



**DIE SKEP EN VERGELYKING VAN GEOMETRIESE LÊERS VAN
DRIEDIMENSIONELE MODELLE**

HENDRIK JOHANNES VAN STADEN

Verhandeling voorgelê ter voldoening aan die vereistes vir die graad

**MAGISTER TECHNOLOGIAE:
INGENIEURSWESE: ELEKTRIES**

in die

Departement Elektriese Ingenieurswese
Fakulteit Ingenieurswese

aan die

Technikon Vrystaat

Studieleier: Mnr. P.S. Veldtsman, M. Tech..(Ing)
Medestudieleier: Mnr. P.E. Hertzog, M. Tech..(Ing)

BLOEMFONTEIN
Desember 2000

BEDANKINGS

Ek wil graag die volgende persone en instansies bedank vir hul hulp en ondersteuning ter voltooiing van die projek:

Die NRF vir die befondsing, wat die studie moontlik gemaak het.

Die studieleier, mnr. P.S. Veldtsman, vir sy bystand, geduld en vriendskap tydens die studie en voltooiing van die projek, asook die medestudieleier mnr. P.E. Hertzog, vir sy belangstelling en bereidwilligheid om te help.

Mnr. D.J. de Beer, vir die vertroue wat hy in my gestel het, deur die projek aan te moedig, asook Prof G.D. Jordaan vir sy belangstelling.

Dr. H.S. Wolfaard vir die taalversorging van die skripsie.

Ludrich, Gerrie, Jacques, Manny, John, Minah en Dries in die meganiese laboratoriums vir hulle hulp met die meganiese aspekte van die projek.

My ouers, familie en vriende vir hulle vriendskap en bemoediging.

Soli Deo Gloria

Daar bestaan 'n behoefte om verskille te identifiseer tussen driedimensionele voorwerpe wat op die oog af identies is. Die vermoë om sulke vergelykings te kan tref sal dit moontlik maak om slytasie op voorwerpe te identifiseer asook om te bepaal hoe akkuraat 'n voorwerp vervaardig is, hetsy dit by wyse van 'n gietvorm of 'n snelprototiperingsproses geskied het.

Die probleem met die vergelyking van driedimensionele data is om die datapunte van die twee modelle op dieselfde relatiewe posisie op die modelle af te neem, sodat die regte geometriese koördinaat-ekwivalente met mekaar vergelyk kan word. Dit is waarom vergelyking van geometriese data, in die vorm van 'n neutrale-lêerformaat, baie gekompliseerd kan wees.

Deur die modelle vooraf met mekaar te oriënteer en daarna data-aftasting te doen, kan die twee stelle data in 'n geskikte formaat gestoor word sodat dit vergelyk kan word. Die oriëntasie word deur die gebruiker gedoen met behulp van oriëntasiehulpmiddels ingesluit in die rekenaarsagteware wat gedurende die navorsing geskryf is. Die sagteware beheer dan twee aftasmasjiene om die data op te neem en vergelyk dan die data, sodat verkille grafies en statisties vertoon kan word.

There is a need to identify differences between three-dimensional objects that seem identical to the naked eye. The ability to compare three-dimensional objects will enable one to determine wear that occurred on mechanical parts as well as to determine the accuracy by which a copy was fabricated, whether it was done by means of a mould or by a rapid prototyping process.

A problem encountered when comparing three-dimensional data is, to digitize the data at the exact same relative points on both models in order to enable comparison of all the data points with their counterparts. This is the reason why comparison of geometric data, generated by existing scanning systems and saved in a neutral file format, is bound to be complicated.

By orientating the models with one another prior to scanning, data recorded on the same relative positions can be compared. Orientation is done by the user with the aid of orientation tools included in the software that was written. This software controls two scanning machines to record the data in order to display differences graphically and statistically.

HOOFSTUK 1

INLEIDING

1.1	Oorsig	1
1.2	Probleemstelling en -oplossing	2
1.3	Doel van die studie	2
1.4	Hipotese	3
1.5	Belangrikheid van die projek	3
1.6	Navorsingsmetode	4
	1.6.1 Ontwerp van vergelykingstegnieke	4
	1.6.2 Ontwerp van 'n elektroniese koppelvlak	5
	1.6.3 Aanpassing van die aftasmasjiene	6
	1.6.4 Ontwerp van voelpenne vir data-opname	6
	1.6.5 Skryf van die rekenaarsagteware	6
	1.6.6 Foutsproing van sagteware en evaluering van die stelsel	7
1.7	Uniekheid van die projek	7
1.8	Opsomming	7

HOOFSTUK 2

NUMERIES BEHEERDE EN FREESMASJIENE

2.1	NC, CNC en CMM in die algemeen	9
2.2	Die Renishaw® Cyclone-aftasmasjien	14

2.3	Die Max NC 10-aftasmasjien	16
2.4	Ander masjiene	19
2.4	Opsomming	20

HOOFSTUK 3

DRIEDIMENSIONELE SKANDERING EN VERGELYKING

3.1	Voelpenne	21
3.1.1	Infrarooi sensors	26
3.1.2	Hall-effeksensors	28
3.1.3	Rekstrokies	30
3.1.4	Nabyheidsensors	32
3.2	Voelpenroetes vir data-opname	34
3.3	Die vergelyking van driedimensionele voorwerpe	36
3.4	Die stapberekeningslogika tydens aftasting	45
3.4.1	Lynaftasting	46
3.4.2	Omtrekaftasting	47
3.5	Opsomming	50

HOOFSTUK 4

GEOMETRIESE LÊERFORMATE

4.1	Inleiding	51
4.2	Neutrale lêertipes	52
4.2.1	Die AutoCAD <i>.dxf</i> - lêerformaat	53

4.2.2	Die <i>.iges</i> -lêerformaat	54
4.2.3	Die <i>.stl</i> -lêerformaat	55
4.3	Spesiale lêertipes vir afgetaste data	56
4.3.1	Die <i>.pos</i> -lêerformaat	56
4.3.2	Die <i>.hvs</i> -lêerformaat	59
4.3.3	Die <i>.set</i> -lêerformaat	60
4.3.4	Die <i>.cal</i> -lêerformaat	60
4.4	Opsomming	61

HOOFSTUK 5

DIE ELEKTRONIESE KOPPELVLAK

5.1	Trappiesmotore	62
5.2	Microchip® PIC-mikrobeheerders	65
5.3	Die parallelpoort	67
5.4	Die koppelvlakborduitleg	69
5.5	Beheerlogika	72
5.6	Mikrobeheerdersagteware	75
5.6.1	Motorbeheersagteware (PIC 1 en 2)	77
5.6.2	Beheersagteware vir Voelpen- en sensorterugvoer (PIC 3)	79
5.7	Opsomming	81

HOOFSTUK 6

REKENAARSAGTEWARE

6.1	Installering van die sagteware	82
6.2	Inisiasie	83
6.3	Die menu	84
6.3.1	“Options”	84
6.3.2	“Jog”	88
6.3.3	“Copy”	89
6.3.4	“Scan”	90
6.4	Die skandering en vergelyking van modelle in Reverse3D	91
6.5	Opsomming	97

HOOFSTUK 7

REKENAARPROGRAMMATUUR

7.1	Visual Basic 6 en Borland C++ 5	98
7.1.1	Borland C++ 5	99
7.1.2	Visual Basic 6	101
7.2	Module-uitleg van die programmatuur	104
7.3	Roetines in <i>.dll</i> -lêers	112
7.4	Opsomming	114

HOOFSTUK 8

STELSELEVALUASIE

8.1	Eksperiment A: Metingsakkuraatheid	115
8.2	Eksperiment B: Rotasie-akkuraatheid	116
8.3	Eksperiment C: Volumeberekeninge	117
8.4	Eksperiment D: Vergelyking van modelle	119
8.5	Eksperiment E: Ewekansige steekproefvergelyking	120
8.6	Akkuraatheidsdegradering	122

HOOFSTUK 9

SAMEVATTING

	Verdere navorsingsvoorstel	127
--	----------------------------	-----

BYLAAG A:	Vloeiogramme van Visual Basic-sagteware	128
------------------	---	-----

BYLAAG B:	Die Rotate-roetine se programlys	135
------------------	----------------------------------	-----

BYLAAG C:	Die InCircDraw-roetine se programlys	139
------------------	--------------------------------------	-----

BYLAAG D:	Skematiese diagram van die koppel-elektronika	140
------------------	---	-----

LITERATUURLYS		141
----------------------	--	-----

Lys van tabelle, figure en lystings

Figuur 1.1	Vloeiagram van die navorsingsmetode	5
Figuur 2.1.	'n NC-program en ponslint	10
Figuur 2.2	Die asse van die Cartesiese vlak	11
Figuur 2.3	Tipiese CNC-masjien	11
Figuur 2.4	Koördinaatterugvoering by as-motors	12
Figuur 2.5	Die Renishaw® Cyclone CMM	15
Figuur 2.6	Glybaan-skroefaandrywing by die Max NC CMM	16
Figuur 2.7	Die Max NC 10-masjien en -sagteware	19
Figuur 3.1	Voelpenne vir aftasting van driedimensionele voorwerpe	22
Figuur 3.2	Die foute by voelpenne a.g.v. fisiese dimensies	23
Figuur 3.3	Die implimentering van hefboomvoelpenne	24
Figuur 3.4	Die beginsel van driedimensionele voelpenne	25
Figuur 3.5	Die implimentering van infrarooi sensors op 'n voelpen	28
Figuur 3.6	Die beginsel van die Hall-effek	29
Figuur 3.7	Hall-effeksensors gebruik vir driedimensionele waarneming	29
Figuur 3.8	Rekstrokies gebruik as voelpensensors	31
Figuur 3.9	Nabyheidsensors	33
Figuur 3.10	Algemene voelpenroetes	35
Figuur 3.11	Spiraaltroete vir data-aftasting	36



