoor deur die frekwensie stoorprogram wat in Paragraaf 5.3.1 bespreek is. Bylae F is 'n

uitdruk van die program. Die volgende subprogramme is deel van die monsterring kanaalprogram:

- Verifikasiesubprogram
- Monsterringkanaal laaisubprogram

5.5.1 VERIFIKASIESUBPROGRAM

Die verifikasieprogram is ontwikkel om te bepaal hoeveel kanale bemonster moet word en om die aantal monsters per kanaal te verifieer.

Tabel 5.1 - Data monsters teen bemonsterringkanale.

Aantal kanale	Maksimum aantal monsters per kanaal
1 - 2	4096
3 - 4	2048
5 - 8	1024
9 - 16	512

Tabel 5.1 toon die maksimum aantal monsters per kanaal wat geneem kan word teen die aantal kanale wat bemonster moet word. Hierdie beperkings geld aangesien die totale aantal data monsters nie 8192 monsters kan oorskry nie. Die rede hiervoor is dat die PC-30 analoog-na-digitale omsetterkaart wat gebruik word 'n tipiese omskakelingstyd van 25ms het. Daarom, indien

8192 monsters omgeskakel word teen 25ms elk word 'n frekwensie van 4.88MHz verkry, wat die frekwensie is waarteen die PC-30 kaart aangedryf word. Dus, indien meer as 8192 monsters geneem word sal daar 'n vertraging in die stelsel voorkom en sal die monsters nie teen 'n konstante frekwensie geneem word nie. Daar bestaan ook 'n moontlikheid dat van die monsters verlore kan gaan. Vandaar die beperking op die maksimum aantal monsters.

5.5.2 MONSTERKANAAL LAASUBPROGRAM

Die program bepaal vanaf die FreqSamp.CFG lêer wat die vorige keuse was van watter kanale bemonster moes word. Die kanale sal behou word vir toekomstige gebruik totdat die operateur dit verander. Indien die lêer nie gevind word nie, sal die monsterkanaal na die oorspronklike kanaal gestel word (kanaal 0).

5.6 MONSTERPROGRAM

Die monsterprogram maak gebruik van die vooraf opgestelde data in die FreqSamp.CFG lêer om die monsters te neem en kan geaktiveer word deur middel van die F9 sleutel. Bylae G is 'n uitdruk van die monsterprogram en die program bestaan uit die volgende subprogramme:

- Verkry stelselkloksubprogram

- Onderbrekingsbeheerde bemonsteringssubprogram
- Vertoon datasubprogram
- Stoor datasubprogram

5.6.1 VERKRY STELSELKLOKSUBPROGRAM

Die subprogram word uitgevoer om die klokfrekwensie van die stelsel te bepaal. Die frekwensie word benodig om die programmeerbare tydreëlaar (8253), wat deur die analoog-na-digitale omsetterkaart gebruik word om 'n onderbrekinginsetsein te lewer, te programmeer. Indien 'n ongeldige frekwensie ingesleutel word, sal die subprogram herhaal word totdat 'n geldige frekwensie (tussen 4MHz en 25.9MHz) ingesleutel word. Die beperking op die frekwensie (4MHz tot 25.9MHz) is as gevolg van die PC-30 kaart wat gebruik is en nie as gevolg van die eienskappe van die rekenaar nie.

Nadat 'n geldige frekwensie ingesleutel is, kan die volgende toets uitgevoer word om die korrektheid van hierdie waarde te bepaal. Die toets is nie 100% akkuraat nie maar dit kan as 'n verwysing gebruik word. Aktiveer die monsterprogram om 8 monsters te neem teen 1Hz van kanaal 0. Meet die tydsduur om te bepaal of dit die stelsel agt sekondes geneem het om die agt monsters te neem.

5.6.2 ONDEBREKINGSBEHEERDE BEMONSTERINGSUBPROGRAM

Twee moontlikhede bestaan om die omskakelingsproses van analoog-na-digitaal te inisieer:

1. **Sagteware klok bemonstering**, waar die sagteware van die bemonsteringsubprogram gebruik word om 'n klok wat die omskakelingsproses inisieer te voorsien. Dit wil sê die monsterfrekwensie van die snelheid en die versnellingsein is afhanklik van die uitvoeringspoed van die subprogram, wat 'n funksie is van die klokfrekwensie van die rekenaar. Met sagteware klok bemonstering is dit moeilik om die presiese monsterfrekwensie te bepaal. 'n Beter tegniek om die monsterfrekwensie te definieer is om die tydreëlaar en die onderbrekingsein te gebruik.
2. **Programmeerbare tydreëlaar**. Deur die gebruik van die programmeerbare tydreëlaar (8253) om die analoog-na-digitale omskakelingsproses (van die AD574 omsetter) te begin, word die probleem om die presiese bemonsteringsfrekwensie te definieer oorkom. Om die programmeerbare tydreëlaar te gebruik moet die PC-30 kaart korrek opgestel wees.

Die inligting in Bylae B (Tabel B.1 en Tabel B.2) word gebruik om die verskillende poorte van die PC-30 kaart op te stel. Sodoende word die onderbrekinginsetsein geaktiveer met tydreëlaar bemonstering. Wanneer 'n analoog-na-digitale omskakeling voltooi is word 'n onderbrekingsein uitgestuur na die rekenaar en word die data in die rekenaargeheue ingelees. Wanneer met die

bemonsteringproses begin word, staak die rekenaar om ander funksies uit te voer totdat die gespesifiseerde hoeveelheid monsters geneem en omgeskakel is.

5.6.3 VERTOON DATASUBPROGRAM

Die vertoon datasubprogram bepaal watter tipe rekenarskerm gebruik word, byvoorbeeld 'n monochroomskerm of 'n kleurskerm. Die inligting word benodig om die korrekte skermdrywers in die geheue van die rekenaar in te lees. Nadat die korrekte drywers verkry is vir die grafiese skerm, word die data van die verskillende kanale wat bemonster is in 'n grafiese formaat op die rekenaar se skerm weergegee. Die grafieke van die verskillende kanale word op 'n spanning-as en 'n tyd-as voorgestel. Die spanning-as is in volts gekalibreer vanaf -10V tot +10V en stel die rotasiesnelheid van motors voor. Die tyd-as is in sekondes en dui die tydsduur aan wat dit die stelsel geneem het om die verlangde aantal monsters te neem.

Indien die operateur ontevrede is met die resultate, kan die bemonstering weer uitgevoer word sonder om terug te gaan na die hoofprogram. Die program sal onmiddellik die bemonstering uitvoer en die nuwe resultate vertoon. Indien die operateur tevrede is met die resultate, kan daar teruggekeer word na die hoofprogram en die stoor opsie vanaf die spyskaart geselekteer word om die bemonsterde data te stoor.

5.6.4 STOOR DATASUBPROGRAM

Voor bemonstering is die stoor funksie onaktief. Die funksie word aktief nadat bemonstering plaasgevind het. Deur middel van die stoor datasubprogram word die bemonsterde data in 'n teksformaat op 'n harde- of sagteskyf gestoor nadat die operateur die program van 'n lêernaam voorsien het. Die stoorprogram toets die lêernaam wat voorsien is om seker te maak dat daar nie alreeds so 'n lêer op die betrokke skyf bestaan nie, voordat die data gestoor word. Indien so 'n lêer alreeds bestaan moet die operateur die naam verander of bevestig dat daar oor die bestaande lêer geskryf kan word. Die teksformaat maak dit moontlik vir ander rekenaarprogramme, byvoorbeeld Quattro Pro of Excel, om die data vanaf die skyf te lees vir verwerking.

5.7 OPSOMMING

Daar is in hierdie navorsingsprojek so ver moontlik gepoog om die sagteware so gebruikersvriendelik en eenvoudig moontlik te hou. Die interaksie van die verskeie roetines, die subprogramme en die hoofprogram dui op 'n goed gedefinieerde stelsel.

Die sagteware ontwikkel kan slegs saam met die PC-30B en PC-30C omsetterkaart gebruik word, aangesien die twee kaarte se insette kan wissel tussen -10V en +10V, terwyl die PC-30DS se inset tussen -5V en +5V kan wissel. Die PC-30B en PC-30C kaarte maak gebruik van die programmeerbare tydreëlaar om die

monsterfrekwensie te stel terwyl die PC-30DS van 'n ander
bemonsteringstegniek gebruik maak.

HOOFSTUK 6

EVALUERING VAN DIE MEETINSTRUMENT

6.1 INLEIDING TOT DIE EVALUERING VAN DIE MEETINSTRUMENT

Die hardeware soos deur die navorser ontwikkel is aanvanklik met behulp van die STAT-30 dataverkrygingsprogram geëvalueer aangesien die navorser se eie sagteware nog nie voltooi was nie.

Tydens die ontwikkeling van die sagteware is daar van programtoetsing en -ontfouting gebruik gemaak om die korrekte werking van die program te verseker. Na voltooiing van die sagteware is dit tesame met die hardeware getoets.

6.2 LINEARITEIT VAN DIE SNELHEIDVERWERKINGSTROOMBAAN

Die meetinstrument is ontwerp om enige van twee verskillende insetseine, naamlik die insetsein vanaf die tweeskyf sensor of die insetsein vanaf die as-enkodeerder, te gebruik. Figuur 6.1 toon die gemete lineariteit van die snelheidverwerkingstroombaan ten opsigte van die inset vanaf die tweeskyf sensorstelsel. Die lineariteit is bepaal deur middel van 'n seingenerator wat aan die inset van die snelheidverwerkingstroombaan gekoppel is om die inset vanaf die tweeskyf sensor teen 'n aantal konstante frekwensies voor te stel.

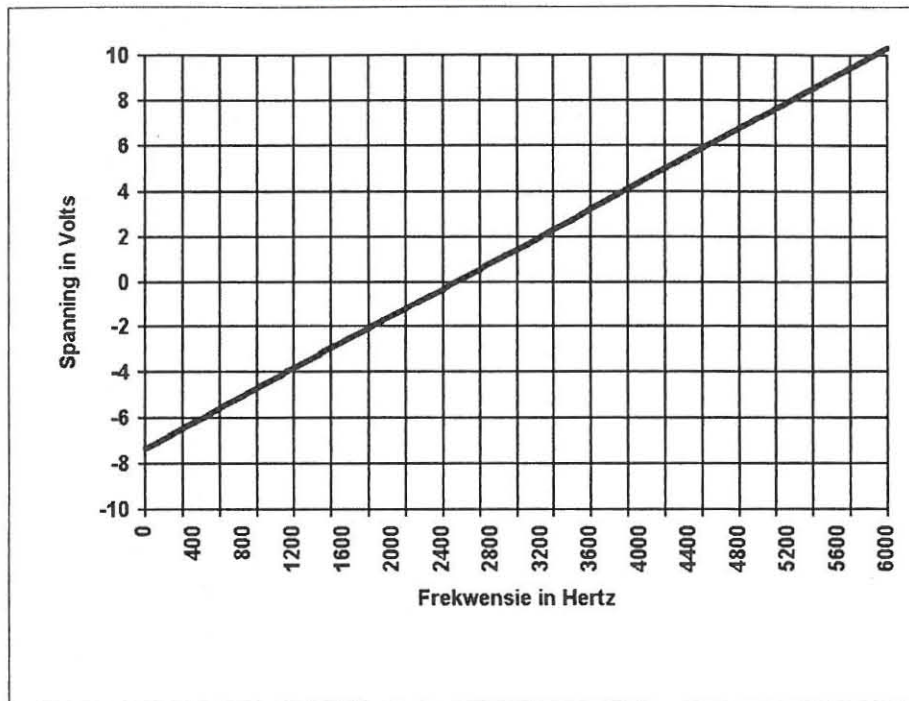


Fig. 6.1 - Lineariteit van die snelheidverwerkingstroombaan sonder frekwensiedeler.

Figuur 6.2 toon die gemete lineariteit van die snelheidverwerkingskring ten opsigte van die insetsein vanaf die as-encodeerder inset.

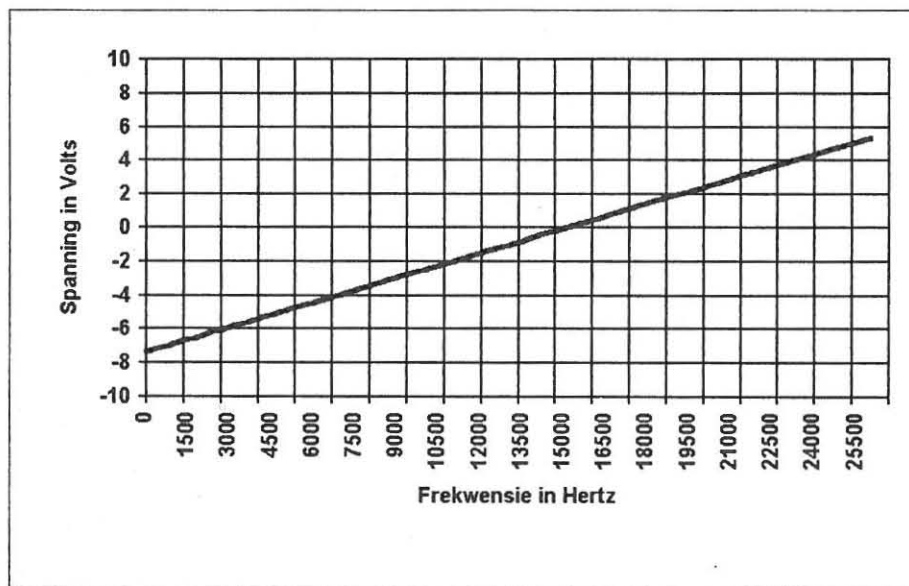


Fig. 6.2 - Lineariteit van die snelheidverwerkingstroombaan met frekwensiedeler.

Deur Figuur 6.1 met Figuur 6.2 te vergelyk is dit duidelik dat waar die gemete spannings ooreenkom die frekwensie in Figuur 6.2 omtrent vyf maal hoër is as die van Figuur 6.1. Uit die twee figure kan die afleiding gemaak word dat die snelheidverwerkingsstroombaan baie goeie lineariteit het vir beide tipes insette.

6.3 EVALUERING VAN SENSORS

Tydens die navorsing is verskeie sensorstelsels geëvalueer ten opsigte van die kompleksiteit van die sensoropstelling, asook die koppelingmetode van die sensor aan die toetsmotor. Die koppeling van die sensor aan die toetsmotor is belangrik aangesien die werking van die toetsmotor nie beïnvloed moet word deur die sensor nie. Die volgende sensors is ondersoek:

6.3.1 TWEESKYF LIGBRON EN LENSSTELSEL

Die tweeskyf ligbron en lensstelsel bestaan uit twee lense wat saam met 'n ligbron en fototransistor gebruik word. Die parallelle ligstraal skyn deur meer as een gaatjie van die skywe. Aan die anderkant van die twee skywe gaan die ligstrale wat deur die gaatjies skyn weereens deur 'n lens wat al die strale kombineer en fokus op die fototransistor. Dus, al is die gaatjies nie op dieselfde plek oor mekaar binne die ligbundel van die kollimeerstelsel nie, word die ligstraal steeds gefokus op die fototransistor. Geen pulse gaan verlore wanneer die twee skywe

draai nie en die ruis opgewek deur die sensoreenheid word verminder.

Figuur 6.3 toon die snelheiduitset wat verkry is met die sensoreenheid van 'n motor wat versnel is vanaf rus tot volspoed terwyl die verwysingsmotor teen 'n snelheid van 1500 revolusies per minuut roteer.