Figuur 3.12 Die vergelyking v	rikse	38
Figuur 3.13 Datapassing rondom die datamiddelpunt-area		39
Figuur 3.14 Koördinaatberekennings vir datarotasië		44
Figuur 3.15 Die roteer-as gemonteer op die Max NC 10 CMM		45
Figuur 3.16 Omtrekaftasting van 'n model		48
Figuur 4.1 Voordele van neutrale-lêerformate		52
Figuur 5.1 Voorstelling van trappiesmotore		63
Figuur 5.2 Wringkragafname by hoë stapfrekwensies		64
Figuur 5.3 Rekenaar gegenereerde trappiesmotorseine soos verkry vanaf 'n logiese analiseerder		65
Figuur 5.4 Die pendigram van die PIC16F877		66
Figuur 5.5 Blokdiagram van die elektroniese koppelvlak		70
Figuur 5.6 Vloedigramme van die twee "Main"-roetines van die mikrobeheerdersagteware		76
Figuur 6.1 Die "Options"-menu uitgerol en resolusiebeheerraampie		85
Figuur 6.2 Traagheidskompensasië vir die trappiesmotore		86
Figuur 6.3 Die beheerraampies betrokke by kalibrasië van die trappiesmotore		87
Figuur 6.4 Die "Jog"-beheerraampie		89
Figuur 6.5 Die skakelingshiërargië van beheerraampies onder die "Scan"-menu-item		91
Figuur 6.6 Die "Positioning"-beheerraampie		91

Figuur 6.7	Omtreksilhoeët van modelle soos vertoon word op die rekenaarskerm	93
Figuur 6.8	Die beelde van die twee modelle om posisionele gelykheid te bevestig	94
Figuur 6.9	Die “Scan Options”-beheerraampie met beeld van afgetaste modelle	96
Figuur 6.10	Die vergelyking van modeldata	97
Figuur 7.1	Die C++ weergawe van Reverse3D	100
Figuur 7.2	Die Visual Basic-IDE	104
Figuur 8.1	Berekening van volume	118
Figuur 8.2	Demensies van die modelle wat vergelyk is	119
Figuur 8.3	Groot koördinaatverskille op vertikale modelrande	122
Figuur 8.4	Foute by die glybaanbewegingsmetode	123
Figuur 8.5	Foutiewe lesings by loodregte modelrande	125
Tabel 3.1	Lusvoorkomende stappe by omtrekaftastng	50
Tabel 5.1	Die beheerseine van die standaard-parallelpoort	67
Tabel 5.2	Die pen-uitleg van die D-tipe 25-penverbinders tussen die elektroniese koppelvlak, die Max NC-masjien en die rekenaar	71
Tabel 5.3	Die opsie-registerkode gekoppel aan die mikrobeheerderreaksie	72
Tabel 5.4	Die status-registerkodering	74
Tabel 5.5	Uiteensetting van sensorvertolking op die status-register	75

Tabel 8.1	Akkurater metings d.m.v. kalibrasie	116
Tabel 8.2	Resultate van Eksperiment 3	120
Tabel 8.3	Datapunte van ewekansige steekproef	121
Kodelys 4.1	'n <i>.stl</i> -lêer	56
Kodelys 5.1	Een van 29 roetines wat die trappiesmotor- bewegingsrigtings beheer	77
Kodelys 5.2	Een van twee soortgelyke roetines wat trappiesmotorkodes genereer	78
Kodelys 5.3	Roetines betrokke by voelpenkodering	80

Lys van Akronieme

3D	Driedimensioneel
ASCII	“American Standard Code for Information Interchange”
API	“Applications Programming Interface”
BIOS	“Basic Input Output System”
CAD	“Computer Aided Design”
CMM	“Coordinate Measuring Machine”
CNC	“Computer Numeric Control”
DLL	“Dynamic Link Library”
DXF	“AutoCad data exchange format”
EEPROM	“Electrically Erasable Programmable Read Only Memory”
GS	Geïntegreerde stroombaan
GUI	“Graphical User Interface”
NC	“Numeric Control”
IDE	“Integrated Development Environment”
I/U	In- of uitsette
PIC	“Programable Intergrated Circuit”
SPP	“Standard parallel port”

Anti-Kopieerprop	“Dongle”
Aftasting	“Scanning”
Kieslys of Menu	“Menu”
Beheerraampie	“Dialog Box”
Bis	“Bit”
Biskaart	“Bitmap”
Drukknop	“Push button”
Gemene-beheerraampie	“Common Dialog Box”
Gids	“Directory”
Muisklik	“Mouse Click”
Nabyheidsensors	“Proximity detectors”
Bedryfstelsel	“Operating System”
Sensorstand	Kode wat die defleksierigting van voelpenne reflekteer
Skermwyser	“Mouse pointer”
Nutsbalk	“Toolbar”
Selflaai	“Bootstrap”
Voelpen	“Probe”

INLEIDING

1.1 Oorsig

In praktyk kyk ons na twee voorwerpe wat veronderstel is om dieselfde te wees en aanvaar dat hulle fisiese afmetings identies is. Wanneer sulke “identiese” modelle egter met mekaar vergelyk sou word, sal waargeneem word dat dit nie die geval is nie. Hoe meer gevorderd die metingsapparaat is wat gebruik word, hoe gouer sal dit duidelik word dat bykans geen mensgemaakte voorwerpe identies is nie.

In die meganiese ingenieurswese is akkuraatheid op skale onsigbaar vir die blote oog soms noodsaaklik. Dit is dus belangrik om ‘n metode daar te stel wat dit moontlik sal maak om te bepaal wat die afwykings tussen eenderse voorwerpe is, waar baie fyn afwykings ter sprake is.

In die meganiese ingenieurswese en spesifiek in die velde van snelprototipering en tru-ingenieurswese is daar ‘n leemte vir ‘n stelsel wat kan bepaal hoe akkuraat ‘n kopié of model vervaardig is. Die vermoë om slytasie op meganiese onderdele te bepaal kan baie velde van ingenieurswese baat, sodat voorkomende maatreëls tydens die ontwerpfase

getref kan word. Die voorkomende maatreëls kan insluit die kompensasië vir inkrimping van 'n materiaal met afkoeling na 'n gietproses, of die afwykings wat gepaard gaan met tolgietingsprosesse.

1.2 Probleemstelling en oplossing

Daar bestaan 'n behoefte om die geometriese data van driedimensionele voorwerpe te vergelyk. Die formaat waarmee data in neutrale lêerformate (aanvaarde standaard lêerformate bv, *.dxf*, *.iges* en *.stl*) gestoor word, kompliseer egter datavergelyking (bespreek in Hoofstuk 4). Die relatiewe posisies van die modelle tydens aftasting skep verder probleme by datavergelyking.

Indien sagteware geskryf word wat twee driedimensionele aftasmasjiene beheer, sodat data in 'n spesifieke formaat opgeneem kan word, kan bepaal word wat die afwyking tussen twee modelle is.

1.3 Doel van die studie

Hierdie studie is daarmee gemoeid om 'n stelsel daar te stel wat geometriese data van eenderse voorwerpe opneem met die doel om dit met mekaar te vergelyk, sodat die verskille statisties en grafies vertoon kan word. Dit sal verwesenlik word deur twee driedimensionele aftasmasjiene

deur middel van 'n rekenaarprogram (geskryf in Visual Basic) via 'n elektroniese koppelvlak te beheer, sodat die data verwerk kan word deur die rekenarsagteware.

1.4 Hipotese

Geometriese data van driedimensionele voorwerpe kan met mekaar vergelyk word, indien die data deur 'n spesifieke metode opgeneem en gestoor word.

1.5 Belangrikheid van die projek

Die vermoë om te bepaal hoeveel modelle met mekaar, of met spesifikasies verskil, sal die meganiese ingenieur in staat stel om tydens die ontwerp van 'n produk te kompenseer vir afwykings in modelafmetings wat voorkom as gevolg van die gietprosesse en verskeie ander faktore. Waar massaproduksie van 'n sekere model plaasvind, kan daar periodies getoets word of die vervaardigde produkte binne die spesifikasies val, deur dit met 'n meester te vergelyk.

Die bepaling van slytasie op onderdele kan ook gebruik word om tydens die ontwerpfase voorkomende stappe te neem, om sodoende slytasie en

meganiese faling tot 'n minimum te beperk. Dit kan ook gebruik word om te bepaal wanneer 'n onderdeel weens slytasie vervang moet word.