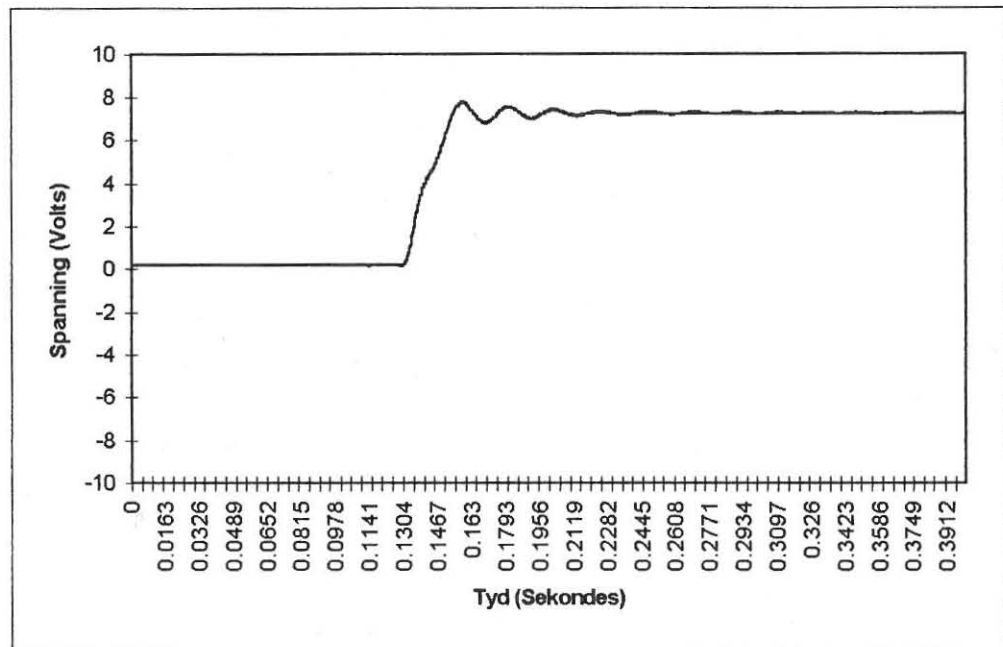


Fig. 6.3 - Snelheiduitset van tweeskyf ligbron en lensstelsel.

Vanaf Figuur 6.3 kan dit waargeneem word dat minimale ruisprodukte teenwoordig is in die snelheiduitsetsein. Dieselfde resultate is verkry terwyl die verwysingsmotor teen 'n 1000 of teen 'n 1800 revolusies per minuut roteer het, met die verskil dat die beginpunt van die snelheidsgrafiek skuif.

Figuur 6.4 toon die versnellinguitset van die sensor as gevolg van die motor wat versnel is vanaf rus tot volspoed terwyl die verwysingsmotor teen 1500 revolusies per minuut roteer. Die

ontbreking van ruiskomponente word ook in die versnellingskurwe waargeneem.

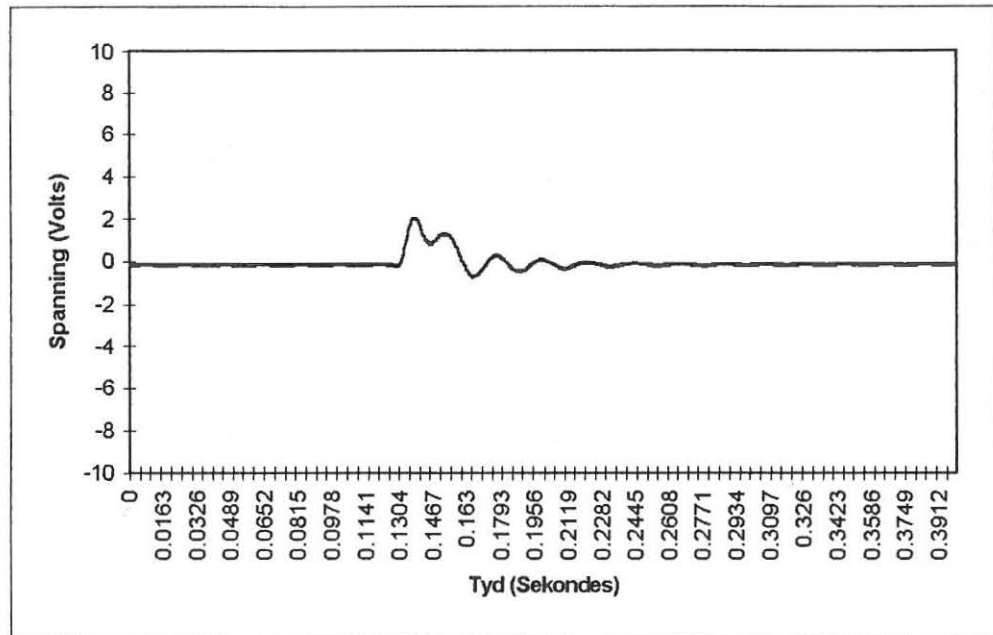


Fig. 6.4 - Versnellinguitset van tweeskyf ligbron en lensstelsel.

Uit die resultate blyk dat die tweeskyf ligbron en lensstelsel bevredigende resultate lewer. Die opstelling van hierdie stelsel is egter kompleks en tydrowend en daarom het die navorser 'n alternatiewe sensoreenheid oorweeg.

6.3.2 INKREMENTELE AS-ENKODEERDER

Die inkrementele as-encodeerder wat deur die navorser gebruik is lewer 1000 pulse per omwenteling. Figuur 6.5 toon die snelheiduitset wat verkry is met die as-encodeerder sensor van 'n motor wat versnel is vanaf rus tot volspoed.

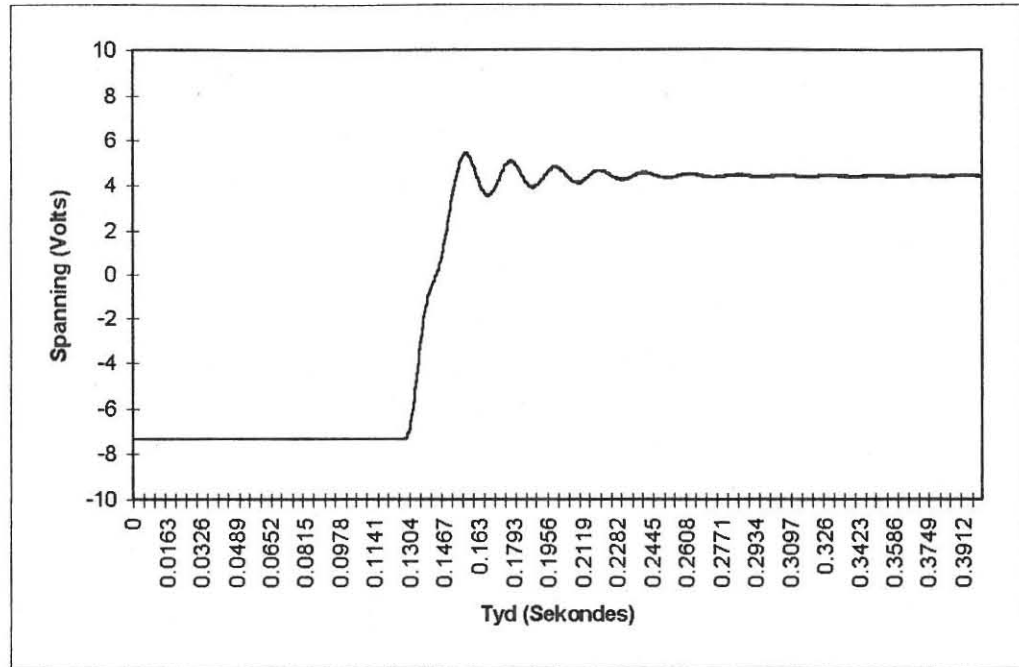


Fig. 6.5 - Snelheiduitset met die as-encodeerder.

Figuur 6.6 toon die versnellinguitset van die motor wat versnel is vanaf rus tot volspoed gemeet met die as-encodeerder.

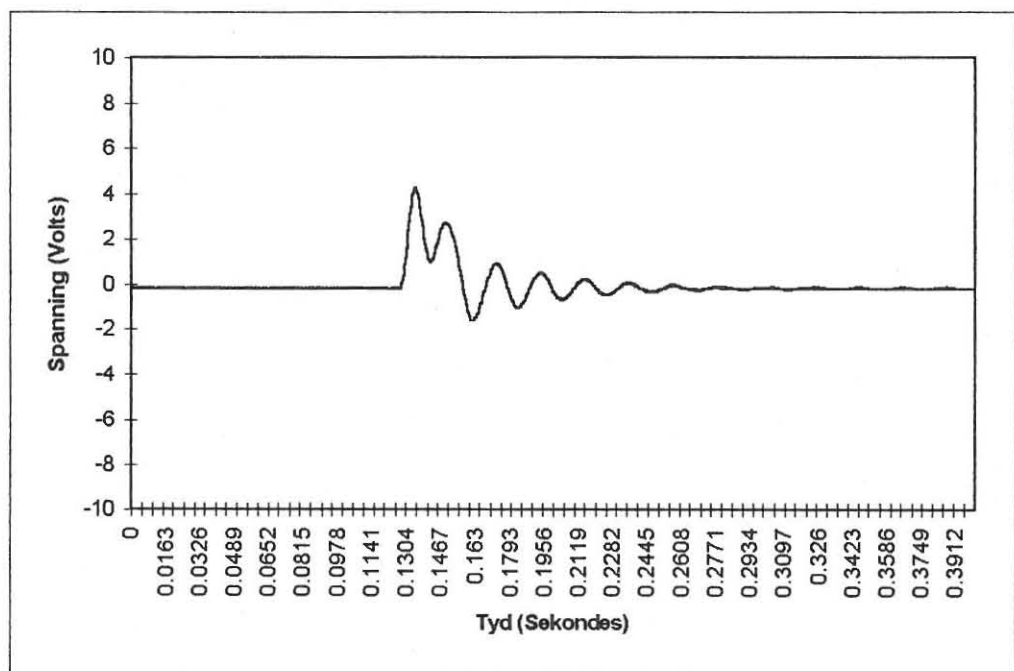


Fig. 6.6 - Versnellinguitset met die as-encodeerder.

Dieselfde motor en toetskondisies is gebruik om die as-encodeerder te evalueer as wat gebruik is om die tweeskyf ligbron en lensstelsel te evalueer. Indien die as-encodeerder uitset (Figure 6.5 en 6.6) met die tweeskyf ligbron en lensstelsel uitset (Figure 6.3 en 6.4) vergelyk word, kan dit waargeneem word dat die resultate omtrent identies is. Hieruit kan afgelei word dat die as-encodeerder dieselfde funksie uitvoer as die tweeskyf ligbron en lensstelsel. 'n Voordeel van die as-encodeerder bo die tweeskyf ligbron en lensstelsel is dat die koppeling aan die toetsmotor baie eenvoudig en vinnig is.

Verdere toetse is met die as-encodeerder uitgevoer, byvoorbeeld om die motor teen volspoed in een rigting te laat hardloop, dit te stop en dan in die teenoorgestelde rigting tot volspoed te versnel.

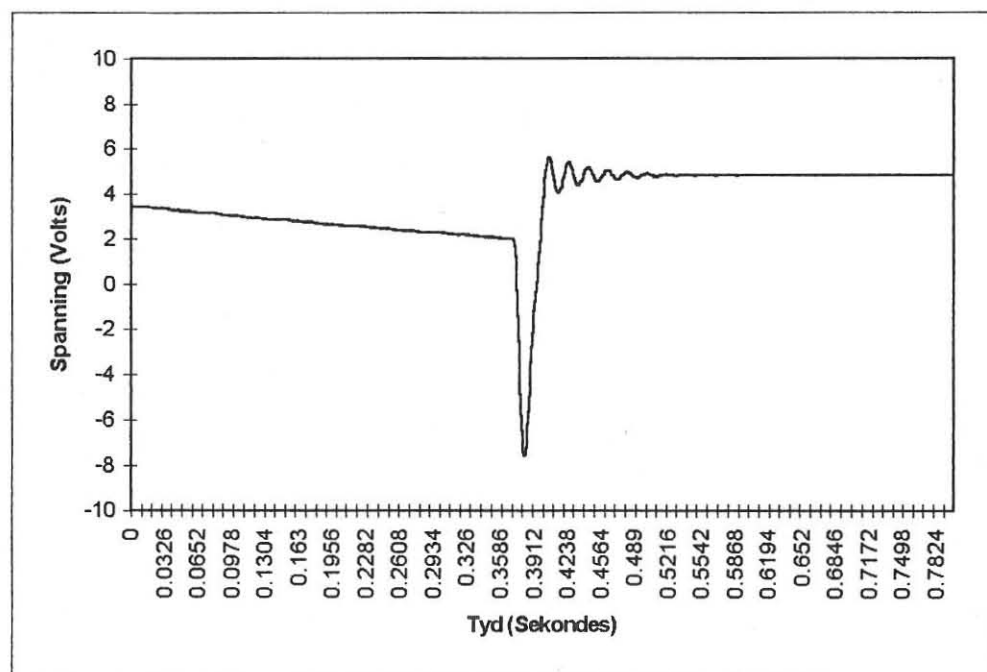


Fig. 6.7 - Snelheiduitset tydens stop en versnelling.

Figuur 6.7 toon die snelheid en Figuur 6.8 die versnelling van die motor wat eers gestop word en dan versnel word in die teenoorgestelde rigting. Dit is duidelik waarneembaar dat namate die snelheid van die motor verander, daar 'n oombliklike verandering in die versnelling is.

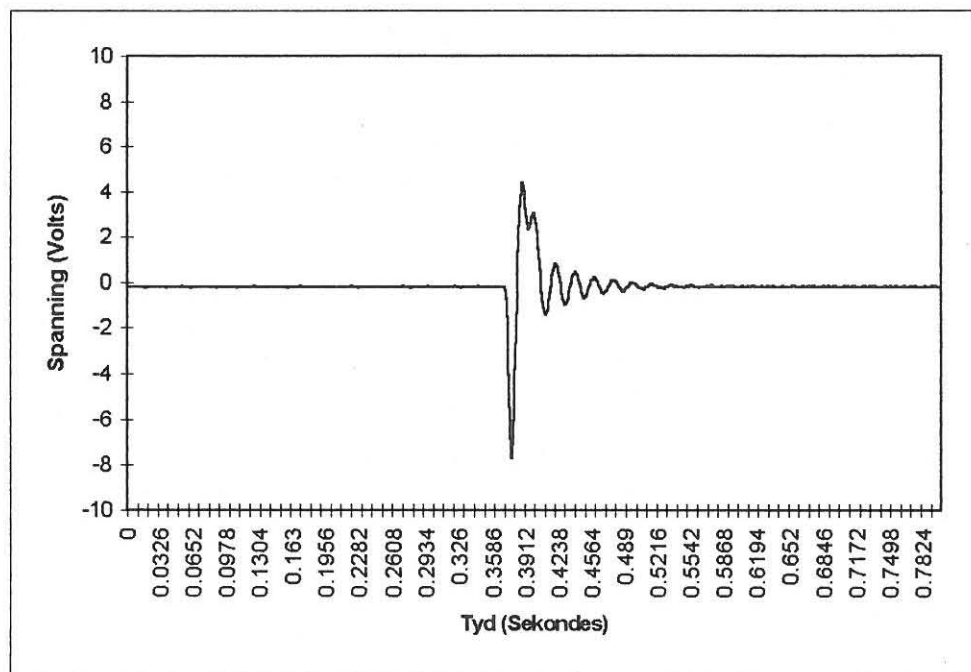


Fig. 6.8 - Versnellinguitset tydens stop en versnelling.

Die hardeware wat ontwikkel is maak nie voorsiening vir verandering in rotasierigting nie, en daarom kan die verandering in rigting nie in Figuur 6.7 waargeneem word nie. Die tweede helfte van die figuur moet onderstebo gedraai word om die verandering in rigting voor te stel.

6.4 EVALUERING VAN SNELHEIDVERWERKINGSTROOMBAAAN

Deur die snelheiduitsette van die verskillende sensors te vergelyk kan gesien word dat al die sensors wat geëvalueer is dieselfde

uitsetkenmerke openbaar. Vanaf die snelheiduitsette verkry kan gesien word dat die tydrespon in twee verdeel kan word, naamlik die oorgangsrespon en die bestendige staat respon (sien Paragraaf 2.2). Die tipiese kenmerke wat gebruik word om transiënte te karakteriseer is ook waarneembaar op die snelheiduitsette naamlik, verbyskiet, vertragingstyd, stygtyd en die vestigingstyd. Om te verseker dat die maksimum verbyskiet wat verkry is as gevolg van die motor versnelling is, en nie 'n produk van die ontwerp van die seinprosesseringseenheid is nie, is die volgende toets uitgevoer:

'n Toetsmotor is versnel vanaf rus tot volspoed en die snelheiduitset verkry is ontleed. Die volgende bevindings is gemaak vanaf die snelheiduitsetsein.

- 1) Die motor is aangeskakel en het begin versnel 32,2ms nadat die eerste monster geneem is.
- 2) Die maksimum verbyskiet is bereik 64,2ms na die aanvang van monsterring.
- 3) Die eindfrekwensie van die motor was 4,06kHz.

Die motor het dus 32ms na aanskakeling maksimum verbyskiet bereik. Die toets is herhaal met die verskil dat die insetsein vanaf 'n seingenerator verkry is en nie vanaf 'n motor nie. Die seingenerator is gestel om 'n konstante 4kHz sein te lewer.

Die toets is begin met 0Hz op die inset van die seinprosesseringseenheid. Nadat die toets begin is, is die 0Hz sein vervang met die 4kHz sein van die seingenerator. Die tydrespon verkry kan soos

die snelheiduitsetsein in twee verdeel word, naamlik die oorgangsrespons en die bestendige staat respons.

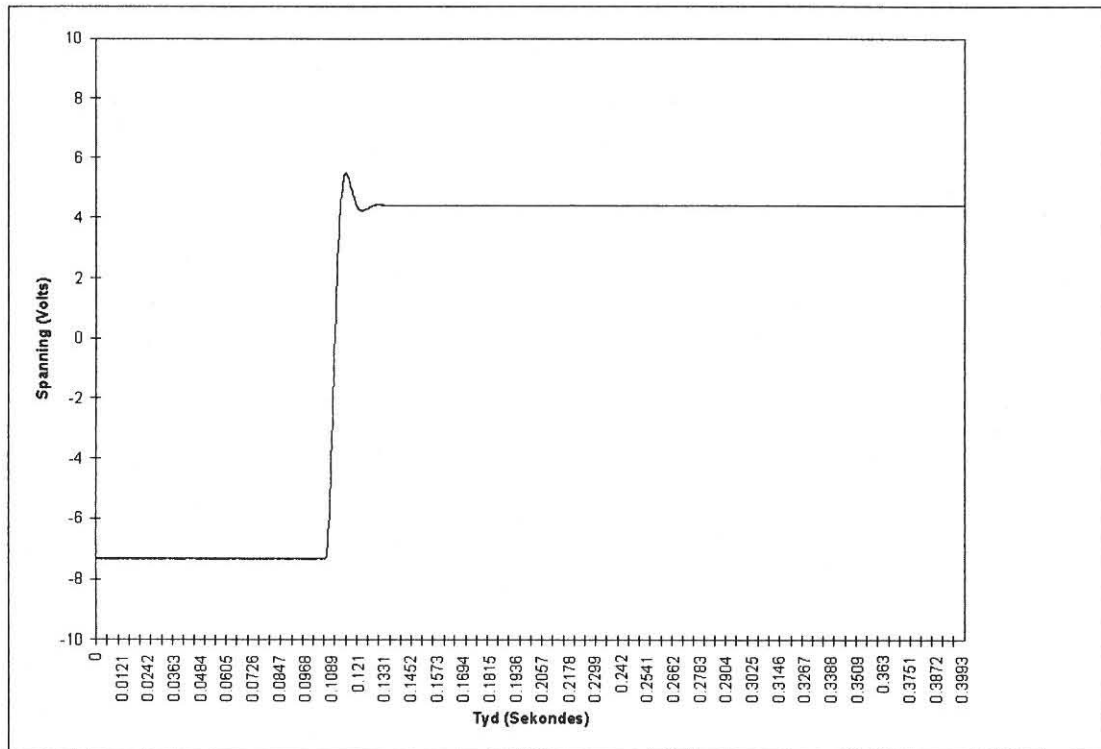


Fig. 6.9 - Tydrespons met 'n seingenerator as inset.

Die volgende bevindings is gemaak vanaf die respons in Figuur 6.9.

- 1) 9,2ms na aanskakeling van 4kHz sein is die piek van maksimum verbyskiet bereik.
- 2) Die seinprosesseringseenheid het 22,7ms na aanskakeling van 4kHz sein gestabiliseer.

Stabiliteit is dus 13,5ms na die maksimum verbyskiet bereik. Dit wil sê die totale tydsduur vir die kring om te stabiliseer vandat die 4kHz sein aan die kring gekoppel is, is 22,7ms. Die transiënt verkry met die insetsein vanaf die seingenerator is as gevolg van die

seinprosesseringseenheid en nie as gevolg van die seingenerator nie.

Die gevolgtrekking kan dus gemaak word dat die maksimum verbyskiet verkry vanaf die toetsmotor wel van die motor afkomstig is. Dit is moontlik om die gevolgtrekking te maak aangesien die motor se maksimum verbyskiet na 32ms plaasvind en die seinprosesseringseenheid volgens die berekenings reeds na 22,7ms gestabiliseer het. Die resultaat van die toets dien as 'n bewys dat die maksimum verbyskiet verkry in die evaluering van die verskeie sensors wel as gevolg van die motor onder toets was, aangesien dieselfde seinprosesseringseenheid tydens die evaluering van al die sensors gebruik is.

6.5 EVALUERING VAN DIE SAGTEWARE

Om die korrekte werking van die ontwikkelde sagteware te verseker is die snelheid en versnelling toetskondisies herhaal wat gebruik is om die verskillende sensors te evalueer. Dit wil sê 'n motor is versnel vanaf rus tot volspoed. Die as-enkodeerder is as sensor gebruik om die sagteware te toets. Figuur 6.10 toon die snelheid weergawe van die motor wat met die navorser se sagteware verkry is.

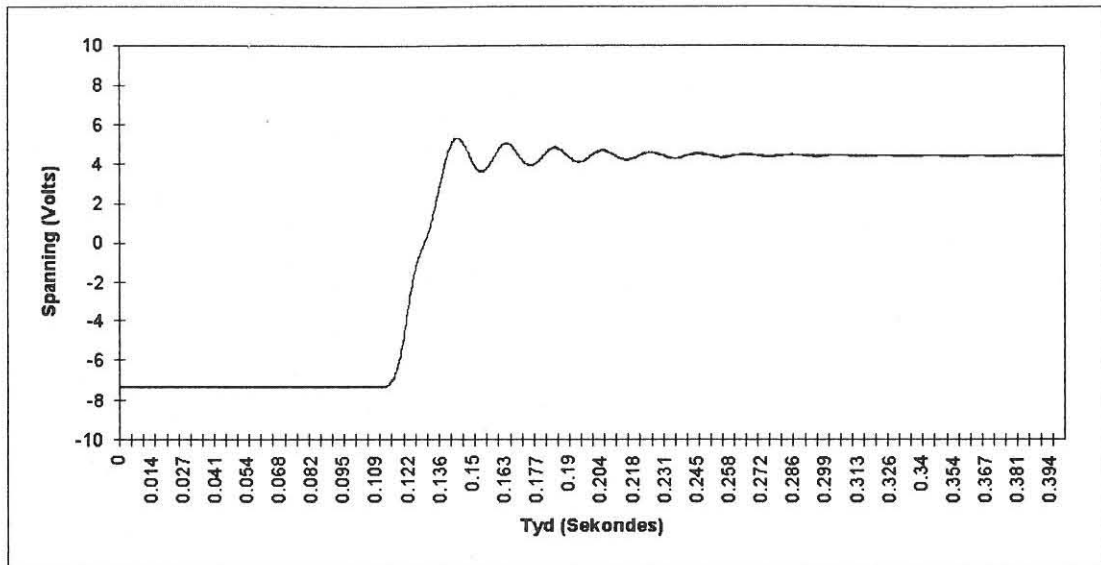


Fig. 6.10 - Snelheiduitset verkry met ontwikkelde sagteware.

Figuur 6.11 toon die versnelling weergawe van die motor wat met die navorser se sagteware verkry is.

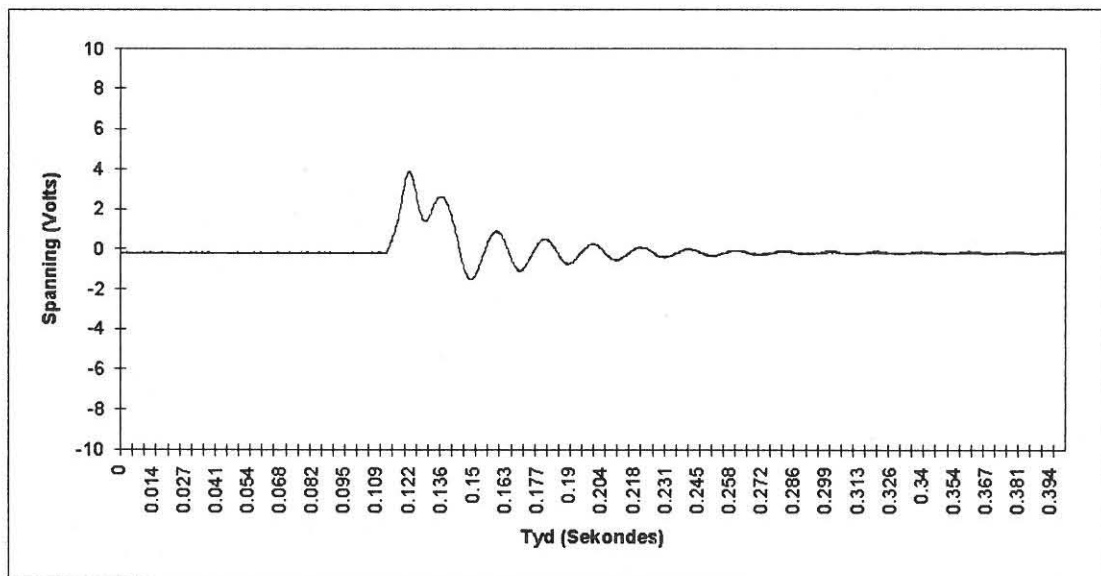


Fig. 6.11 - Versnellinguitset verkry met ontwikkelde sagteware.

Deur Figure 6.10 en 6.11 met Figure 6.5 en 6.6, wat met behulp van die STAT-30 dataverkrygingsprogram verkry is, te vergelyk kan die gevolgtrekking gemaak word dat die sagteware wat ontwikkel is

korrek funksioneer. Die gevolgtrekking word gestaaf aangesien die STAT-30 sagteware goed bewese is met wye aanwending in die industrie.

6.6 GEVOLGTREKKING

Die resultate behaal met die STAT-30 dataverkrygingsprogram en die ontwikkelde sagteware dui op die korrekte werking van die sagteware wat ontwikkel is, maar verteenwoordig nie 'n omvattende evalueringsproses nie. Daar is egter 'n wye verskeidenheid toetse met groot sukses uitgevoer waar motors vanaf rus tot volspoed versnel is.

Uit die resultate blyk dat die tweeskyf ligbron en lensstelsel bevredigende resultate lewer. Die metode van opstelling vir hierdie sensor is egter kompleks en tydrowend.

Die resultate behaal met die as-enkodeerder stem ooreen met die resultate van die tweeskyf ligbron en lensstelsel. Na evaluering van die sensors se resultate ten opsigte van die kompleksiteit van die sensor opstelling, asook die metode van koppeling van die sensor aan die toetsmotor, is die gevolgtrekking gemaak dat die as-enkodeerder die mees geskikte sensor vir die ontwikkelde meetinstrument is. Die as-enkodeerder metode lewer ook 'n uitsetpuls met lae ruis en wat geen versterking benodig nie.

Die uitset van die as-enkodeerder is baie akkuraat aangesien geen pulse verlore gaan nie. Indien die as-enkodeerder egter aan die

seinprosesseringseenheid gekoppel word gaan 'n gedeelte van die uitsetsein teen lae snelhede verlore. Die gedeelte van die sein wat verlore gaan is as gevolg van die onderdeurlaatfilter in die seinprosesseringseenheid. Die onderdeurlaatfilter keer dat die frekwensie-na-spanning omskakeling begin totdat die uitsetfrekwensie van die as-enkodeerder tot bokant 40Hz styg. As gevolg van die groot getal pulse per revolusie van die as-enkodeerder kan die snelheid van 'n stadig roterende as egter vinnig bepaal word en is die effek van die seinprosesseringseenheid op die transiënt van die roterende as minimaal.

'n Belangrike gevolgtrekking wat gemaak kan word is dat die koppelingmetode van die sensor, en die omskakeling van die uitsetsein van frekwensie-na-spanning, die kern is waarom die meetinstrument draai.

Die hardware wat ontwikkel is tydens die navorsing het aan al die vereistes voldoen. Die seinprosesseringseenheid het 'n vinnige reaksietyd en die omskakeling van frekwensie-na-spanning, wat die snelheid van die motor voorstel, was akkuraat. Die enigste nadeel van die hardware is dat dit nie rigting-sensitief is nie.

Die afhandeling van die projek het gelei tot die ontwikkeling van 'n meetinstrument met 'n eenvoudige meganiese koppeling wat nie die werking van die toetsmotor beïnvloed nie. 'n Volledige stel sagteware is ook ontwikkel om die uitsette van die meetinstrument in 'n persoonlike rekenaar in te lees om die meting van die transiënte van 'n roterende masjien te bepaal.

HOOFSTUK 7

SAMEVATTING

Die hoofdoel van die projek was om 'n meetinstrument te ontwikkel wat die hoekversnelling en snelhede van roterende masjiene kan meet.

'n Teoretiese oorsig van transiënte en transduseerders is in Hoofstuk 2 gelewer. Die frekwensie-na-spanning omsetter word ook in hierdie hoofstuk bespreek.

In Hoofstuk 3 word 'n teoretiese oorsig gelewer oor die operasionele versterker se werking as 'n integreerder, 'n differensieerder en as 'n filter. Die ontwerp van veral Butterworth onderdeurlaatfilters word ook hier beskryf.

Die hardware van die meetinstrument word in Hoofstuk 4 bespreek. Hier word die ontwerp van die sensors sowel as die seinverwerkingskringe bespreek.