1.6 Metode van navorsing

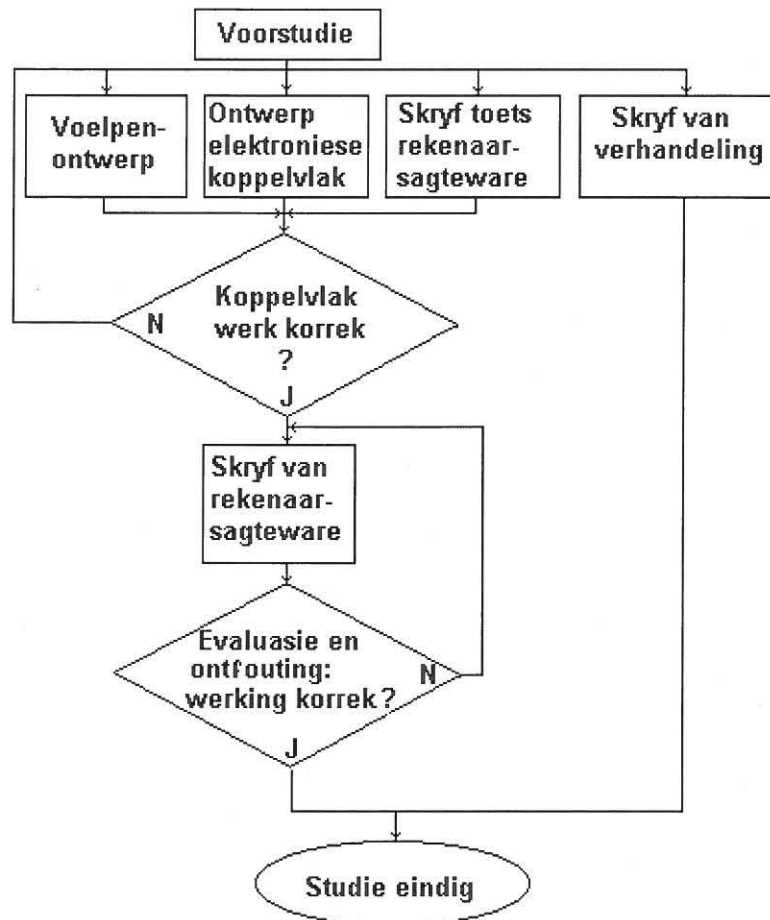
Die navorsing is in die volgende fases uitgevoer:

- Ontwerp van vergelykingstegnieke
- Ontwerp van 'n elektroniese koppelvlak
- Aanpassing van die aftasmasjiene vir data-opname
- Ontwerp van voelpenne vir data-opname
- Skryf van die rekenaarsagteware
- Foutsporing van sagteware en evaluering van die stelsel

Elke fase word in die volgende paragrawe kortliks bespreek. Figuur 1.1 is 'n vloeiagram wat die verloop van die navorsing deur die verskillende fases wat bespreek word, illustreer.

1.6.1 Ontwerp van vergelykingstegnieke

Die eerste taak was om die beste metode te vind om die data op te neem en te stoor. Metodes om die modelle te oriënteer moes gevind word, sodat die relatiewe datapunte in dieselfde relatiewe posisies op die twee modelle wat vergelyk word, opgeneem kon word.



Figuur 1.1 Vloeiagram van die navorsingsmetode

1.6.2 Ontwerp van 'n elektroniese koppelvlak

As gevolg van 'n beperkte aantal I/U penne is dit onmoontlik om twee masjiene direk vanaf 'n enkel parallelpoort te beheer. Daarom moes daar 'n elektroniese koppelvlak ontwerp word wat instruksies vanaf die rekenaar vertolk in die nodige masjienebeheerseine, asook om die voelpendata terug te lei na die rekenaar.

1.6.3 Aanpassing van aftasmasjiene

Daar moes sekere veranderings aan die masjiene aangebring word om hulle geskik te maak vir die doel waarvoor hulle aangewend is. Onder andere moes limietsensors op die asse aangebring word, asook 'n bykomende motor vir modelrotasie geïnstalleer word.

1.6.4 Ontwerp van voelpenne vir data-opname

Navorsing is gedoen om 'n baie sensitiewe voelpen te produseer sodat hoogs akkurate metings moontlik gemaak kon word. Daar is met verskillende soorte sensors geëksperimenteer om tas waar te neem, asook meganiese modelle waarop die sensors geïnstalleer kan word.

1.6.5 Skryf van rekenaarsagteware

Rekenaarsagteware is geskryf in Visual Basic om die model-aftasting en -oriëntasie te beheer, asook om die data te vergelyk. Die skryf van rekenaarsagteware neem baie tyd in beslag, omdat daar soveel oorhoofse take betrokke is, wat in wese niks met die spesifieke toepassing te doene het nie.

1.6.6 Foutsporing van sagteware en evaluering van die stelsel

Foutsporing van enige sagteware is 'n voortdurende proses, wat gedoen word soos probleme aan die lig kom. Die stelsel is geëvalueer om te bepaal hoe akkuraat vergelykings getref word.

1.7 Uniekheid van projek

Daar is baie masjiene en sagtewarepakette beskikbaar wat spesifiek ontwerp is vir data-aftasting, sodat die data deur ander CAD-sagteware gemanipuleer kan word. Geen sagteware is egter beskikbaar wat geometriese data met mekaar kan vergelyk nie.

Dit was dus nodig om navorsing in hierdie rigting te begin doen, omdat daar vir so 'n stelsel baie toepassings bestaan.

1.8 Opsomming

In Hoofstuk 2 word inleidende inligting oor verskillende numeriese beheerde masjiene en driedimensionele-aftasmasjiene gegee. Hoofstuk 3 handel oor die navorsing wat gedoen is oor driedimensionele aftasting en die vergelyking van driedimensionele voorwerpe. Hoofstuk 4 behels 'n

beskywing oor sommige algemene lêerformate, wat gebruik word vir die stoor van geometriese data, asook die lêerformate wat spesifiek vir die navorsing ontwerp is. In Hoofstuk 5 word die elektroniese koppelvlak en die logika daaragter verduidelik. Hoofstuk 6 sit die rekenaarsagteware uiteen, terwyl Hoofstuk 7 handel oor die programmatuur daaragter. In Hoofstuk 8 word die resultate wat verkry is, bespreek.

HOOFSTUK 2

NUMERIES BEHEERDE EN FREESMASJIENE

In hierdie hoofstuk word 'n oorsig gegee van sommige stelsels beskikbaar op die mark. Die tendens van die betrokke masjiene in die industrie word voorts bespreek. Inligting oor die masjiene wat vir die navorsing gebruik is, word ook verkaf om as grondslag te dien vir latere besprekings.

2.1 NC, CNC en CMM in die algemeen

Wanneer aan die meganiese ingenieur die taak opgelê word om 'n werkstuk uit 'n materiaal, bv. metaal of plastiek, te masjineer, maak hy van 'n freesmasjien gebruik. 'n Freemasjien is 'n gespesialiseerde boormasjien, wat sowel horisontaal as vertikaal kan boor of sny. Wanneer so 'n masjien automaties, in plaas van deur 'n permanente operateur, beheer word, word dit 'n numeries beheerde masjien (NC-masjien) genoem. Die data wat die motorbewegings spesifiseer, ten einde boorbewegings te bewerkstellig is vroeër op ponslint (Figuur 2.1b) gestoor. Met die koms van die persoonlike rekenaar kon die data deur alternatiewe metodes gestoor word en vanuit die rekenargeheue na die masjien versend word; vandaar die moderne akroniem CNC ("Computer Numerical Control") [12, p.118]. 'n NC-program bestaan uit 'n stel

instruksies wat die aksies beskryf wat die masjien moet uitvoer. Die instruksies (“Part Program”) word in blokke gerangskik en bestaan onder andere uit die G-kodes (voorbereidingsinstruksies), M-kodes (hulp-instruksies), T-kodes (gereedskapsinstruksies), F-kodes (spoedbeheer) en S-kodes (spilspoedbeheer) [23, pp. 11-31]. In Figuur 2.1a word ‘n voorbeeld van ‘n NC-program verskaf.

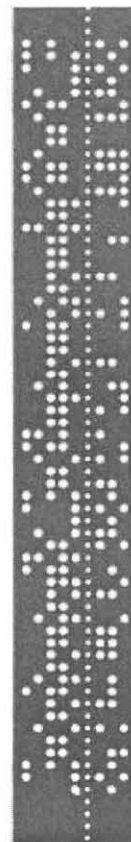
Metric Machining Center Example Program

```

N0010 G90      Absolute programming
N0020 G71      Metric programming
N0030 G94      Millimeter per minute feedrate
N0040 G17      X-Y circular interpolation plane
N0050 G40 T00  Cancel cutter diameter compensation and tool length compensation
N0060 G80 Z25.0 Retract Z axis
N0070 G00 X-50.0 Y50.0 T01 M06 Tool change "25.0 endmill"
N0080 X-12.5 Y12.5 Z2.5 S400 M03 Rapid to position 1, start spindle
N0090 Z-27.5 M08 Rapid to Z depth turn coolant on
N0100 G01 Y-95.0 F284 Feed to position 2
N0110 X151.68 Feed to position 3
N0120 X184.0 Y-62.68 Feed to position 4
N0130 Y12.5 Feed to position 5
N0140 X115.5 Feed to position 6
N0150 Y0 Feed to position 7
N0160 G02 X90.0 Y-25.5 I25.5 J0 Circular interpolate to position 8
N0170 X64.5 Y0 I0 J25.5 Circular interpolate to position 9
N0180 G01 Y12.5 Feed to position 10
N0190 X-12.5 Feed to position 11
N0200 G00 Z2.5 M09 Clear part
N0210 G80 X-50.0 Y50.0 Z63.0 T02 M06 Tool change "6.0 dia. drill"
N0220 X13.0 Y-13.0 Z2.5 S1600 M03 Rapid to position 12, turn spindle on
N0230 G81 Z-29.3 R2.5 F244 M08 Drill position 12, turn coolant on
N0240 Y-63.5 Drill position 13
N0250 X152.0 Y-19.0 Drill position 14, turn coolant off
N0260 G80 X-50.0 Y50.0 Z89.0 T03 M06 Tool change "24.0 dia drill"
N0270 X90.0 Y-57.5 Z2.5 S425 M03 Rapid to position 15, turn spindle on
N0280 G81 Z-34.7 R2.5 F216 M08 Drill position 15, turn coolant on
N0290 M09 Turn coolant off
N0300 G80 X-50.0 Y50.0 Z25.0 T04 M06 Tool change, "25.0 dia boring bar"
N0310 X90.0 Y-57.5 Z2.5 S1200 M03 Rapid to position 16, turn spindle on
N0320 G81 Z-27.5 R2.5 F183 M08 Bore position 16, turn coolant on
N0330 M09 Turn coolant off
N0340 G80 X-50.0 Y50.0 Z51.0 M02 Rapid to tool change position and end program