Die bemonstering sagteware wat ontwikkel is word in Hoofstuk 5 bespreek. Die opstelling van die programmeerbare tydreeëlaar word ook in hierdie hoofstuk bespreek. Die data benodig om die tydreeëlaar te programmeer is in Bylae B getabuleer.

Die evaluering van die seinverwerkingskringe en die verskillende sensors is in Hoofstuk 6 beskryf. Die uitset van die tweeskyf

ligbron en lensstelsel, wat ooreenkom met die uitset van die as-enkodeerder, dui daarop dat beide sensors dieselfde doel verrig. Die as-enkodeerder het die voordeel bo die tweeskyf sensor dat die koppeling metode baie eenvoudiger is. Die resultate verkry met die navorsing se sagteware wat ooreenkom met die resultate van die STAT-30 dataverkrygingsprogram dui daarop dat die bemonsteringsprogram korrek funksioneer.

BYLAAG A

SEINPROSESSERINGSEENHEID KRINGDIAGRAM

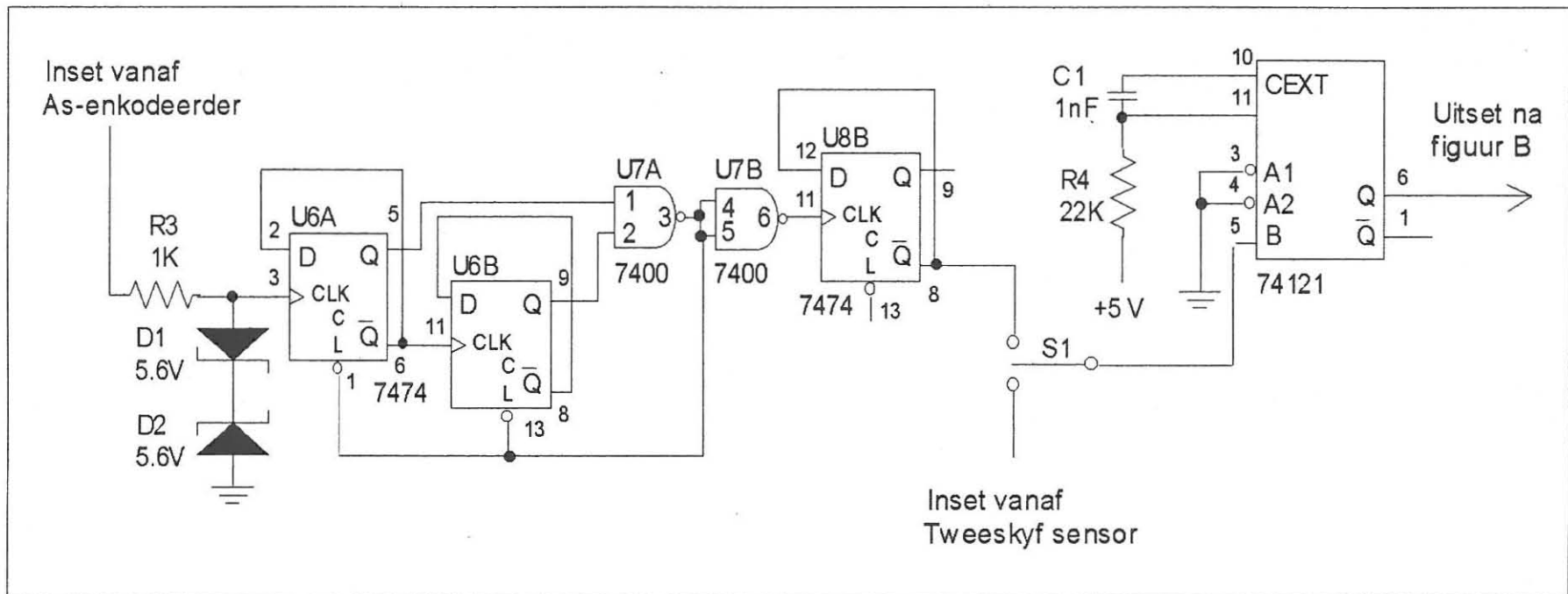


Fig. A.1 - Kringdiagram van die seinprosesseringseenheid.

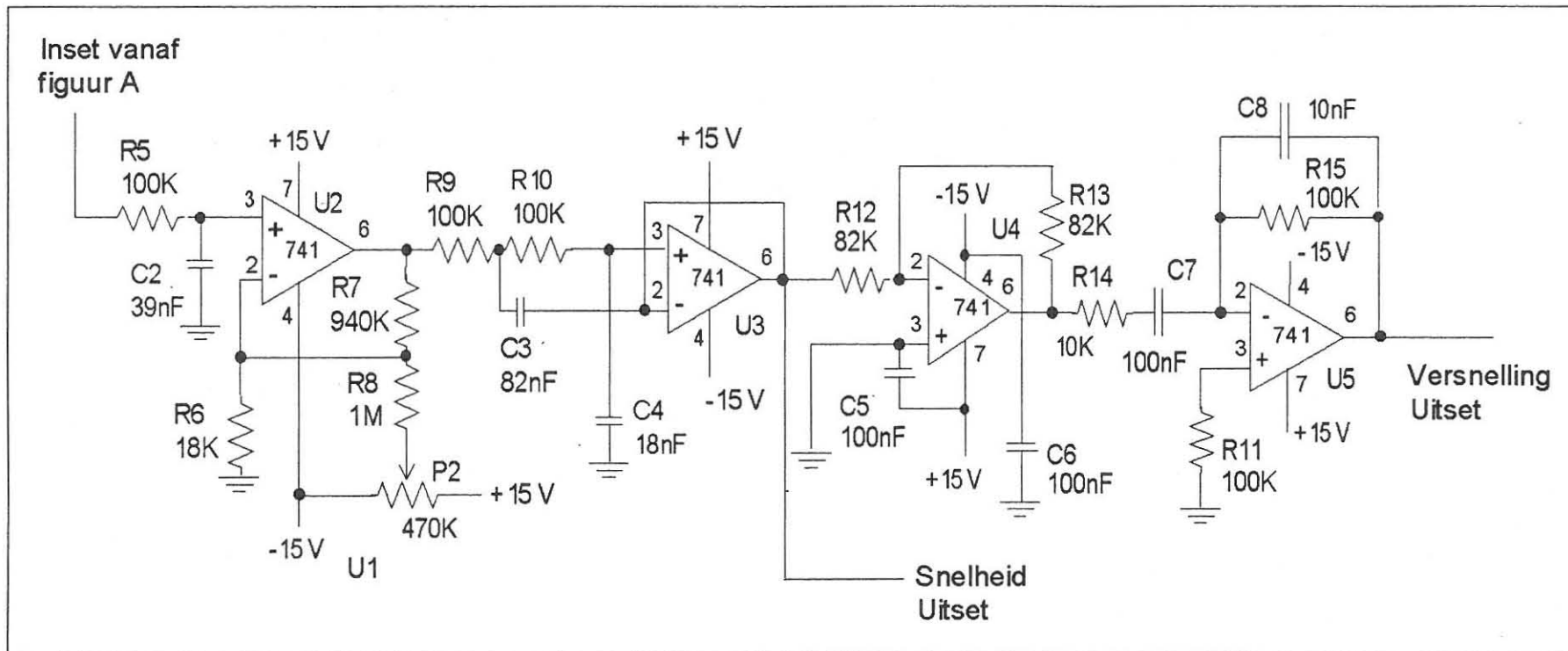


Fig. A.1 - (Vervolg) Kringdiagram van die seinprosseringseenheid.

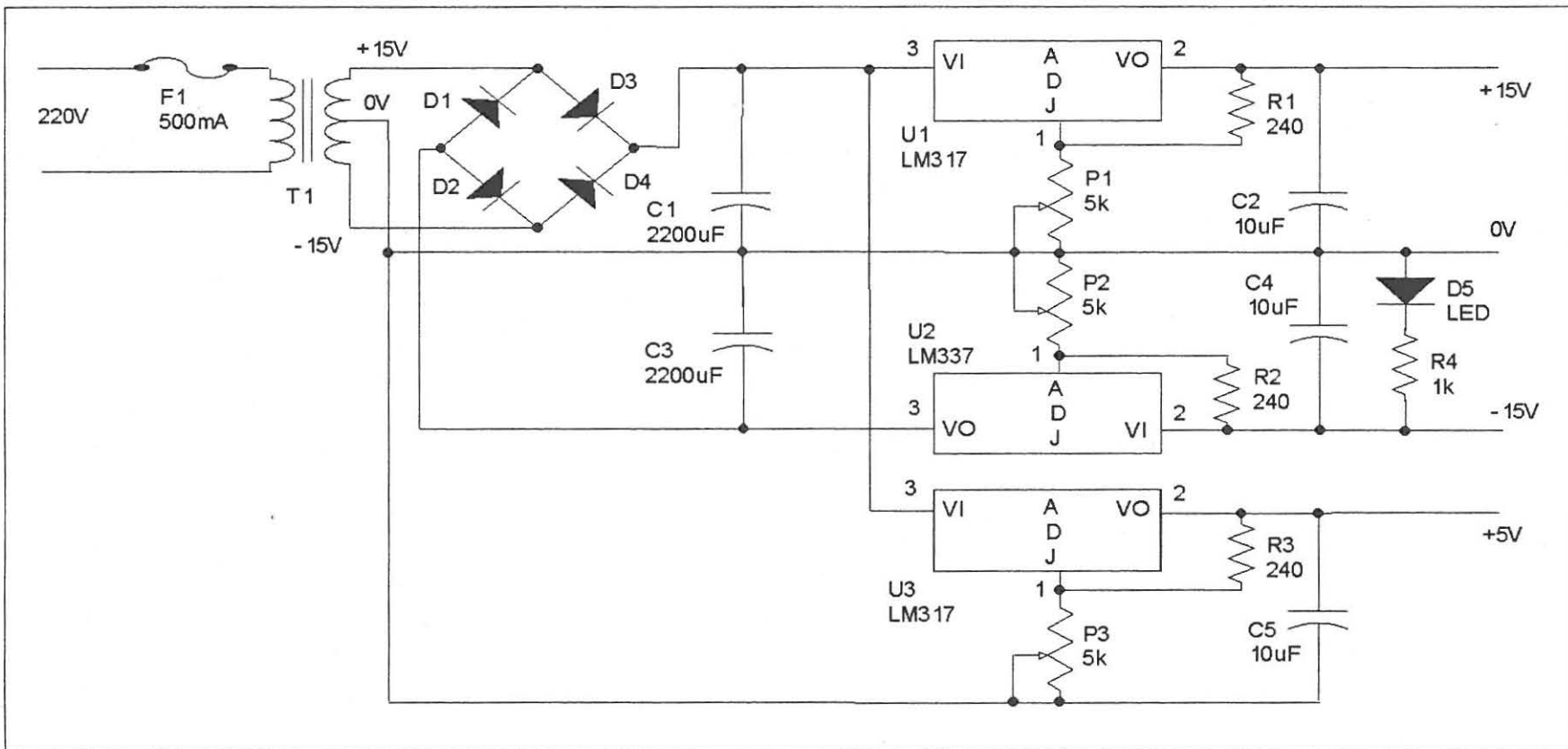


Fig. A.2 - Kragbron van die seinprosesseringseenheid.

BYLAAG B

ADRES-OPSTELLING VAN PC-30 KAART

Die opstelling van die inset-/uitsetadres spasies van die PC-30 kaart behels die gebruik van 'n basisadres, poortadres en kontrolewoorde. Die basisadres (BA) van die PC-30 kaart kan gestel word vir 700 Hex of 780 Hex deur middel van oorleiding 3 op die PC-30 kaart. Tabel B.1 toon die poortadresse wat deur die AD574 analoog-na-digitale omsetter gebruik word.

Tabel B.1 - Poortadresse van AD574 omsetter.

	Adres	Funksie
Poort A	Basisadres	Bisse A0 tot A7: 8 mins beduidende bisse van die digitale uitset met A0 die mins beduidende bis.
Poort B	Basisadres+1	Bisse B0 tot B3: 4 mees beduidende bisse van die digitale uitset met B3 die mees beduidende bis. Bis B4: TTL snellerinset.
Poort C	Basisadres+2	Bisse C4 tot C7: Kanaaladresse van die multipleksor. Bis C0: Sagteware klok snellering. Bis C1 = 1: Sagteware klok. Bis C1 = 0: Tydreëlaar klok. Bis C3 = 1: Aktiveer onderbrekingsein.

Tabel B.2 toon die verskillende poortadresse en die funksie van elke adres wat deur die PC-30 kaart gebruik word.

Tabel B.2 - Adresse van PC-30 kaart.

Poortadres	Funksie
BA+00	AD574 Inset: 8 mins beduidende bisse.
BA+01	AD574 Inset: 4 mees beduidende bisse.
BA+02	Bis 0 - bis 3: Kontrolebisse vir A/D. Bis 4 - bis 7: Multipleksor kanaal keuse.
BA+03	Kontrolewoord vir 8255 PPI (=92)
BA+04	Teller 0; 8253 tydreëlaar
BA+05	Teller 1; 8253 tydreëlaar
BA+06	Teller 2; 8253 tydreëlaar
BA+07	Kontrolewoord; 8253 tydreëlaar
BA+08	Poort A 8255 PPI 2
BA+09	Poort B 8255 PPI 2
BA+10	Poort C 8255 PPI 2
BA+11	Kontrolewoord 8255 PPI 2
BA+12	D/A-12 1 4 mins beduidende bisse.
BA+13	D/A-12 1 8 mees beduidende bisse.
BA+16	D/A-12 2 4 mins beduidende bisse.
BA+17	D/A-12 2 8 mees beduidende bisse.
BA+20	D/A-8 1 8 bis
BA+21	D/A-8 2 8 bis

BYLAAG C

HOOFPGRAM

```

{*****}
{ This is the main program that display a desktop from where a choice can }
{ be made between the different options viz. channel to sample, sample }
{ frequency and the number of samples. }
{*****}

```

```

{$C-}
{$F+}

```

```

Program TDA_1D;

```

```

Uses

```

```

  Dos, Objects, Drivers, Views,
  Menus, App, Dialogs, StdDlg,
  MsgBox, SampOpt, NumOpt, ChanOpt,
  Sample, Graph, Memory;

```

```

{ Pascal units to be linked to this program }

```

```

Const

```

cmAbout	= 100;	{ About-program command number }
cmHelp	= 101;	{ Help-program command number }
cmFreqSettings	= 112;	{ Sample frequency-program command number }
cmOptionsReset	= 113;	{ Reset-program command number }
cmNumSettings	= 114;	{ Number of samples-program command number }
cmChanSettings	= 115;	{ Select channels-program command number }
cmSample	= 120;	{ Sample-program command number }

BYLAAG C

HOOFPROGRAM

```

systemMenuChar = #240;
hcAbout        = 1000;           { About-help context constant }
hcHelp         = 1001;           { Help-help context constant }
hcFileSave     = 1002;           { Save file-help context constant }
hcFileExit     = 1003;           { Exit-help context constant }
hcSamFreq      = 1004;           { Sample frequency-help context constant }
hcNumSam       = 1005;           { Number of samples-help context constant }
hcSamChan      = 1006;           { Select channels-help context constant }
hcOpReset      = 1007;           { Reset-help context constant }