```

Punched tape

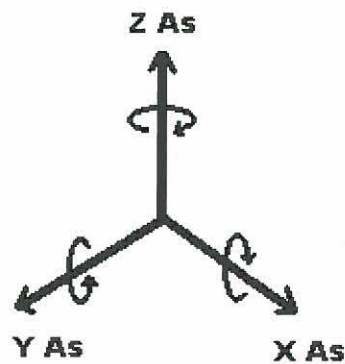


a

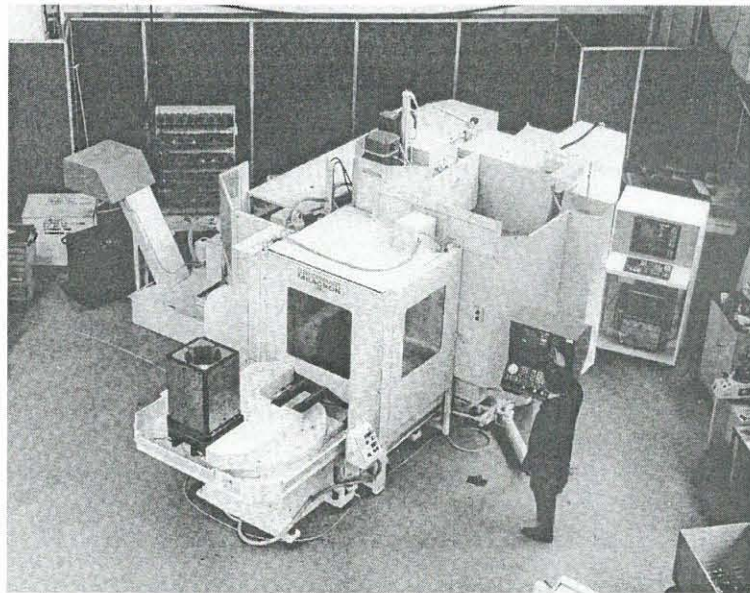
b

Figuur 2.1 ‘n NC-program en ponslint

'n CNC-masjien bestaan gewoonlik uit drie motore reghoekig tot mekaar gerig, sodat die beitelpunt al die moontlike posisies binne die perke van die Cartesiese vlak kan bereik [31, p. 91]. Addisionele motore kan bygevoeg word, wanneer dit nodig word vir roterende sye [12, p. 13][23, p. 3]. Figuur 2.2 wys die Cartesiese vlakke met rotasie, terwyl Figuur 2.3 'n tipiese CNC-masjien toon.

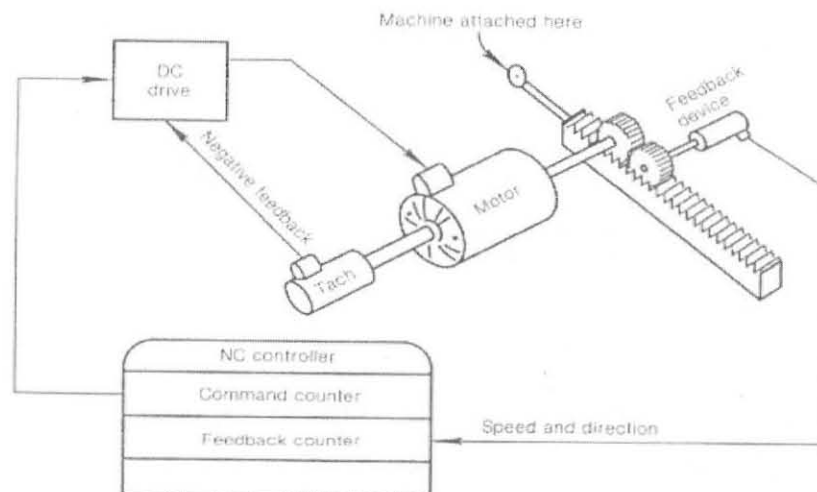


Figuur 2.2 Die asse van die Cartesiese vlak



Figuur 2.3 Tipiese CNC-masjien

In sommige gevalle word beweging van die onderskeie asse gemonitor deur middel van 'n terugvoersensor. Figuur 2.4 toon 'n terugvoersensor tussen die motorbeheer en die motor, asook een op die bewegingsas vir koördinaatbepaling. Dit verseker dat die bewegings wel plaasgevind het en word ook gebruik om die koördinaatposisies te bepaal. Sodoende kan foute reggestel word wanneer dit plaasvind.



Figuur 2.4 Koördinaat-terugvoering by as-motors

Nog 'n algemene term by CNC-masjiene is die "aanvangspunt". Dit is vanaf hierdie punt dat die rekenaar alle koördinate bereken en word bepaal deur sensors geleë op die punte van die X-, Y-, en Z-asse [12, p. 5][23, p. 148][31, p. 157]. Normaalweg sal die aanvangspunt van 'n sekere as bepaal word deur die betrokke motor in die rigting van die sensor te beweeg, totdat die sensor bepaalde terugvoering gee.

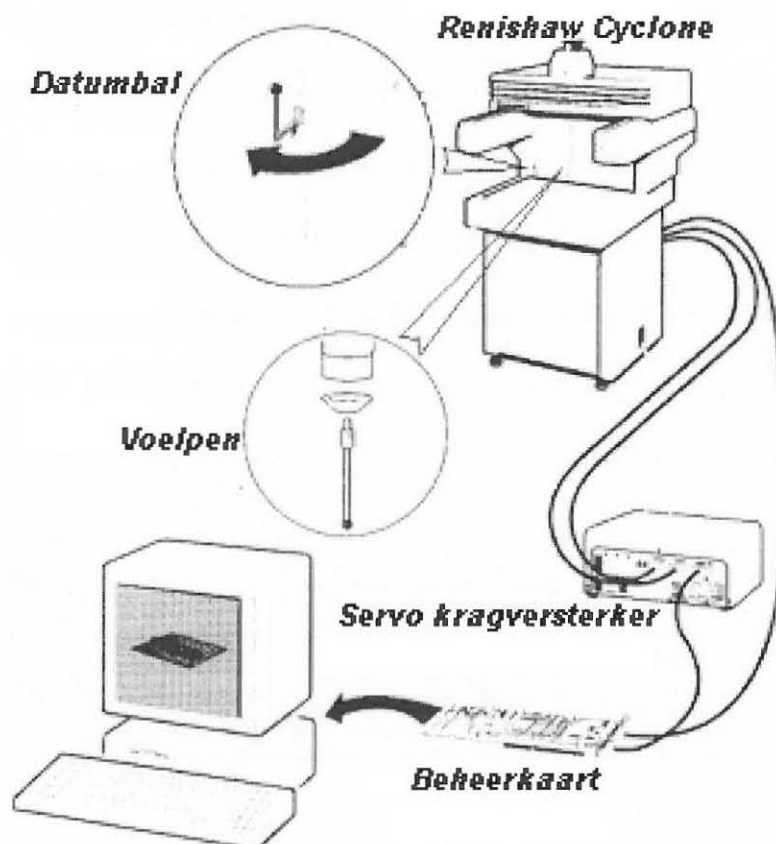
‘n Driedimensionele-aftasmasjien (CMM) verskil egter van die CNC-masjien in dié opsig dat ‘n voelpen die boor- of beitelpunt vervang. ‘n Voelpen is ‘n sensor wat terugvoering gee, indien sy punt in aanraking met ‘n voorwerp kom [31, p.199]. Voelpenne word in Hoofstuk 3 breedvoerig beskryf.

Nog ‘n verskil tussen CMM- en CNC-masjiene is dat ‘n CMM-masjien ‘n tasroete volg wat nie *a priori* gespesifiseer word nie en dus onbekend is tydens dataversyfering. Die CNC-masjien voer sy bewegings uit volgens sy programinstruksies. Wanneer die modeloppervlak versyfer word, bestaan daar ‘n paar moontlike raamwerke waarvolgens data opgeneem kan word, maar wanneer die omtrek van ‘n model versyfer word, berus dit geheel en al op die sagteware om die nodige besluite te neem. Om hierdie rede neem die versyferingsproses soveel langer as masjinerie. In dié studie is daar gekyk na CMM-masjiene, na gelang van hul beskikbaarheid met die oog op modelversyfering vir datavergelyking. Hier volg ‘n kort beskrywing van CMM’s wat ondersoek is.

2.2 Die Renishaw® Cyclone-aftasmasjien

Die Renishaw® Cyclone is 'n versyferingstelsel spesifiek gerig tot data-opname van twee- en driedimensionele voorwerpe [25, p. 49]. Dit beskik oor drie motore, reghoekig tot mekaar gerig, met ingeboude tagometers op elkeen vir koördinaatterugvoering. Lugdrukreguleerders en -filtreerders voorsien die luglaers (waarop die drie asse voer) van die nodige lugdruk. Die gebruik van luglaers met 'n lae inersie maak akkurate skandering van voorwerpe teen hoë spoed moontlik. Die stelsel se skandeerspoed word gespesifiseer as 3 m/min en sy maksimum spoed as 6 m/min, terwyl sy akkuraatheid 50 μm en sy resolusie 7 μm is [25, pp. 1-4]. Die Renishaw® Cyclone word voorsien van 'n kalibrasiesfeer (of datumsfeer), asook 'n Renishaw® SPSM-voelpe toegegerus met faalbeskerming. Die kalibrasie van die voelpe en as-aandrywing word volgens dié sfeer gedoen. Die beheerseine vanaf die Tracecut sagteware word via die aftasbeheerrekenaarkaart na die servo-kragversterker gestuur vir die beheer van sy drie as-motore, wat die voelpe oor die voorwerp beweeg. Die posisie van die asse, en gevolglik die voelpeposisie, word deur drie Renishaw® skaal- en leessensors teruggevoer, via die aftasbeheerbord. Die diagram van die Renishaw® Cyclone met die verskillende komponente wat bespreek is, word in Figuur 2.5 getoon.

Die Tracecut sagteware stel die ingenieur in staat om 'n voorwerp te skandeer en die data oor die oppervlakgeometrie in verskillende lêerformate te stoor. Die data kan verder gemanipuleer word in die vorm van 'n NC-program, sodat 'n kopie van die oorspronklike voorwerp op 'n NC-masjien gemasjineer kan word [25, pp. 1-2][2, p. 2]. Waar die data in 'n neutrale-lêerformaat omskep word, kan dit deur gespesialiseerde CAD-sagteware (bv Microstation®) gemanipuleer word en die veranderde data gebruik word om 'n gietvorm of 'n NC-program te genereer. Neutrale-lêerformate word in Afdeling 2.4.1 bespreek.

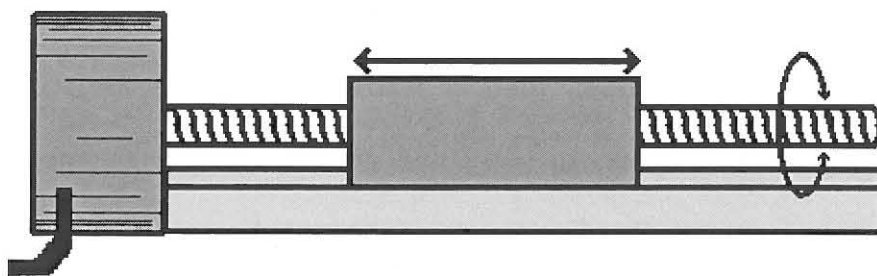


Figuur 2.5 Die Renishaw® Cyclone CMM

Die Renishaw® Cyclone kan egter nie gebruik word as 'n freemasjien nie, weens die luglaers wat op die asse gebruik word. Dit kan wel gedoen word met die Max NC masjien wat in die volgende afdeling bespreek word. Die Renishaw® Cyclone is ook relatief duur in vergelyking met die Max NC, wat navorsing daarmee bemoeilik.

2.3 Die Max NC 10 aftasmasjien

Soos die Renishaw® Cyclone, het die Max NC 10 ook drie asse, loodreg tot mekaar gerig, om driedimensionele beweging van 'n boor of voelpen te bewerkstellig. Die Max NC 10 is primêr 'n freemasjien, met die gevolg dat daar geen posisionele terugvoering op die asse teenwoordig is nie. Drie trappiesmotore is verantwoordelik vir die beheer van die posisies van die X-.Y- en Z- koördinate. Asbewegings word deur middel van glybaan-skroefaandrywing bewerkstellig soos geïllustreer in Figuur 2.6.



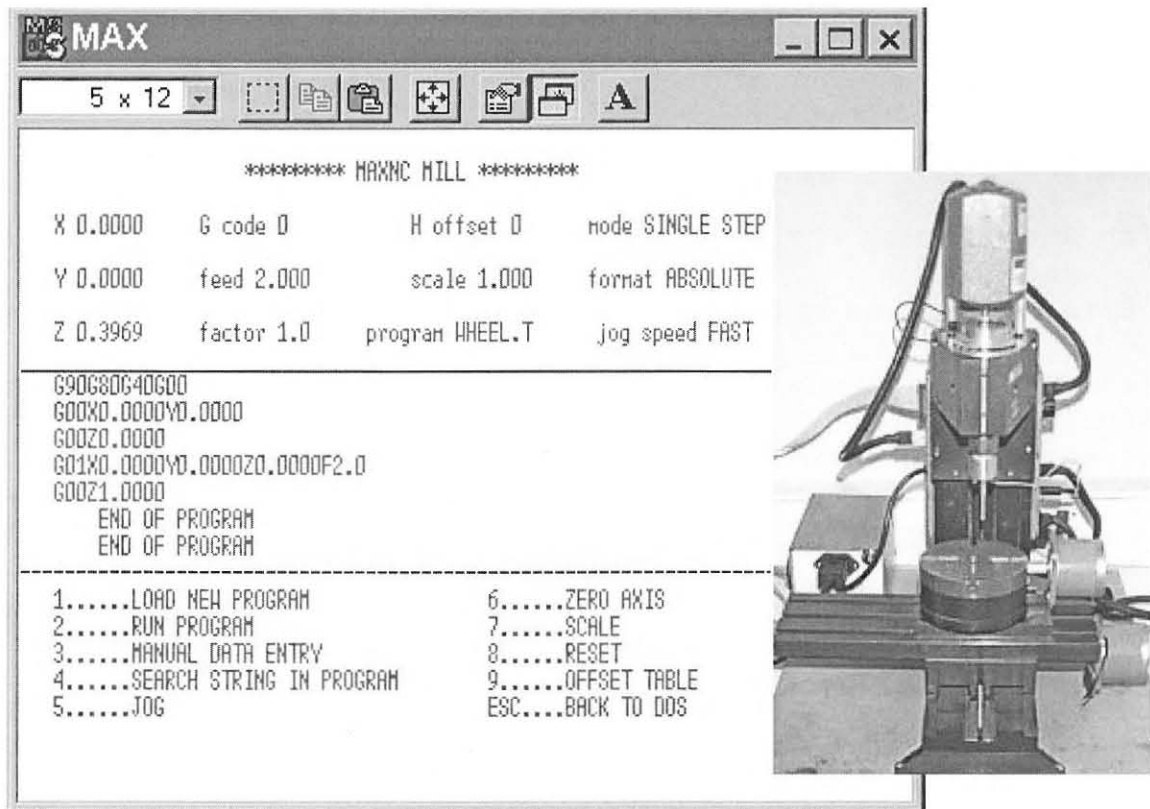
Figuur 2.6 Glybaan-skroefaandrywing by die Max NC-CMM

Die skroefdraad wat die Max NC 10 aandryf, het 'n 1.296mm draadmaat. Teen 1.8° /stap moet 200 stappe geneem word om een rotasie te voltooi. Die kleinste beweging wat dus op die Max NC moontlik is, is dan $1.296/200 = 6.48\mu\text{m}$. Dit is die kleinste moontlike resolusie waarteen aftasting op die Mac NC 10 CMM gedoen kan word. Hierdie inligting is belangrik vir koördinaatkodering tydens die versyferingsproses.

Die drie asse van die Max NC CMM is só gerangskik dat die voelpe self slegs in die rigting van die Z-as beweeg. Die model beweeg op die X- en Y-vlak onder die voelpe rond, in teenstelling met die Renishaw® Cyclone, waar die model glad nie beweeg nie. Dit is om dié rede dat die Renishaw® Cyclone oor 'n funksie beskik waardeur die voelpe geposisioneer kan word deur slegs die voelpe, wat driedimensioneel sensitief is, in 'n spesifieke rigting te trek. Die sagteware reageer op die terugvoering van die voelpe deur die motore te beweeg in 'n rigting waar die voelpe geen defleksie terugvoer nie. Hierdie funksie is onprakties by die Max NC 10, aangesien die voelpe slegs op en af beweeg, maar dit sou geïmplimenteer kon word. Verder is daar geen limietsensors by die Max NC 10 teenwoordig nie en dus kan geen aanvangspunte bepaal word nie. Die aanvangspunt word deur die gebruiker in die sagteware gespesifiseer en die bewegings deur 'n NC-program beheer. Die sagteware wat die Max NC 10 vergesel, is nie grafies-gebruiker-skakelend (GUI) nie, in teenstelling met die Tracecut 2.0-sagteware en is beperk tot slegs die

nodigste funksies om NC-programmering en modelmasjinerings toe te laat [17, p. 5]. Die voelers vir data-opname is slegs 'n skakelaar en verskaf dus nie inligting oor die tasrigting nie. Figuur 2.7 toon die Max Nc 10 masjien en sagteware.