```

Type

```

{***** THint Collection *****)
PStatusLine    = ^THintStatusLine;
THintStatusLine = Object(TStatusLine)
  Function Hint(AHelpCtx : Word) : String; Virtual;
End;

{***** TScroller Collection *****)

{ ----- TLines ----- }
PLinesCollection = ^TLinesCollection;
TLinesCollection = Object(TCollection)
  Procedure FreeItem(P : Pointer); Virtual;
End;

{ ----- TLineScroller ----- }
PLineScroller    = ^TLineScroller;
TLineScroller    = Object(TScroller)
  Lines : PCollection;
  IsOK   : Boolean;
  Constructor Init(Var Bounds : TRect;

```

BYLAAG C

HOOFPROGRAM

```

        AHScrollBar, AVScrollBar : PScrollBar);
Destructor Done; Virtual;
Procedure Draw; Virtual;
Function Valid(Command : Word) : Boolean; Virtual;
End;

{ ----- TFileWindow ----- }
PFileWindow    = ^TFileWindow;
TFileWindow    = Object(TWindow)
  Constructor Init;
End;

{***** Application object *****/}
{ The application messages are received in the form of a TEvent record }
{ named Event. The messages are examined to determine if the program }
{ have to respond to these messages, if not the message can be ignored. }
{ The HandleEvent is a kind of dispatcher whose job it is to recognize }
{ events as they occur, and then either direct those event messages to }
{ another location, or call a subroutine that responds to a specific }
{ event. }
{*****}

PSamApp = ^SamApp;
SamApp  = Object(TApplication)
  Constructor Init;
  Procedure HandleEvent(Var Event : TEvent); Virtual;
  Procedure DoAboutBox; Virtual;
  Procedure InitMenuBar; Virtual;
  Procedure InitStatusLine; Virtual;
  Procedure FileOpen;
End;

```

BYLAAG C

HOOFPROGRAM

```

{***** Construct Application object ***** }

Constructor SamApp.Init;

Begin
  TApplication.Init;           { Call ancestor constructor }
  InitFreqSamOptions;         { Initialize sample frequency option }
  InitNumSamOptions;          { Initialize number of samples option }
  InitChanSamOptions;         { Initialize select channels option }
  DoFreqOptionsRetrieve;      { Recall sample frequency }
  DoNumOptionsRetrieve;       { Recall number of samples }
  DoChanOptionsRetrieve;      { Recall channels to sample }
  DisableCommands([cmSave]);
End;

{***** Construct THint Text ***** }
{ Draws the status line by writing the text string for each status item }
{ that has one, then any hints defined for the current help context. }
{***** }

Function THintStatusLine.Hint(AHelpCtx : Word) : String;

Begin
  Case AHelpCtx Of
    hcAbout      : Hint := 'Show version and copyright information.';
    hcHelp       : Hint := 'This is the Help file.';
    hcFileSave   : Hint := 'Save the sampled data from all channels.';
    hcFileExit   : Hint := 'Terminate the application.';
    hcSamFreq    : Hint := 'Select the sampling rate.';
    hcNumSam     : Hint := 'Select number of samples per channel.';
    hcSamChan    : Hint := 'Select all channels you wish to sample.';
  End;

```

BYLAAG C

HOOFPROGRAM

```

    hcOpReset   : Hint := 'Reset to default values.';
  Else
    Hint       := '';
  End;
End;

```

```

{***** About Program *****)
{ In programs with pulldown menu, the About Program dialog box typically
{ display the copyright or a personal notice.
{*****

```

```

Procedure SamApp.DoAboutBox;

```

```

Begin
  MessageBox(#3'Transient Data Acquisition'#13 +
    #3'Version 1.0D'#13 +
    #3'Copyright (c) 1994'#13 +
    #3'by Daniel S. van den Berg',
    nil, mfInformation or mfOKButton);
End;

```

```

{***** Construct TScroller Collection *****)
{ The TScroller provides a scrolling virtual window onto a larger view.
{ That is, a scrolling view lets the user scroll a large view within a
{ clipped boundary. All methods needed to provide both scroll bar and
{ keyboard scrolling are build into TScroller.
{*****

```

```

{ ----- TLinesCollection ----- }

```

```

Procedure TLinesCollection.FreeItem(P : Pointer);

```

BYLAAG C

HOOFPROGRAM

```
Begin
  DisposeStr(P);
End;

{ ----- TLineScroller ----- }

Constructor TLineScroller.Init(Var Bounds : TRect;
                               AHScrollBar,
                               AVScrollBar : PScrollBar);

Var
  F      : Text;
  Line   : String;
  MaxWidth : Integer;

Begin
  TScroller.Init(Bounds, AHScrollBar, AVScrollBar);
  GrowMode := gfGrowHiX + gfGrowHiY;
  IsOK      := True;
  Lines     := New(PLinesCollection, Init(5, 5));
  {$I-}
  Assign(F, 'Data.Txt');
  Reset(F);
  If IOResult <> 0 Then
    Begin
      MessageBox('Cannot open file Data.Txt',
                 nil, mfError + mfOkButton);
      IsOK := False;
    End
  Else
    Begin
      MaxWidth := 0;
    End
  End
End
```

BYLAAG C

HOOFPROGRAM

```
While not Eof(F) and (not LowMemory) Do
Begin
  ReadIn(F, Line);
  If Length(Line) > MaxWidth Then
    MaxWidth := Length(Line);
  Lines^.Insert( NewStr(Line));
End;
Close(F);
End;
{$I+}
SetLimit(MaxWidth, Lines^.Count);
End;
```

```
Destructor TLineScroller.Done;
```

```
Begin
  Dispose(Lines, Done);
  TScroller.Done;
End;
```

```
Procedure TLineScroller.Draw;
```

```
Var
  ZBuff      : TDrawBuffer;
  ColorAttr  : Byte;
  I          : Integer;
  AString    : String;
  StringPtr  : PString;
```

```
Begin
  ColorAttr := GetColor(1);
  For I := 0 To Size.Y - 1 Do
  Begin
```

BYLAAG C

HOOFPROGRAM

```
MoveChar(ZBuff, ' ', ColorAttr, Size.X);
If Delta.Y + 1 < Lines^.Count Then
Begin
  StringPtr := Lines^.At(Delta.Y + 1);
  If StringPtr <> nil Then
    AString := Copy(StringPtr^, Delta.X + 1, Size.X)
  Else
    AString := '';
  MoveStr(ZBuff, AString, ColorAttr);
End;
WriteLine(0, 1, Size.X, 1, ZBuff);
End;
End;

Function TLineScroller.Valid(Command : Word) : Boolean;

Begin
  Valid := IsOK;
End;

{ ----- TFileWindow ----- }

Constructor TFileWindow.Init;

Var
  R : TRect;

Begin
  Desktop^.GetExtent(R);
  TWindow.Init(R, 'Help Window', 0);
  Options := Options or ofTileable;
  GetExtent(R);
  R.Grow(-1, -1);
```


BYLAAG C

HOOFPROGRAM

```

Insert(New(PLineScroller, Init(R,
  StandardScrollBar(sbHorizontal + sbHandleKeyboard),
  StandardScrollBar(sbVertical + sbHandleKeyboard))));
End;

```

```

Procedure SamApp.FileOpen;

```

```

Var
  WindowPtr : PWindow;

```

```

Begin
  WindowPtr := PWindow(ValidView(New(PFileWindow, Init)));
  If WindowPtr <> nil Then Desktop^.Insert(WindowPtr);
End;

```

```

{***** Yes or No *****/
{ Prompt user for Yes or No answer. Return true for Yes.
{*****

```

```

Function Yes(Prompt : String) : Boolean;

```

```

Var
  YNDialog : PDialog;
  R        : TRect;

```

```

Begin
  R.Assign(0, 0, 60, 7);
  YNDialog := New(PDialog, Init(R, 'Please answer Yes or No'));
  If (YNDialog <> nil) Then With YNDialog^ Do
  Begin
    Options := Options or ofCentered;
    R.Grow(-1, -2);

```

BYLAAG C

HOOFPROGRAM

```

Insert(New(PStaticText, Init(R, ^C + Prompt)));
R.Assign(35, 4, 43, 6);
Insert(New(PButton, Init(R, '~N~o', cmNo, bfNormal)));
R.Assign(17, 4, 26, 6);
Insert(New(PButton, Init(R, '~Y~es', cmYes, bfDefault)));
Yes := Desktop^.ExecView(YNDiag) = cmYes;
Dispose(YNDiag, Done)
End;
End;

```

```

{***** Events *****)
{ Respond to events; execute program's commands }
{*****}

```

```

Procedure SamApp.HandleEvent(Var Event : TEvent);

```

```

Procedure DoFreqSettings; { Sample frequency event }

```

```

Begin
  If GetFreqSamOptions <> cmCancel Then { Get sample frequency setting }
    DoFreqOptionsSave { Save frequency setting }
End;

```

```

Procedure DoNumSettings; { Number of samples event }

```

```

Begin
  If GetNumSamOptions <> cmCancel Then { Get number of samples setting }
    DoFreqOptionsSave { Save setting }
End;

```

```

Procedure DoChanSettings; { Select channels to sample event }

```

BYLAAG C

HOOFPROGRAM

```

Begin
  If GetChanSamOptions <> cmCancel Then           { Get selected channels }
    DoFreqOptionsSave                             { Save setting }
End;

Procedure Do_File_Save;                           { Save file event }

Begin
  If GetFileSave <> cmCancel Then
    Do_Sample_File_Save;
End;

Procedure DoOptionsReset;                         { Reset options event }

Begin
  If Yes( 'Reset options to default settings?')
    Then InitFreqSamOptions;
    InitNumSamOptions;
    InitChanSamOptions;
    DoFreqOptionsSave;
End;

Procedure Sample_Interrupt;                       { Interrupt programming event }

Begin
  Sample_Inter;                                  { Interrupt sampling }
  InitVideo;                                     { Initialize graphics }
  ShowMouse;                                    { Display mouse cursor }
  EnableCommands([cmSave]);                     { Enable Save command }
End;

```

BYLAAG C

HOOFPROGRAM

```
{***** SamApp.HandleEvent ***** }
```

```
Begin
```

```
  TApplication.HandleEvent(Event);
```

```
  If Event.What = evCommand Then
```

```
    Begin
```

```
      Case Event.Command of
```

```
        cmAbout      : Begin
```

```
          DoAboutBox;
```

```
          ClearEvent(Event);
```

```
        End;
```

```
        cmHelp       : FileOpen;
```

```
        cmSave       : Do_File_Save;
```

```
        cmFreqSettings : DoFreqSettings;
```

```
        cmNumSettings : DoNumSettings;
```

```
        cmChanSettings : DoChanSettings;
```

```
        cmOptionsReset : DoOptionsReset;
```

```
        cmSample      : Sample_Interrupt;
```

```
      Else
```

```
        Exit
```

```
      End;
```

```
      ClearEvent(Event)
```

```
    End;
```

```
End;
```

```
{***** Pull-down menus ***** }
```

```
{ Initialize application's pull-down menus and display a menubar. }
```

```
{***** }
```

```
Procedure SamApp.InitMenuBar;
```

```
Var R : TRect;
```

BYLAAG C

HOOFPROGRAM

```

Begin
  GetExtent(R);
  R.B.Y := R.A.Y + 1;
  MenuBar := New(PMenuBar, Init(R, NewMenu(
    NewSubMenu('~'#240'~, hcNoContext, NewMenu(
      NewItem('~A~bout...', '', kbNoKey, cmAbout, hcAbout,
      NewItem('~H~elp', 'F1', kbF1, cmHelp, hcHelp,
      nil))),
    NewSubMenu('~F~ile', hcNoContext, NewMenu(
      NewItem('~S~ave', 'Alt-S', kbAltS, cmSave, hcFileSave,
      NewItem('E~x~it', 'Alt-X', kbAltX, cmQuit, hcFileExit,
      nil))),
    NewSubMenu('~O~ptions', hcNoContext, NewMenu(
      NewItem('Sample ~F~requency...', '', kbNoKey, cmFreqSettings, hcSamFreq,
      NewItem('~N~umber of Samples...', '', kbNoKey, cmNumSettings, hcNumSam,
      NewItem('Sample ~C~hannels...', '', kbNoKey, cmChanSettings, hcSamChan,
      NewLine(
      NewItem('R~e~set', '', kbNoKey, cmOptionsReset, hcOpReset,
      nil)))))),
    nil))))
  ))
End;

```

```

{***** Status line *****)
{ Initialize application's status line.
{*****

```

```

Procedure SamApp.InitStatusLine;

```

```

Var R : TRect;

```

BYLAAG C

HOOFPROGRAM

```
Begin
  GetExtent(R);
  R.A.Y := R.B.Y -1;
  StatusLine := New(PStatusLine, Init(R,
    NewStatusDef(0,$FFFF,
      NewStatusKey(',', kbF10, cmMenu,
        NewStatusKey('~Sample~ F9', kbF9, cmSample,
          NewStatusKey('~Alt-X~ Exit', kbAltX, cmQuit,
            NewStatusKey('~Alt+F3~ Close', kbAltF3, cmClose,
              nil))))),
    nil)
  ))
End;

Var
  Application : SamApp;

Begin
  Get_System_Clock;
  Application.Init;           { Init prepares a slew of internal variables. }
  Application.Run;           { Run starts the application. }
  Application.Done;         { Done cleans up before the program ends. }
End.
```

BYLAAG D

MONSTERFREKWENSIEPROGRAM

```
{*****}
{ This program displays a menu from where the required sample rate can }
{ be selected. Only one frequency can be selected at a time. }
{*****}
```

Unit SampOpt;

Interface

Uses Dos, Objects, Views, Dialogs, App; { Pascal units to be linked to this program }

```
{*****}
{ Options record to hold contents of Options dialog box }
{*****}
```

Type

```
FreqOptionsRec = Record
  OFrequency      : Word;           { Frequency radio button }
  OConfig         : String[65];    { Configuration path }
  SFreq           : Longint;
  ONumSamples     : Word;           { Number of samples button }
  SNum            : Integer;
  OChanSamples    : Word;           { Channel button }
  SChan          : Integer;
  SampleChannel0  : Integer;
  SampleChannel1  : Integer;
```

BYLAAG D

MONSTERFREKWENSIEPROGRAM

```
SampleChannel2 : Integer;  
SampleChannel3 : Integer;  
SampleChannel4 : Integer;  
SampleChannel5 : Integer;  
SampleChannel6 : Integer;  
SampleChannel7 : Integer;  
SampleChannel8 : Integer;  
SampleChannel9 : Integer;  
SampleChannel10 : Integer;  
SampleChannel11 : Integer;  
SampleChannel12 : Integer;  
SampleChannel13 : Integer;  
SampleChannel14 : Integer;  
SampleChannel15 : Integer;  
End;
```

```
Var  
  FreqSamOptions : FreqOptionsRec;           { Global options data }  
  SampleFrequency : Longint;
```

```
Procedure InitFreqSamOptions;  
Function FreqSampDialog: PDialog;  
Function GetFreqSamOptions: Word;  
Procedure DoFreqOptionsSave;  
Procedure DoFreqOptionsRetrieve;
```

Implementation

```
{*****}  
{Initialize global Frequency Options defaults}  
{*****}
```