Die Max NC is aansienlik goedkoper (in verhouding $\approx 1:20$) as die hoogs akkurate Renishaw® Cyclone. Daar kan ook geen vergelyking getref word tussen die sagteware wat die stelsels beheer nie. Die Max NC 10 kan egter gebruik word vir aftasting sowel as masjinerings van materiale, insluitende metaal. Die Max NC is nie die ideale opsie wat sy akkuraatheid aan betref nie [13, p. 1], maar dit het die navorsing moontlik gemaak. Die tekortkominge van die Max NC en faktore wat aftas-akkuraatheid benadeel, word in Hoofstuk 8 bespreek.



Figuur 2.7 Die Max NC 10 masjien en sagteware

2.4 Ander masjiene

Die Modela 3D-plotter, gekombineer met die DeskProto 1.0-sagteware, is 'n kompakte, bekostigbare freesmasjien, ideaal vir snelprototipering van klein modelle uit poreuse materiale. Die sagteware laat die masjinering van modelle vanaf *.stl*-, en sommige *.dxf*-lêersoorte toe, sonder dat 'n NC-program nodig is [8, p. 7]. Die sagteware laat die gebruiker verder toe om skaalveranderings te doen en bereken snyroetes vir verskillende tipes boorpunte. Dié sagteware laat nie aftasting toe nie, dus vergesel geen

voelpe die stelsel nie. Daar is egter wel CMM-masjiene onder dié handelsmerk beskikbaar.

Die EMCO PC Mill 100 is 'n CNC-freesmasjien wat gebruik word om groot voorwerpe uit materiale (insluitende metaal) te masjineer en word deur NC-programmatuur beheer [10, p. 3]. 'n Voelpe word soms ingesluit in die gereedskapsmagasyn van sulke masjiene om gatgroottes en -middelpunte te bepaal [2, p. 25]. Gedurende die snyproses kan bore of kerfbytels verruil word vir 'n voelpe om te toets of die snywerk aan die model binne die spesifikasies val [12, p. 410]. Soms word 'n voelpe ook gebruik om die toestand van die boor of bytel te bepaal. Beperkte beskikbaarheid van dié masjiene maak navorsing daarmee egter moeilik.

2.4 Opsomming

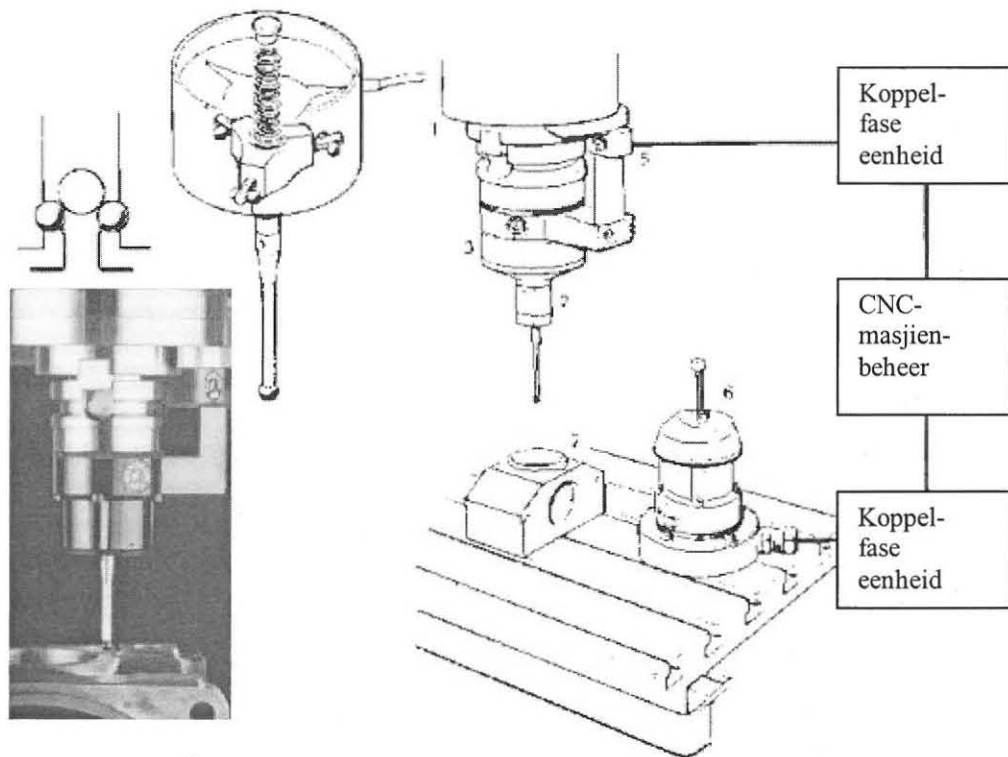
Dié hoofstuk het inleidende inligting verskaf oor verskillende hardeware wat te doene het met numeriese beheer en data-aftasting. Daar is vergelyking getref tussen twee moontlike masjiene wat vir gebruik in dié studie oorweeg is.

DRIEDIMENSIONELE SKANDERING EN VERGELYKING

In dié hoofstuk word die voelpenne vir die opneem van geometriese data van driedimensionele voorwerpe bespreek. Die vergelykingsmetodes en -tegnieke word verduidelik en die lêerformate en vereistes om data te vergelyk, word uiteengesit.

3.1 Voelpenne

Die voelpen is 'n baie belangrike komponent van aftasing, aangesien die akkuraatheid van die metings grotendeels daarvan afhang. Die primêre doel van 'n voelpen is om 'n sein te genereer wanneer dit met 'n voorwerp in aanraking kom. Die sein kan ook inligting bevat oor die rigting van die aanraking, wat so 'n voelpen dan "intelligent" maak. Kontakvoelpenne maak staat op die geleiding van elektrisiteit deur die penpunt en die voorwerp wat, uit die aard van die saak, geleidend en dus gewoonlik metaal sal moet wees. Tas-snellervoelpenne maak staat op meer gevorderde tegnologie en is nie beperk tot slegs metaalmodelle nie [31, p.112]. Figuur 3.1 toon 'n voorbeeld van 'n tas-snellervoelpen, soortgelyk aan die wat in dié projek gebruik is.

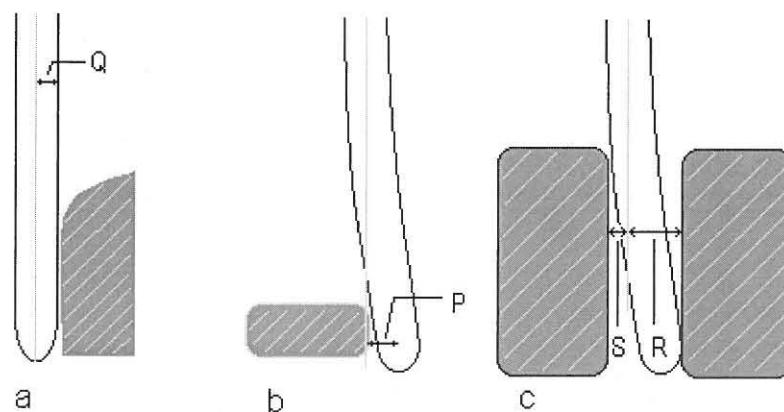


Figuur 3.1 Voelpenne vir aftasting van driedimensionele voorwerpe

Die defleksie op die voelpunt wat vereis word om 'n sein te genereer, bepaal die sensitiwiteit van die voelpen. Hierdie sensitiwiteit moet dus hoër wees as die motorresolusie. Indien hierdie verhouding nie reg is nie, word die akkuraatheid benadeel.

Die lengte en deursnee van die pen beïnvloed ook die akkuraatheid van aftasting. Die pendikte veroorsaak dat die lesing wat tydens voelpendefleksie geneem word, met die helfte van die pendeursnee afwyk, aangetoon as Q in Figuur 3.2a. Die afwykings veroorsaak dat die virtuele

modelvolume (die data gestoor in die lêer) in alle rigtings toeneem met 'n lengte gelyk aan die helfte van die pendikte. Daar moet dus in die sagteware gekompenseer word vir die fout, deurdat van die operateur verwag word om die korrekte pendikte te selekteer. Die ideale pendikte sou een wees wat gelyk is aan die aftasresolusie. In werklikheid sal 'n pen wat te dun is, by aanraking buig, sonder dat die sensor enige defleksie waarneem. Die verhouding van die lengte tot die dikte van die pen bepaal die onvermydelike defleksie wat op die pen plaasvind, voordat die basis van die pen genoegsaam gedeflekteer het om die sensor te aktiveer. Dit word in Figuur 3.2b geïllustreer, waar die afstand, P, die fout verteenwoordig.