BYLAAG D

MONSTERFREKWENSIEPROGRAM

```
Procedure InitFreqSamOptions;
```

```
Var
```

```
  D : DirStr;           { Directory path }
  N : NameStr;         { File name }
  E : ExtStr;          { File extension }
```

```
Begin
```

```
  FSplit(ParamStr(0), D, N, E); { Parse home directory path }
  N := 'FreqSamp';             { Change name to FreqSamp }
  E := '.CFG';                 { Change extension to CFG }
```

```
  With FreqSamOptions Do
```

```
  Begin
```

```
    OFrequency      := 8;
    OConfig         := D + N + E;
    SFreq           := 1;
    ONumSamples     := 0;           { Initialize sample number, default }
    SNum            := 8;
    OChanSamples    := $01;        { Initialize channels to sample, default }
    SChan           := 1;
    SampleChannel0  := 0;
    SampleChannel1  := 0;
    SampleChannel2  := 0;
    SampleChannel3  := 0;
    SampleChannel4  := 0;
    SampleChannel5  := 0;
    SampleChannel6  := 0;
    SampleChannel7  := 0;
    SampleChannel8  := 0;
    SampleChannel9  := 0;
    SampleChannel10 := 0;
    SampleChannel11 := 0;
    SampleChannel12 := 0;
```

BYLAAG D

MONSTERFREKWENSIEPROGRAM

```

SampleChannel13 := 0;
SampleChannel14 := 0;
SampleChannel15 := 0;
SampleFrequency := FreqSamOptions.SFreq;
End;
End;
```

```

Function FreqSampDialog: Pdialog;                                { Perform Frequency options dialog }

Var
  Dialog   : Pdialog;                                           { Pointer to Tdialog object }
  V        : Pview;                                             { For creating various controls }
  R        : Trect;                                             { For specifying various rectangles }

Begin
  R.Assign(0, 0, 36, 13);                                        { Create the dialog object }
  Dialog := New(PDialog, Init(R, 'Sample Frequency'));
  If Dialog <> nil Then With Dialog^ Do
  Begin
    Options := Options or ofCentered;                            { Center dialog window }
    R.Assign(7, 3, 30, 8);                                       { Create Sample Frequency radio buttons }
    V := New(PRadioButtons, Init(R,
      NewSItem('20kHz',
      NewSItem('10kHz',
      NewSItem('5kHz',
      NewSItem('2kHz',
      NewSItem('1kHz',
      NewSItem('500Hz',
      NewSItem('200Hz',
      NewSItem('100Hz',
      NewSItem('1Hz',
      nil))))))))))
```

BYLAAG D

MONSTERFREKWENSIEPROGRAM

```

));
Insert(V);

R.Assign(6, 10, 16, 12);           { Create Cancel and Ok buttons }
Insert(New(PButton, Init(R, 'Cancel', cmCancel, bfNormal)));
R.Assign(21, 10, 31, 12);
Insert(New(PButton, Init(R, 'O~k~', cmOk, bfDefault)));
End;
FreqSampDialog := Dialog
End;

Function GetFreqSamOptions: Word;           { Create and execute Options dialog }

Var
Dialog : Pdialog;                       { Pointer to Tdialog object }
C      : Word;                           { Holds result of dialog execution }

Begin
Dialog := FreqSampDialog;               { Make options dialog }
If Dialog <> nil Then
Begin
Dialog^.SetData(FreqSamOptions);
C := Desktop^.ExecView(Dialog);
If C <> cmCancel Then Dialog^.GetData(FreqSamOptions);
With FreqSamOptions Do
Case OFrequency of
0: SampleFrequency := 20000;
1: SampleFrequency := 10000;
2: SampleFrequency := 5000;
3: SampleFrequency := 2000;
4: SampleFrequency := 1000;
5: SampleFrequency := 500;

```

BYLAAG D

MONSTERFREKWENSIEPROGRAM

```
        6: SampleFrequency := 200;
        7: SampleFrequency := 100;
        8: SampleFrequency := 1;
    End;
    FreqSamOptions.SFreq := SampleFrequency;
    Dispose(Dialog, Done);
    GetFreqSamOptions := C { Return result to caller }
End;
End;

{$I-}
Procedure DoFreqOptionsSave; { Procedure to save Sample frequency data }

Var
    F : File of FreqOptionsRec;

Begin
    Assign(F, FreqSamOptions.OConfig);
    Rewrite(F);
    Write(F, FreqSamOptions);
    Close(F);
    If IoResult <> 0 Then {ignore error};
End;

Procedure DoFreqOptionsRetrieve; { Procedure to read Sample frequency }

Var
    F : File of FreqOptionsRec;

Begin
    Assign(F, FreqSamOptions.OConfig);
```

BYLAAG D

MONSTERFREKWENSIEPROGRAM

```
Reset(F);  
If IoResult = 0 Then  
Begin  
  Read(F,FreqSamOptions);  
  Close(F);  
  SampleFrequency := FreqSamOptions.SFreq;  
End;  
End;  
{!+}  
End.
```

BYLAAG E

MONSTERDATAPROGRAM

```
{*****}
{ This program display a menu from where the required amount of samples }
{ can be selected, e.g. 8 or 1024. }
{*****}
```

Unit NumOpt;

Interface

Uses Dos, Objects, Views, Dialogs, App; { Pascal units to be linked to this program }

```
{*****}
{ Options record to hold contents of Options dialog box }
{*****}
```

Type

```
NumOptionsRec = Record
  ONumSamples : Word; { Number of Samples radio button }
  SNum : Integer; { Configuration path }
End;
```

Var

```
NumSamOptions : NumOptionsRec; { Global options data }
SampleNumber : Integer;
```

Procedure InitNumSamOptions;

BYLAAG E

MONSTERDATAPROGRAM

```
Function NumSampDialog: PDialog;
Function GetNumSamOptions: Word;
Procedure DoNumOptionsRetrieve;
```

Implementation

Uses SampOpt, ChanOpt;

```
{*****}
{ Initialize global Number of Samples defaults }
{*****}
```

Procedure InitNumSamOptions;

Begin

```
  NumSamOptions.ONumSamples := FreqSamOptions.ONumSamples;
```

```
  NumSamOptions.SNum       := FreqSamOptions.SNum;
```

```
  With NumSamOptions Do
```

```
    Begin
```

```
      ONumSamples := 0;           { Set number of samples to default of 8 samples }
```

```
      SNum       := 8;
```

```
      SampleNumber := NumSamOptions.SNum; { Reset to last number of samples selected }
```

```
    End;
```

```
End;
```

```
Function NumSampDialog: Pdialog;           { Perform Number of Samples options dialog }
```

Var

```
  Dialog : Pdialog;           { Pointer to Tdialog object }
```

```
  V      : Pview;            { For creating various controls }
```

```
  R      : Trect;           { For specifying various rectangles }
```

BYLAAG E

MONSTERDATAPROGRAM

```

Begin
  R.Assign(0, 0, 33, 12);                                { Create the dialog object }
  Dialog := New(PDialog, Init(R, 'Number of Samples'));
  If Dialog <> nil Then With Dialog^ Do
  Begin
    Options := Options or ofCentered;                    { Center dialog window }
    R.Assign(7, 3, 26, 8);                                { Create Sample radio buttons }
    V := New(PRadioButtons, Init(R,
      NewSItem('8',
      NewSItem('16',
      NewSItem('32',
      NewSItem('64',
      NewSItem('128',
      NewSItem('256',
      NewSItem('512',
      NewSItem('1024',
      NewSItem('2048',
      NewSItem('4096',
      nil))))))))))
    ));
    Insert(V);

    R.Assign(4, 9, 14, 11);                                { Create Cancel and Ok buttons }
    Insert(New(PButton, Init(R, 'Cancel', cmCancel, bfNormal)));
    R.Assign(19, 9, 29, 11);
    Insert(New(PButton, Init(R, 'O~k~', cmOk, bfDefault)));
  End;
  NumSampDialog := Dialog
End;

Function GetNumSamOptions: Word;                          { Create and execute Options dialog }

```


BYLAAG E

MONSTERDATAPROGRAM

```

Var
  Dialog : Pdialog;           { Pointer to Tdialog object }
  C      : Word;             { Holds result of dialog execution }

Begin
  Dialog := NumSampDialog;   { Make options dialog }
  If Dialog <> nil Then
  Begin
    Dialog^.SetData(NumSamOptions);
    C := Desktop^.ExecView(Dialog);
    If C <> cmCancel Then Dialog^.GetData(NumSamOptions);
    With NumSamOptions Do
    Case ONumSamples Of
      0: SampleNumber := 8;
      1: SampleNumber := 16;
      2: SampleNumber := 32;
      3: SampleNumber := 64;
      4: SampleNumber := 128;
      5: SampleNumber := 256;
      6: SampleNumber := 512;
      7: SampleNumber := 1024;
      8: SampleNumber := 2048;
      9: SampleNumber := 4096;
    End;

    NumSamOptions.SNum      := SampleNumber;
    FreqSamOptions.ONumSamples := NumSamOptions.ONumSamples;
    FreqSamOptions.SNum     := SampleNumber;
    ValidateSampleNumber;
    Dispose(Dialog, Done);
    GetNumSamOptions       := C           { Return result to caller }
  End;

```

BYLAAG E

MONSTERDATAPROGRAM

End;

Procedure DoNumOptionsRetrieve; { Procedure to read number of samples }

Var

F: File of FreqOptionsRec;

Begin

NumSamOptions.ONumSamples := FreqSamOptions.ONumSamples;

NumSamOptions.Snum := FreqSamOptions.SNum;

SampleNumber := NumSamOptions.SNum;

End;

{ \$!+ }

End.

BYLAAG F

MONSTERKANAALPROGRAM

```
{*****}
{ This program display a menu from where the channels to be sampled }
{ can be selected. More than one channel could be selected. }
{*****}
```

Unit ChanOpt;

Interface

Uses Dos, Objects, Views, Dialogs, App;

{ Pascal units to be linked to this program }

Const

```
Chan0      = $01;
Chan1      = $02;
Chan2      = $04;
Chan3      = $08;
Chan4      = $10;
Chan5      = $20;
Chan6      = $40;
Chan7      = $80;
Chan8      = $100;
Chan9      = $200;
Chan10     = $400;
Chan11     = $800;
Chan12     = $1000;
Chan13     = $2000;
```

{ Channel options check box values }

BYLAAG F

MONSTERKANAALPROGRAM

```
Chan14    = $4000;
Chan15    = $8000;
```

```
{*****}
{ Options record to hold contents of Options dialog box }
{*****}
```

Type

```
ChanOptionsRec = Record
    OChanSamples    : Word;           { Number of Channels check boxes }
    SChan           : Integer;
    SampleChannels  : Integer;
End;
```

Var

```
SamChannel      : Integer;
ChanSamOptions  : ChanOptionsRec;   { Global options data }
SampleChannelArray : Array[0..15] of ChanOptionsRec;
```

```
Procedure InitChanSamOptions;
Function ChanSampDialog: PDialog;
Function GetChanSamOptions: Word;
Procedure ValidateSampleNumber;
Procedure Init_SampleChannel_Array;
Procedure Save_SampleChannel_Array;
Procedure DoChanOptionsRetrieve;
```

Implementation

Uses SampOpt, NumOpt;

BYLAAG F

MONSTERKANAALPROGRAM

```

{*****}
{ Initialize global ChanOptions defaults }
{*****}

Procedure InitChanSamOptions;

Begin
  ChanSamOptions.OChanSamples := FreqSamOptions.OChanSamples;
  ChanSamOptions.SChan       := FreqSamOptions.SChan;
  Init_SampleChannel_Array;
  With ChanSamOptions Do
  Begin
    OChanSamples := $01;           { Set channels to sample to default }
    SChan       := 1;             { Reset to channels previously sampled }
    SamChannel  := ChanSamOptions.SChan;
  End;
End;

Function ChanSampDialog: Pdialog;           { Perform Channels to Sample options dialog }

Var
  Dialog : Pdialog;           { Pointer to Tdialog object }
  V      : Pview;             { For creating various controls }
  R      : Trect;             { For specifying various rectangles }

Begin
  R.Assign(0, 0, 45, 14);       { Create the dialog object }
  Dialog := New(PDialog, Init(R, 'Select Channels to be Sampled'));
  If Dialog <> nil Then With Dialog^ Do
  Begin
    Options := Options or ofCentered; { Center dialog window }
    R.Assign(10, 2, 35, 10);        { Create Channels sample check boxes }
  End;
End;

```

BYLAAG F

MONSTERKANAALPROGRAM

```

V := New(PCheckBoxes, Init(R,
  NewSItem('Chan 0',
  NewSItem('Chan 1',
  NewSItem('Chan 2',
  NewSItem('Chan 3',
  NewSItem('Chan 4',
  NewSItem('Chan 5',
  NewSItem('Chan 6',
  NewSItem('Chan 7',
  NewSItem('Chan 8',
  NewSItem('Chan 9',
  NewSItem('Chan 10',
  NewSItem('Chan 11',
  NewSItem('Chan 12',
  NewSItem('Chan 13',
  NewSItem('Chan 14',
  NewSItem('Chan 15',
  nil)))))))))))))
));
Insert(V);

R.Assign(10, 11, 20, 13);                                { Create Cancel and Ok buttons }
Insert(New(PButton, Init(R, 'Cancel', cmCancel, bfNormal)));
R.Assign(25, 11, 35, 13);
Insert(New(PButton, Init(R, 'O~k~', cmOk, bfDefault)));
End;
ChanSampDialog := Dialog
End; {channel dialog}

Function GetChanSamOptions: Word;                          { Create and execute Options dialog }

Var

```

BYLAAG F

MONSTERKANAALPROGRAM

```

Dialog : Pdialog;           { Pointer to Tdialog object }
C       : Word;             { Holds result of dialog execution }
i       : Integer;

Begin
Dialog := ChanSampDialog;   { Make options dialog }
If Dialog <> nil Then
Begin
Dialog^.SetData(ChanSamOptions);
C := Desktop^.ExecView(Dialog);
If C <> cmCancel Then Dialog^.GetData(ChanSamOptions);
FreqSamOptions.OChanSamples := ChanSamOptions.OChanSamples;
SamChannel := 0;
For i := 0 to 15 Do         {Clear SampleChannelArray}
    With SampleChannelArray[i] Do
        Begin
            SampleChannels := 0;
        End;
If ChanSamOptions.OChanSamples and Chan0 <> 0   { Determine which channels are selected }
Then Begin
    With SampleChannelArray[SamChannel] Do
        Begin
            SampleChannels := 0;
        End;
        INC (SamChannel);
    End;
If ChanSamOptions.OChanSamples and Chan1 <> 0
Then Begin
    With SampleChannelArray[SamChannel] Do
        Begin
            SampleChannels := 1;
        End;
        INC (SamChannel);

```

BYLAAG F

MONSTERKANAALPROGRAM

```
End;
If ChanSamOptions.OChanSamples and Chan2 <> 0
Then Begin
  With SampleChannelArray[SamChannel] Do
  Begin
    SampleChannels := 2;
  End;
  INC (SamChannel);
End;
If ChanSamOptions.OChanSamples and Chan3 <> 0
Then Begin
  With SampleChannelArray[SamChannel] Do
  Begin
    SampleChannels := 3;
  End;
  INC (SamChannel);
End;
If ChanSamOptions.OChanSamples and Chan4 <> 0
Then Begin
  With SampleChannelArray[SamChannel] Do
  Begin
    SampleChannels := 4;
  End;
  INC (SamChannel);
End;
If ChanSamOptions.OChanSamples and Chan5 <> 0
Then Begin
  With SampleChannelArray[SamChannel] Do
  Begin
    SampleChannels := 5;
  End;
  INC (SamChannel);
End;
End;
```


BYLAAG F

MONSTERKANAALPROGRAM

```
If ChanSamOptions.OChanSamples and Chan6 <> 0
  Then Begin
    With SampleChannelArray[SamChannel] Do
      Begin
        SampleChannels := 6;
      End;
    INC (SamChannel);
  End;
If ChanSamOptions.OChanSamples and Chan7 <> 0
  Then Begin
    With SampleChannelArray[SamChannel] Do
      Begin
        SampleChannels := 7;
      End;
    INC (SamChannel);
  End;
If ChanSamOptions.OChanSamples and Chan8 <> 0
  Then Begin
    With SampleChannelArray[SamChannel] Do
      Begin
        SampleChannels := 8;
      End;
    INC (SamChannel);
  End;
If ChanSamOptions.OChanSamples and Chan9 <> 0
  Then Begin
    With SampleChannelArray[SamChannel] Do
      Begin
        SampleChannels := 9;
      End;
    INC (SamChannel);
  End;
If ChanSamOptions.OChanSamples and Chan10 <> 0
```

BYLAAG F

MONSTERKANAALPROGRAM

```
    Then Begin
        With SampleChannelArray[SamChannel] Do
            Begin
                SampleChannels := 10;
            End;
            INC (SamChannel);
        End;
    If ChanSamOptions.OChanSamples and Chan11 <> 0
    Then Begin
        With SampleChannelArray[SamChannel] Do
            Begin
                SampleChannels := 11;
            End;
            INC (SamChannel);
        End;
    If ChanSamOptions.OChanSamples and Chan12 <> 0
    Then Begin
        With SampleChannelArray[SamChannel] Do
            Begin
                SampleChannels := 12;
            End;
            INC (SamChannel);
        End;
    If ChanSamOptions.OChanSamples and Chan13 <> 0
    Then Begin
        With SampleChannelArray[SamChannel] Do
            Begin
                SampleChannels := 13;
            End;
            INC (SamChannel);
        End;
    If ChanSamOptions.OChanSamples and Chan14 <> 0
    Then Begin
```

BYLAAG F

MONSTERKANAALPROGRAM

```

        With SampleChannelArray[SamChannel] Do
            Begin
                SampleChannels := 14;
            End;
        INC (SamChannel);
    End;
If ChanSamOptions.OChanSamples and Chan15 <> 0
Then Begin
    With SampleChannelArray[SamChannel] Do
        Begin
            SampleChannels := 15;
        End;
    INC (SamChannel);
End;

ChanSamOptions.SChan := SamChannel;
FreqSamOptions.SChan := SamChannel;
Save_SampleChannel_Array;
ValidateSampleNumber;
Dispose(Dialog, Done);
GetChanSamOptions := C
End;
End;
```

```

Procedure ValidateSampleNumber;                                { Determine how many samples can be taken of }
                                                                { each channel selected }
Begin
If ((SamChannel>2) and (SamChannel<=4)) and (FreqSamOptions.SNum=4096)
Then Begin
    SampleNumber                := 2048;
    FreqSamOptions.SNum         := 2048;
    NumSamOptions.ONumSamples  := 8;
```

BYLAAG F

MONSTERKANAALPROGRAM

```
        FreqSamOptions.ONumSamples := 8;
    End;
If ((SamChannel>4) and (SamChannel<=8)) and (FreqSamOptions.SNum>=2048)
Then Begin
    SampleNumber           := 1024;
    FreqSamOptions.SNum    := 1024;
    NumSamOptions.ONumSamples := 7;
    FreqSamOptions.ONumSamples := 7;
End;
If ((SamChannel>8) and (SamChannel<=16)) and (FreqSamOptions.SNum>=1024)
Then Begin
    SampleNumber           := 512;
    FreqSamOptions.SNum    := 512;
    NumSamOptions.ONumSamples := 6;
    FreqSamOptions.ONumSamples := 6;
End;
End;
```

Procedure Init_SampleChannel_Array;

```
Begin
SampleChannelArray[0].SampleChannels := FreqSamOptions.SampleChannel0;
SampleChannelArray[1].SampleChannels := FreqSamOptions.SampleChannel1;
SampleChannelArray[2].SampleChannels := FreqSamOptions.SampleChannel2;
SampleChannelArray[3].SampleChannels := FreqSamOptions.SampleChannel3;
SampleChannelArray[4].SampleChannels := FreqSamOptions.SampleChannel4;
SampleChannelArray[5].SampleChannels := FreqSamOptions.SampleChannel5;
SampleChannelArray[6].SampleChannels := FreqSamOptions.SampleChannel6;
SampleChannelArray[7].SampleChannels := FreqSamOptions.SampleChannel7;
SampleChannelArray[8].SampleChannels := FreqSamOptions.SampleChannel8;
SampleChannelArray[9].SampleChannels := FreqSamOptions.SampleChannel9;
SampleChannelArray[10].SampleChannels := FreqSamOptions.SampleChannel10;
```

BYLAAG F

MONSTERKANAALPROGRAM

```
SampleChannelArray[11].SampleChannels := FreqSamOptions.SampleChannel11;  
SampleChannelArray[12].SampleChannels := FreqSamOptions.SampleChannel12;  
SampleChannelArray[13].SampleChannels := FreqSamOptions.SampleChannel13;  
SampleChannelArray[14].SampleChannels := FreqSamOptions.SampleChannel14;  
SampleChannelArray[15].SampleChannels := FreqSamOptions.SampleChannel15;  
End;
```

```
Procedure Save_SampleChannel_Array;
```

```
Begin
```

```
  FreqSamOptions.SampleChannel0 := SampleChannelArray[0].SampleChannels;  
  FreqSamOptions.SampleChannel1 := SampleChannelArray[1].SampleChannels;  
  FreqSamOptions.SampleChannel2 := SampleChannelArray[2].SampleChannels;  
  FreqSamOptions.SampleChannel3 := SampleChannelArray[3].SampleChannels;  
  FreqSamOptions.SampleChannel4 := SampleChannelArray[4].SampleChannels;  
  FreqSamOptions.SampleChannel5 := SampleChannelArray[5].SampleChannels;  
  FreqSamOptions.SampleChannel6 := SampleChannelArray[6].SampleChannels;  
  FreqSamOptions.SampleChannel7 := SampleChannelArray[7].SampleChannels;  
  FreqSamOptions.SampleChannel8 := SampleChannelArray[8].SampleChannels;  
  FreqSamOptions.SampleChannel9 := SampleChannelArray[9].SampleChannels;  
  FreqSamOptions.SampleChannel10 := SampleChannelArray[10].SampleChannels;  
  FreqSamOptions.SampleChannel11 := SampleChannelArray[11].SampleChannels;  
  FreqSamOptions.SampleChannel12 := SampleChannelArray[12].SampleChannels;  
  FreqSamOptions.SampleChannel13 := SampleChannelArray[13].SampleChannels;  
  FreqSamOptions.SampleChannel14 := SampleChannelArray[14].SampleChannels;  
  FreqSamOptions.SampleChannel15 := SampleChannelArray[15].SampleChannels;  
End;
```

BYLAAG F

MONSTERKANAALPROGRAM

Procedure DoChanOptionsRetrieve;

Var

F: File of FreqOptionsRec;

Begin

ChanSamOptions.OChanSamples := FreqSamOptions.OChanSamples;

ChanSamOptions.SChan := FreqSamOptions.SChan;

Init_SampleChannel_Array;

SamChannel := ChanSamOptions.SChan;

End;

{\$!+}

End.

BYLAAG G

MONSTERPROGRAM

```

{$C-}
{$F+}
{*****}
{ With this unit the following subprograms are run. }
{ 1 - Determine clock speed of system }
{ 2 - Program 8253 timer }
{ 3 - Opens a new window to draw graph of results }
{*****}

Unit Sample;

Interface

Uses Dos, Objects, Views, Dialogs, App, Crt, Graph;           { Pascal units to be linked to this program }

Type
  ScreenImage = Array [ 0..1999 ] of word;                   { Screen record to save screen }

  FrameRec = Record
    UpperLeft, LowerRight : Word;
    ScreenMemory           : ScreenImage;
  End;

Const
  MaxSamples = 8200;                                         { Maximum amount of samples to be taken }

```

BYLAAG G

MONSTERPROGRAM

```

Var
  Done           : Boolean;           { Indicate when process is completed }
  VideoSegment  : Word;
  SnapShot      : ^ScreenImage;
  FrameStore    : Array [ 1..10 ] of ^FrameRec;
  DataSample    : Array [ 0..MaxSamples ] of Integer;
  WindowNumber  : Byte;
  i,j, ch, chtm : Integer;           { Misc variables for general use }
  S_Clock       : Real;              { Variable for system clock }
  S             : String[20];
  Sample_File   : Text;
  FileExists    : Integer;           { To determine if filename already exist }
  File_Save     : String[30];        { Name of file to be saved }

```

```

Procedure Get_System_Clock;
Procedure Sample_Inter;
Procedure MixedModes;
Function FileSaveDialog: PDialog;
Function GetFileSave: Word;
Procedure Do_Sample_File_Save;

```

Implementation

```

Uses SampOpt, NumOpt, ChanOpt;

```

```

Procedure Get_System_Clock;           { Determine system clock speed }

```

```

Var
  CatStr       : String[2];
  CatTemp      : Byte;
  ErrCode      : Integer;

```


BYLAAG G

MONSTERPROGRAM

```

Begin
  Clrscr;
  Repeat
    GotoXY(19,12);
    Write('Enter System clock frequency in MHz :      ');
    GotoXY(24,16);
    Write(' Between 4Mhz and 25Mhz. ');
    GotoXY(19,12);
    Write('Enter System clock frequency in MHz : ');
    Readln(CatStr);
    Val(CatStr, CatTemp, ErrCode)
  Until (ErrCode = 0) and
        (CatTemp In [ 4..25]);
  S_Clock := CatTemp;
End;

Procedure OpenWindow( UpLeftX, UpLeftY, LoRightX, LoRightY : Byte ); { Open a new window to draw graph
}

Begin
  WindowNumber := WindowNumber + 1;
  New( FrameStore[ WindowNumber ] );
  With FrameStore[ WindowNumber ]^ Do
  Begin
    ScreenMemory := SnapShot^;
    UpperLeft    := WindMin;
    LowerRight   := WindMax;
  End;
  Window( UpLeftX, UpLeftY, LoRightX, LoRightY );
End;

```

BYLAAG G

MONSTERPROGRAM

```

Procedure CloseWindow;                                { Restore window to original window }

Begin
  With FrameStore[ WindowNumber ]^ Do
  Begin
    SnapShot^ := ScreenMemory;
    Window( (Lo(UpperLeft)+1), (Hi(UpperLeft)+1),
            (Lo(LowerRight)+1), (Hi(LowerRight)+1) );
  End;
  Dispose( FrameStore[ WindowNumber ] );
  WindowNumber := WindowNumber - 1;
End;

Procedure SetVideoSegment;                            { Determine screen type e.g. colour or mono }

Begin
  If LastMode = 7 { Mono } Then
    VideoSegment := $B000
  Else
    VideoSegment := $B800;
End;

Procedure MixedModes;                                { Draw X, Y axes on screen for the graph }
                                                { and draw the graph on it }

Const
  Esc = #27;

Var
  GraphDriver, GraphMode                : Integer;    { Misc variables to draw graph }
  XCenter, YCenter                      : Integer;
  X, Y, Z, N, M                         : Integer;
  YValue, XValue                        : Integer;
  YValue_Old, YValue_New, YValue_Current : Integer;

```

BYLAAG G

MONSTERPROGRAM

```
XValue_Old, XValue_New, XValue_Current : Integer;  
X_Value, Y_Value                       : Real;  
D                                       : Char;
```

Begin

```
SetVideoSegment;  
GraphDriver := Detect;  
SnapShot := Ptr( VideoSegment, $0000 );  
WindowNumber := 0;  
OpenWindow( 3, 3, 3, 3 );  
InitGraph(GraphDriver, GraphMode, "");  
XCenter := GetMaxX div 2;  
YCenter := GetMaxY div 2;  
Line(70, YCenter, GetMaxX, YCenter);  
Line(70, 30, 70, (GetMaxY-30));  
OutText(' Press <Enter>...to SAMPLE again.');
```

```
OutText(' <Any Key>...Return to Desktop.');
```

```
OutTextXY(0, YCenter - 50, ' V ');  
OutTextXY(0, YCenter - 25, ' O ');  
OutTextXY(0, YCenter, ' L ');  
OutTextXY(0, YCenter + 25, ' T ');  
OutTextXY(0, YCenter + 50, ' S ');  
OutTextXY(70, GetMaxY-22, '0');
```

```
Str(SampleNumber, S);  
OutTextXY(GetMaxX-95, GetMaxY-18, S );  
OutTextXY(GetMaxX-73, GetMaxY-18, ' Samples ');  
OutTextXY(XCenter-20, GetMaxY-9, ' T I M E ');  
YValue := (GetMaxY - 60) div 10;  
For X := 0 to 10 Do  
  Line(65, 34 + YValue * X, 70, 34 + YValue * X);  
MoveTo(18, 30);  
OutText(' 10.0 ');  
MoveRel(-40, YValue);
```

BYLAAG G

MONSTERPROGRAM

```
OutText(' 8.0 ');
MoveRel(-40, YValue);
OutText(' 6.0 ');
MoveRel(-40, YValue);
OutText(' 4.0 ');
MoveRel(-40, YValue);
OutText(' 2.0 ');
MoveRel(-40, YValue);
OutText(' 0.0 ');
MoveRel(-48, YValue);
OutText(' -2.0 ');
MoveRel(-48, YValue);
OutText(' -4.0 ');
MoveRel(-48, YValue);
OutText(' -6.0 ');
MoveRel(-48, YValue);
OutText(' -8.0 ');
MoveRel(-56, YValue);
OutText(' -10.0 ');
X_Value := (GetMaxX - 70) / (SampleNumber - 1);
Y_Value := (GetMaxY - 60) / 4096;
For ch := 1 to SamChannel Do
Begin
  M := ch - 1;
  MoveTo(70, YCenter);
  SetColor(15);
  For Y := 0 To (SampleNumber -1) Do
  Begin
    XValue_Old := GetX;
    YValue_Old := GetY;
    MoveTo(70, (GetMaxY-30));
    YValue_New := Round(Y_Value * DataSample[(Y * SamChannel) + M]);
    MoveRel((Round(X_Value * Y)), (YValue_New*-1));
```

BYLAAG G

MONSTERPROGRAM

```
XValue_Current := GetX;
YValue_Current := GetY;
MoveTo(XValue_Old, YValue_Old);
LineTo(XValue_Current, YValue_Current);
SetColor(ch);
End;
End;
Repeat
Until Keypressed;

D := Readkey;
If D = #13 Then
Begin
  CloseGraph;
  CloseWindow;
  Sample_Inter;
End
Else
Begin
  CloseGraph;
  CloseWindow;
End;
End;

Procedure Disable_IRQx(IRQ : Byte);           { This procedure disable the IRQ line number 5 }

Var
  Imr, Mask : Integer;

Begin
  Mask      := (1 Shl IRQ);
  Imr       := Port[$21];                     { Get interrupt mask register from 8259 }
  Imr       := Imr Or Mask;                   { Set mask bit of IRQ }
```