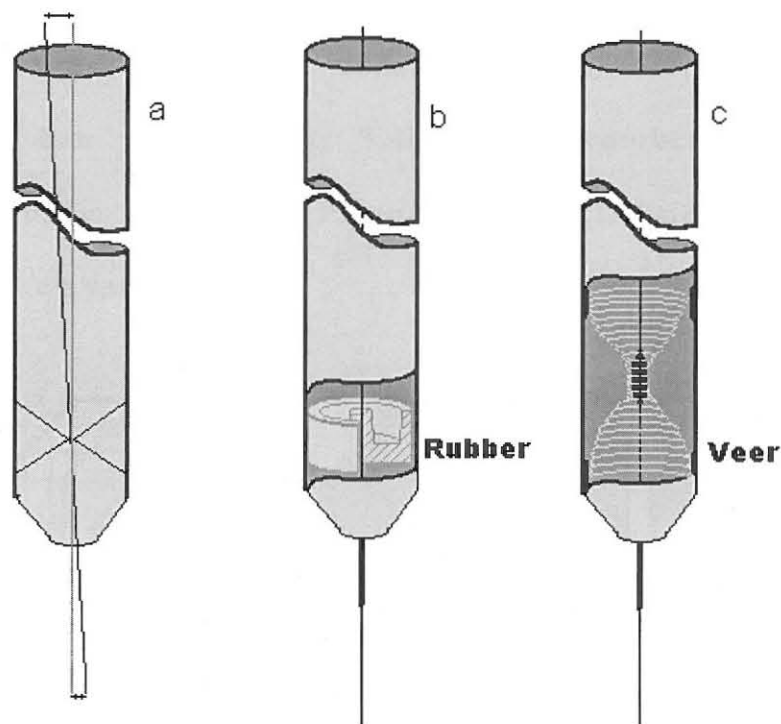


Figuur 3.2 Die foute by voelpenne a.g.v. dimensies

Hoe langer en dunner die voelpen, hoe groter hierdie defleksie. Die lengte van die pen bepaal die diepte in die Z-as wat gemeet kan word. Vir hoër akkuraatheid word die prys betaal van 'n klein tasdiepte. Daar moet dus

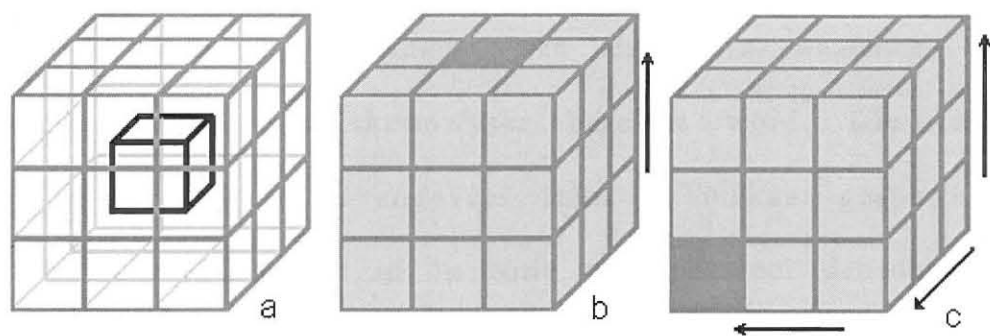
gesoek word na 'n goue middeweg tussen die voelpenlengte en -dikte, óf voelpenne moet geruil word, na gelang van die modeldimensies.

Voordat daar met aftasting begin word, moet die algemene toestand van die voelpen nagegaan word. Dit is baie belangrik dat die voelpen so reguit as moontlik is, veral wanneer die modelomtrek afgetas word. 'n Skewe voelpen veroorsaak foutiewe lesings op die X- en Y- asse, veral waar op loodregte rante aftasting gedoen word. Dit is om dié rede dat dit nodig is dat die pen volkome silindries moet wees. Figuur 3.2c toon die verskil in die horisontale afstande, R en S, op 'n loodregte modelrand as gevolg van 'n skewe of onsilindriese voelpen.



Figuur 3.3 Die implementering van hefboomvoelpeene

Nog 'n metode om die defleksie mee waar te neem is om van 'n hefboomstelsel gebruik te maak. Figuur 3.3a toon hoe die defleksie aan die bopunt van die hefboom veel groter is as op die onderpunt. Die sensors word aan die bokant, hierna verwys as die sensorkant, geplaas waar die defleksie maksimaal is. Sodoende word die sensitiwiteit verbeter. Om tussen driedimensionele tasrigtings te kan onderskei moet defleksie in die X-, Y-, en Z-riktings meetbaar wees. Daar moet dus ten minste drie sensors teenwoordig wees; een vir elke basiese bewegingsrigting. Figuur 3.4a toon 'n 3³-matriks, vergelykbaar aan 'n Rubik-kubus. Die onsigbare blokkie op die kubus (getoon in swart in Figuur 3.4a) wat grens aan al die sigbare blokkies stel die rusposisie (tydens geen defleksie op die voelpe nie) voor. Wanneer defleksie plaasvind, sal die sensorkant in een van die omliggende areas inbeweeg, sodat dit deur die betrokke sensor waargeneem kan word. Figuur 3.4b toon byvoorbeeld defleksie wat loodreg van onder was, terwyl Figuur 3.4c 'n defleksie toon wat van bo, die agterkant en van regs was.



Figuur 3.4 Die beginsel van driedimensionele voelpe

'n Probleem wat ondervind word om 'n voelpen te produseer, is om die meganiese beweeglikheid te bewerkstellig. Die voelpen moet driedimensioneel beweeglik wees en altyd terugkeer na sy geentas-posisie wanneer daar geen defleksie is nie. Figuur 3.3b stel die idee voor van 'n voelpen wat in 'n rubberhuls lê, om driedimensionele beweeglikheid te bewerkstellig. Figuur 3.3c maak vir dié doel van 'n veer gebruik.

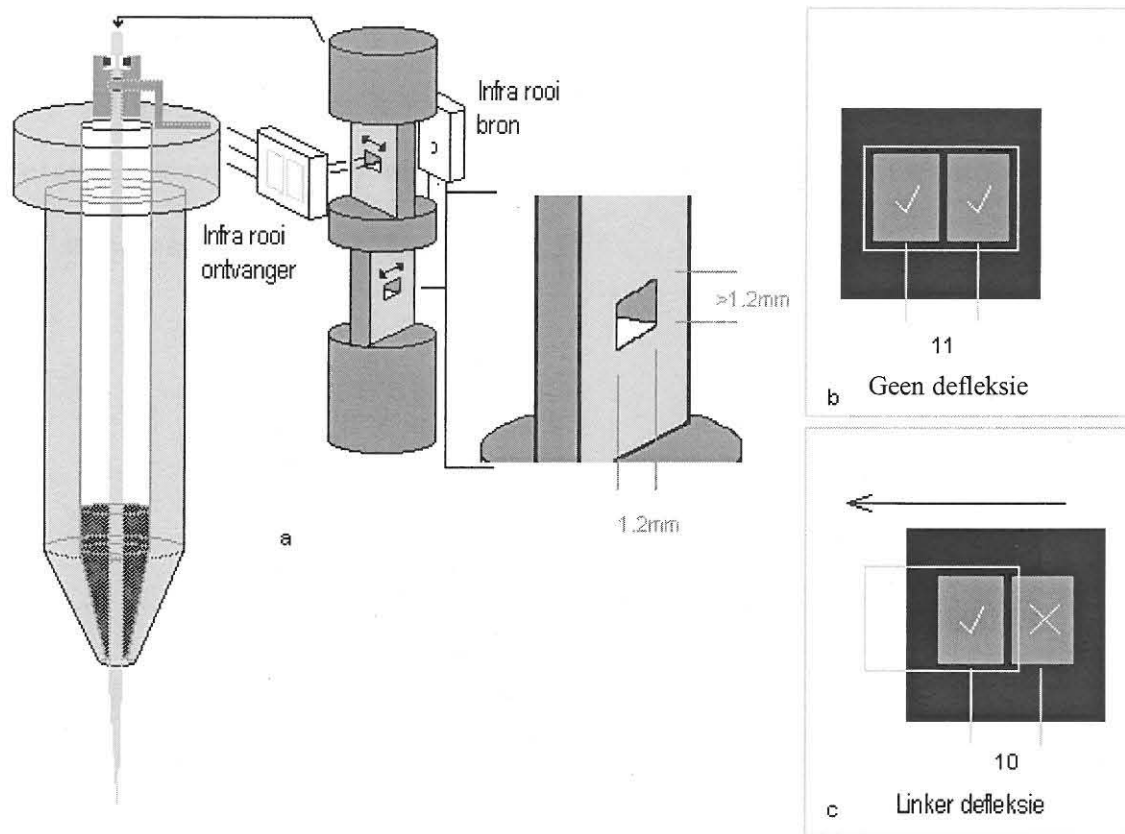
Behalwe die konstruksie hang die sensitiwiteit van 'n voelpen ook af van die sensors. Daar is geëksperimenteer met Hall-effeksensors, rekstrookies, infrarooi sensors en nabyheidsensors.

3.1.1 Infrarooi sensors

Eksperimente is gedoen met infrarooi senders en ontvangers om dit as sensors te gebruik om die posisie van die sensorpunt te bepaal. Hierdie infrarooi senders en ontvangers is soortgelyk aan dié wat in rekenaarmuise voorkom. In die rekenaarmuise is dit dié sensors wat verantwoordelik is vir die opname van die data waarmee die koördinaatposisies van die skermwyser bereken word. Die twee ontvangers, in een pakket, is ongeveer 1mm uit mekaar gespaseer. Beweging van die sensorpunt in 'n sone word gemeet deurdat die sensorpunt voor die infrarooi bron inbeweeg, sodat geen infrarooi lig die ontvangers bereik nie. Dié konsep word eenvoudiger uitgebeeld by wyse

van Figuur 3.5. Figuur 3.5b. toon 'n skets van die ontvangerpakke met sy twee ontvangers. Die sender en ontvanger word geskei deur 'n plastiese obstruksie met 'n opening groot genoeg sodat, indien dit reg in die middel van die sensors geplaas word, die infrarooi lig beide ontvangers kan bereik. In Figuur 3.5b is al twee die ontvangers deur die opening sigbaar, terwyl Figuur 3.5c vanweë die defleksierigting slegs een sigbare sensor toon. Om die idee te implementeer, is 'n spesiale vensterblokkie met presies die regte grootte openinge gemaak en op die sensorpunt gemonteer terwyl die sensors gerangskik is rondom die vensterblokkie. Figuur 3.5a toon die vensterblokkie, asook hoe dit met die stelsel geïntegreer word.