BYLAAG G

MONSTERPROGRAM

```

    Port[$21] := Imr;                                     { Return to controller }
End;

Procedure Enable_IRQx(IRQ : Byte);                       { This procedure enable the IRQ line number 5 }

Var
    Imr, Mask : Integer;

Begin
    Mask      := Not (1 Shl IRQ);
    Imr       := Port[$21];                             { Get interrupt mask register from 8259 }
    Imr       := Imr And Mask;                          { Clear mask bit of IRQ }
    Port[$21] := Imr;                                   { Return to controller }
End;

{*****}
{ This procedure allows the user to program the on-board 8253 timer to }
{ the required interval in order to control the sampling frequency of the }
{ PC-30 board. }
{*****}

Procedure Prog8253Timer;

Var
    Count0, Count1          : Integer;
    LBCount0, HBCount0, LBCount1, HBCount1 : Byte;
    Factor                   : Real;

Begin
    Factor := (S_Clock * 250000.0)/SampleFrequency;
    Count0 := TRUNC(Factor/32767) + 1;
    If Count0 <= 1 Then Count0 := 2;
    Count1 := ROUND(Factor/Count0);

```

BYLAAG G

MONSTERPROGRAM

```

LBCount0 := LO(Count0);           { Calculate low and high byte values for the counters }
HBCount0 := HI(Count0);
LBCount1 := LO(Count1);
HBCount1 := HI(Count1);
Port[$707] := $34;                { Program Counter 0, Control word counter 0 }
Port[$704] := LBCount0;           { Low byte counter 0 }
Port[$704] := HBCount0;           { High byte counter 0 }
Port[$707] := $74;                { Program Counter 1, Control word counter 1 }
Port[$705] := LBCount1;
Port[$705] := HBCount1;
Port[$707] := $B4;                { Program Counter 2, Control word counter 2 }
Port[$706] := $2;
Port[$706] := $0;
End;

Procedure Int_Handler;

Begin
  For chtm := 0 To (SamChannel- 1) Do
  Begin
    Port[$702] := (SampleChannelArray[ chtm ].SampleChannels Shl 4) + $2;
    { Channel selection and clear software-clock bit }
    Port[$702] := (SampleChannelArray[ chtm ].SampleChannels Shl 4) + $3;
    { Channel selection and set of software-clock bit }

    For I := 1 To 20 Do
    Begin
      End;
    DataSample[J] := (PORT[$701] And $0F) * 256 + PORT[$700];
    Inc(J);
  End;
  chtm := 0;
  Port[$702] := (SampleChannelArray[ chtm ].SampleChannels Shl 4) + $8;
End;

```

BYLAAG G

MONSTERPROGRAM

Procedure Interrupt_5_Service_Routine;

Begin

```

Inline ($1E/
  $50/
  $53/
  $51/
  $52/
  $57/
  $56/
  $06);

```

```

{ PUSH DS }
{ PUSH AX }
{ PUSH BX }
{ PUSH CX }
{ PUSH DX }
{ PUSH DI }
{ PUSH SI }
{ PUSH ES }

```

Int_Handler;

If J > ((SampleNumber*SamChannel)-1) Then

Begin

Port[\$702] := 2;

Done := True;

End;

PORT [\$0020] := \$20;

Inline (\$07/

\$5E/

\$5F/

\$5A/

\$59/

\$5B/

\$58/

\$1F/

\$FB/

\$CF);

```

{ nonspecific EOI to the 8259 PIC }
{ POP ES }
{ POP SI }
{ POP DI }
{ POP DX }
{ POP CX }
{ POP BX }
{ POP AX }
{ POP DS }
{ STI }
{ IRET - return from interrupt }
{ End of PROCEDURE

```

End;

Interrupt_5_Service_Routine }

BYLAAG G

MONSTERPROGRAM

```

Procedure Set_IVT(entry : Integer);           { entry corresponds with the number of the }
                                             { interrupt line IRQ to enter in IVT}

Var
  offset, segment, first_word, second_word : Integer;

Begin
  offset           := OFS(Interrupt_5_Service_Routine) + 7;
  segment          := Cseg;
  first_word       := (entry + 8) * 4;
  second_word      := first_word + 2;
  MEMW[$0000:first_word] := offset;
  MEMW[$0000:second_word] := segment;
End;

Function Yes(Prompt : String) : Boolean;     { Prompt user for Yes or No. Return true for Yes }

Var
  YNDialog : PDialog;
  R         : TRect;

Begin
  R.Assign(0, 0, 60, 7);
  YNDialog := New(PDialog, Init(R, 'Please answer Yes or No'));
  If (YNDialog <> nil) Then With YNDialog^ Do
  Begin
    Options := Options or ofCentered;
    R.Grow(-1, -2);
    Insert(New(PStaticText, Init(R, ^C + Prompt)));
    R.Assign(35, 4, 43, 6);
    Insert(New(PButton, Init(R, '~N~o', cmNo, bfNormal)));
    R.Assign(17, 4, 26, 6);
    Insert(New(PButton, Init(R, '~Y~es', cmYes, bfDefault)));
  End;

```

BYLAAG G

MONSTERPROGRAM

```

    Yes := Desktop^.ExecView(YNDiag) = cmYes;
    Dispose(YNDiag, Done)
End;
End;

Function FileSaveDialog : PDialog;                                { Perform Samp options dialog }

Var
    Dialog : PDialog;
    V      : PView;
    R      : TRect;

Begin                                                            { Create the dialog object }
    R.Assign(0, 0, 44, 10);
    Dialog := New(PDialog, Init(R, 'File Save'));
    If Dialog <> nil Then With Dialog^ Do
        Begin
            Options := Options or ofCentered;
            R.Assign(17, 3, 41, 4);
            V := New(PInputLine, Init(R, 15));
            Insert(V);
            R.Assign(2, 3, 15, 4);
            Insert(New(PLabel, Init(R, 'Filename :', V)));
            R.Assign(7, 6, 17, 8);                                { Create Cancel and Ok buttons }
            Insert(New(PButton, Init(R, 'Cancel', cmCancel, bfNormal)));
            R.Assign(27, 6, 37, 8);
            Insert(New(PButton, Init(R, 'O~k~', cmOk, bfDefault)));
        End;
    FileSaveDialog := Dialog
End;

```

BYLAAG G

MONSTERPROGRAM

```

Function GetFileSave : Word;                                { Create and execute Options dialog }

Var
  Dialog : PDialog;
  C      : Word;

Begin
  Dialog := FileSaveDialog;
  If Dialog <> nil Then
  Begin
    Dialog^.SetData(File_Save);
    C := Desktop^.ExecView(Dialog);
    If C <> cmCancel Then Dialog^.GetData(File_Save);
    Dispose(Dialog, Done);
    GetFileSave := C
  End;
End;

Procedure SaveData;

Var
  Counter : Integer;

Begin
  Rewrite(Sample_File);
  For ch := 0 to (SamChannel-1) Do
  Begin
    Write(Sample_File, "Channel ");
    Write(Sample_File, SampleChannelArray[ch].SampleChannels);
    Write(Sample_File, ' : Volts');
    Writeln(Sample_File, ' ');
  End;
  Writeln(Sample_File, ' ');

```

BYLAAG G

MONSTERPROGRAM

```

Write(Sample_File, "Channel : ");
For ch := 0 to (SamChannel-1) Do
Write(Sample_File, ' ', ' ', SampleChannelArray[ch].SampleChannels );
Writeln(Sample_File, ' ');
Writeln(Sample_File, ' ');
Counter := 0;
J := 0;
Repeat
Write(Sample_File, Counter);
Inc(Counter);
For ch := 1 To SamChannel Do
Begin
Write(Sample_File, ' ; ', DataSample[J]);
Inc(J);
End;
Writeln(Sample_File, ' ');
Until J = SampleNumber * SamChannel;
Close(Sample_File);
End;

Procedure Do_Sample_File_Save;

Var
AboutPtr : PDialog;
R : TRect;

Begin
Assign(Sample_File, File_Save+'.TXT');
{$I-}
Reset(Sample_File);
{$I+}
FileExists := IOResult;
If FileExists = 0 Then

```

BYLAAG G

MONSTERPROGRAM

```

Begin
  If Yes( ' File already EXISTS. Overwrite ?') Then
    SaveData;
  End
  Else
    SaveData;
End;

Procedure Sample_Inter;

Begin
  Port [$703] := $92;           { Initialize 8255 }
  Port [$702] := 2;
  Done := False;
  For J := 0 To MaxSamples Do  { Clear array }
    DataSample[J] := 0;
  J := 0;
  Prog8253Timer;
  Disable_IRQx(0);
  Set_IVT(5);                   { Initialize new interrupt procedure }
  Enable_IRQx(5);
  chtm := 0;
  Port[$702] := (SampleChannelArray[ chtm ].SampleChannels Shl 4) + $8;
  Repeat Until Done;
  Disable_IRQx(5);
  Enable_IRQx(0);
  Mixedmodes;                   { Draw graph }
End;
End.

```

BYLAAG H

HELPSUBPROGRAM

***** TRANSIENT DATA ACQUISITION VER 1.0D *****

*** MENU ***

To select a menu press Alt + F or O. For example press Alt+F to select the file menu or press Alt+O to select the option menu. To move from one menu to another, either use the cursor keys or press Alt + F or O.

*** SAMPLE FREQUENCY ***

Press <Tab> to start operation. Using the cursor keys, move the radio button to select the sampling frequency of the A/D convertor. Pressing <Esc> aborts, and pressing <Enter> makes that the new sampling frequency.

BYLAAG H

HELPSUBPROGRAM

*** NUMBER OF SAMPLES ***

Press <Tab> to start operation. Using the cursor keys, move the radio button to select the number of samples required per selected channel. Pressing <Esc> aborts, and pressing <Enter> keep the number of samples.

*** SELECTING SAMPLED CHANNELS ***

Press <Tab> to start operation. Using the cursor keys move the highlighted bar to the channel(s) to be sampled. Use the <space> key to mark the check boxes of the channels to be sampled. Select all channels you wish to sample and press <Enter> to keep the changes. Pressing <Esc> will abort the selection process.

BYLAAG H

HELPSUBPROGRAM

*** RESET ***

Set the system to default system settings : Sample Frequency = 1Hz;

Number of Samples = 8;

Channel to sample = Ch 0.

*** USING THE MOUSE ***

To move the cursor, simply move the mouse. To access the menu click either button. To select a menu choice, click the left button. In general clicking the left button substitutes for <Enter>. To use the mouse, a mouse driver must be loaded.

*** SAVE TEXT FILE ***

This saves the sampled data from all channels in a text file format. The

BYLAAG H

HELPSUBPROGRAM

fields in the text file is delimited by semicolon. This format is compatible with almost all spreadsheets.

***** EXIT *****

Selecting this option will quit the program and return to DOS.

LITERATUURLYS

1. **Brindley, K.** Sensors and Transducers. London: Heinemann Professional Publishing, 1988.
2. **Clayton, G.B. and Newby, B.W.G.** Operational Amplifiers. 3rd ed. Oxford: Newnes, 1992.
3. **De Sa, A.** Principles of Electronic Instrumentation. 2nd ed. Great Britain: Chapman and Hall, Inc., 1990.
4. **Franco, S.** Design with Operational Amplifiers and Analog Integrated Circuits. New York: McGraw-Hill Book Company, 1988.
5. **Horowitz, P and Hill, W.** The Art of Electronics. 2nd ed. Cambridge: Press Syndicate of the University of Cambridge, 1993.
6. **Hughes, A.** Electric Motors and Drives. Oxford: Heinemann Newnes, 1990.
7. **Hughes, F.W.** Op Amp Handbook. Englewood Cliffs: Prentice-Hall, Inc., 1981.
8. **Ibrahim, K.F.** Electronic Systems + Techniques. 2nd ed. England: Longman Scientific + Technical, 1994.

9. Keynes, M. Instrumentation Units 5,6 and 7. The Open University: Walton Hall, 1974.
10. Klimaschewski, J. Process Control. Pulse, March 1989, pp. 35-40.
11. Kuo, C.K. Automatic Control Systems. 4th ed. Englewood Cliffs: Prentice Hall, Inc., 1982.
12. Loveday, G.C. Electronic Testing + Fault Diagnosis. 3rd ed. England: Longman Scientific + Technical, 1995.
13. Marston, R.M. Optoelectronics Circuits Manual. Oxford: Heinemann Newnes, 1988.
14. Millman, J. and Halkias, C.C. Integrated Electronics: Analog and Digital Circuits and Systems. Tokyo: McGraw-Hill Book Company, 1985.
15. National Semiconductor. LM2917 Frequency-to-Voltage Converter. Electronic Digest, Summer 1985, pp. 25-29.
16. Niewiadomski, S. Filter Handbook: A Practical Design Guide. Oxford: Heinemann Newnes, 1989.
17. Oblack, R. Capture Transient events. Pulse, May 1989, pp. 41-50.

18. Sage, P. Transducers and Systems. Pulse, October 1990, pp. 24-31.
19. Savant, C.J., Roden, M.S. and Carpenter, G.L. Electronic Design: Circuits and Systems. 3rd ed. Redwood City, California: Benjamin/Cummings Publishing Company, Inc., 1991.
20. Spectratech. RS Data Library. Norwich: Jarrold Printing, 1994.
21. Stephenson, J.M. New Low-Noise Tachometer. Proc. IEE, vol. 116, no. 11, Nov 1969, pp. 1981-1983.
22. Tedeschi, F.P. The Active Filter Handbook. United States of America: Tab Books Inc., 1979.
23. Tobey, G.E., Graeme, J.G. and Huelsman, L.P. Operational Amplifiers: Design and Applications. Auckland: McGraw-Hill Book Company, 1989.
24. Usher, M.J. Sensors and Transducers. London: MacMillan Publishers Ltd., 1985.
25. Williams, A.B. and Taylor, F.J. Electronic Filter Design Handbook: LC, Active and Digital Filters. 2nd ed. New York: McGraw-Hill Publishing Company, 1988.

26. Wobshall, D. Circuit Design for Electronic Instrumentation: Analog and Digital Devices from Sensor to Display. 2nd ed. New York: McGraw-Hill Book Company, 1987.

ADDISIONELE BRONNE

(GERAADPLEEG, MAAR NIE NA VERWYS NIE)

1. **Antoniou, A.** Digital Filters Analysis and Design. United States of America: McGraw-Hill, Inc., 1979.
2. **Cemus, J. and Hamata, V.** Transient Stability Analysis of Synchronous Motors. New York: Elsevier Science Publishing Company, Inc., 1990.
3. **Cooper, W.D. and Helfrick, A.D.** Electronic Instrumentation and Measurement Techniques. 3rd ed. Englewood Cliffs: Prentice-Hall, Inc., 1985.
4. **Floyd, T.L.** Digital Fundamentals. 5th ed. New York: MacMillan Publishing Company, 1994.
5. **Floyd, T.L.** Principles of Electric Circuits. 3rd ed. Columbus: Merrill Publishing Company, 1989.
6. **Grimbleby, J.B.** Computer-aided Analysis and Design of Electronic Networks. United Kingdom: Pitman Publishing, 1990.
7. **Malvino, A.P.** Electronic Principles. 3rd ed. United States of America: McGraw-Hill, Inc., 1984.

8. O'Kelly, D. Performance and Control of Electrical Machines. London: McGraw-Hill, 1991.
9. Raven, F.H. Automatic Control Engineering. 3rd ed. Tokyo: McGraw-Hill, 1978.
10. Sen, P.C. Principles of Electric Machines and Power Electronics. New York: John Wiley & Sons, 1989.
11. Swan, T. Mastering Turbo Pascal 6. 4th ed. United States of America: Hayden Books, 1991.
12. Yester, M. Using Turbo Pascal 6. 2nd ed. United States of America: Que Corporation, 1991.