'n Probleem van dié konsep is dat die infrarooi straalhoek vals lesings veroorsaak wanneer die venster die bron nader as gevolg van 'n defleksie. So sal bv. al twee die X-as-ontvangers infrarooi lig waarneem, in plaas van slegs een wanneer die venster die bron nader tydens defleksie in die Y-sowel die X-as. Infrarooi sensors kan wel gebruik word om 'n eendimensionele voelpe daar te stel.

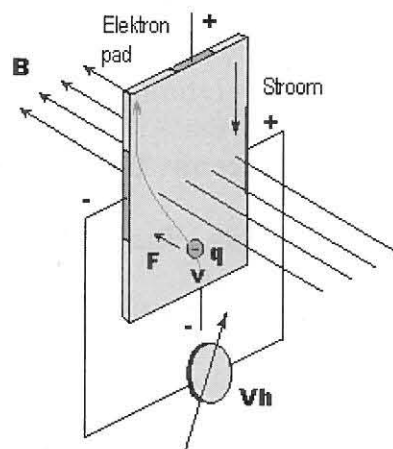


Figuur 3.5 Die implementering van infrarooi sensors op 'n voelpen

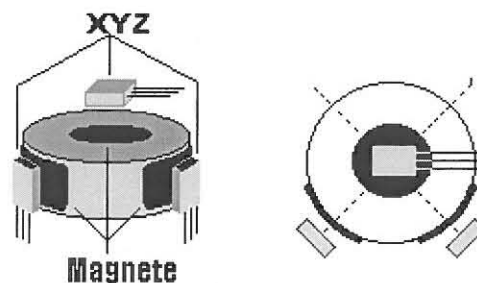
3.1.2 Hall-effeksensors

Die Hall-effek wat in 1897 deur E.H. Hall waargeneem is, maak staat op die kragte wat op elektrone inwerk, wanneer dit blootgestel word aan 'n magnetiese veld [7, p. 785]. Beskou Figuur 3.6. Elektrone, voorsien vanaf 'n stroombron gekoppel aan die bo- en onderkante van die plaat, vloei in die plat geleidende plaat, wat in 'n magnetiese veld geplaas word. Voltmeter, V_h , word aan die linker- en regterkant van die plaat gekoppel.

As gevolg van die magnetiese veld word elektrone na die regterkant gedeflekteer, wat daardie kant dan negatief laai teenoor die linkerkant. Die lesing op V_h word die "transverse potensiaalverskil" genoem. Die polariteit daarvan hang af van die magnetiese veldsterkte, intensiteit, en die stroomsterkte [11, p.78].



Figuur 3.6 Die beginsel van die Hall-effek



Figuur 3.7. Hall-effeksensors, gebruik vir driedimensionele waarneming

Indien 'n magneet aan die sensorpunt vasgemaak word, en drie Hall-effek sensors gerangskik word soos aangetoon in Figuur 3.7, kan driedimensionele beweging van die magneet waargeneem word. Wat hierdie sensors verder geskik maak vir dié doel, is dat die analoogsein wat verkry word, geïnterpreteer kan word in terme van hoeveel die puntdefleksie was. Die sagteware kan dan sodanig geskryf word, dat die voelpe gedurende die aftasting teen 'n sekere drukking aan die model raak, wat sal verhoed dat 'n model verlore raak tydens sy omtreksbepaling (bespreek in Hoofstuk 4). Tydens normale koördinaataftasting kan die grootte van die defleksie ook geïnterpreteer word in terme van sub-resolusieafmetings. Sodoende kan die akkuraatheid verbeter word, sodat dit hoër is as die resolusie wat die CMM toelaat. Die verlangde sensitiwiteit van die voelpe word deur die gebruiker in die rekenaarsagteware gespesifiseer.

3.1.3 Rekstrokies

Lord Kelvin het in 1856 ontdek dat die weerstand van 'n elektriese geleier verander, wanneer die geleier vervorm word. Die weerstand van 'n geleier word uitgedruk as [32, p. 4]

$$R = \rho l/A, \quad (3.1)$$

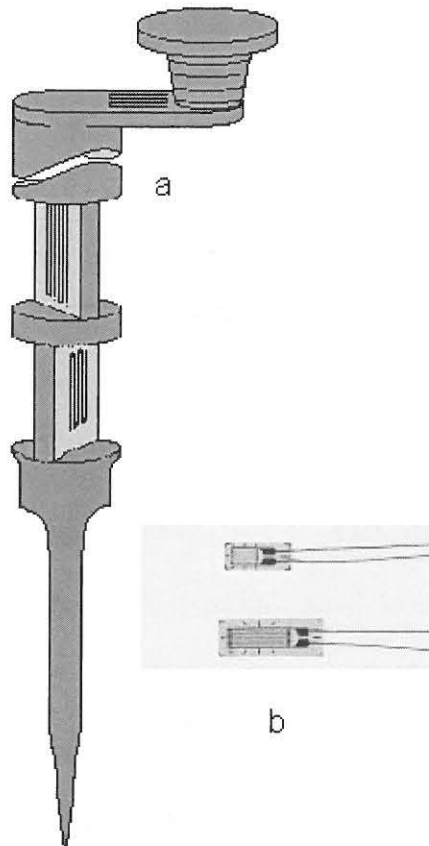
waar R = weerstand

l = lengte van geleier

A = deursnitarea en

ρ = resistiviteit

Rekstrookies, waarvan 'n voorbeeld in Figuur 3.8b getoon word, is veral bekend vir hul akkuraatheid. Wanneer dit geplak word op 'n staaf met 'n vorm soos getoon op die illustrasie in Figuur 3.8a, kan driedimensionele beweging waargeneem word.



Figuur 3.8 Rekstrookies gebruik as voelpensensors

Dit is belangrik dat die materiaal wat gebruik word, van so aard is dat daar altyd terugkeer word na die oorspronklike vorm, wanneer daar geen defleksie is nie. Die weerstandsverandering word waargeneem deur

middel van 'n Wheatstone-brug, wat verder versterk word deur 'n operasionele versterker. Die uitset (V_{uit}) van die brug word gedefinieer as:

$$V_{uit} = K \epsilon N V_{in} / 4 \quad [32, \text{p. 153}]. \quad (3.2)$$

Waar K = strokiefaktor,

V_{in} = brugspanning,

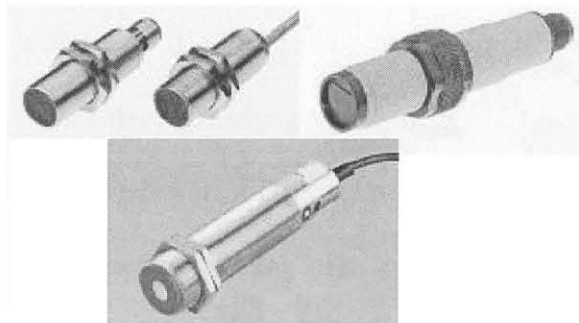
ϵ = stres en

N = aantal aktiewe arms in die brug.

3.1.4 Nabyheidsensors

Nabyheidsensors is sensors wat nie fisies aan 'n voorwerp raak nie, maar 'n sein genereer wanneer die voorwerp op 'n sekere afstand van die sensor is. Deur die sensor op 'n sekere afstand van die voorwerp te hou deur middel van bewegings in die Z-as, kan dié sensors vir data-opname gebruik word. Sommige nabyheidsensors lewer 'n analoë uitsetsein, wat die teenwoordigheid van die Z-as onnodig kan maak, omdat die analoogsein geïnterpreteer kan word as 'n afstand. Induktiewe nabyheidsensors tans beskikbaar, kan afstande in die omgewing van 0.1 tot 20 mm waarneem. Induktiewe nabyheidsensors is veral geskik vir die opspoor van koper, aluminium en vlekvrige staal. Kapasitiewe nabyheidsensors kan weer gebruik word om nie-geleidende materiale (bv. plastiek en hout) op te spoor. Spoorafstande van voorwerpe wissel na gelang van die tipe materiaal. Ultrasoniese nabyheidsensors is geskik vir

enige materiaal en kan tot 1200 mm en groter afstande bepaal. Die herhaalbaarheid van metings wat tans verkry kan word, is $\approx 1\text{mm}$ [27, p. 1179], wat onaanvaarbaar is vir data-opname in die μm -strek. Groter akkuraatheid is wel teen hoër koste beskikbaar en sal waarskynlik in die toekoms ook verbeter. Die nadeel van nabyheidsensors is dat dit tans redelik duur is. Betroubaarheid en akkuraatheid sal ook verbeter soos die tegnologie groei. Dit is wel somtyds nodig om versyfering te doen sonder fisiese kontak. Daarom is dit belangrik om in die toekoms nabyheidsensors in gedagte te hou. Figuur 3.9 toon die drie soorte nabyheidsensors soos genoem en wat vanaf RS Componets SA verkry kan word [26, pp. 1177-1186].



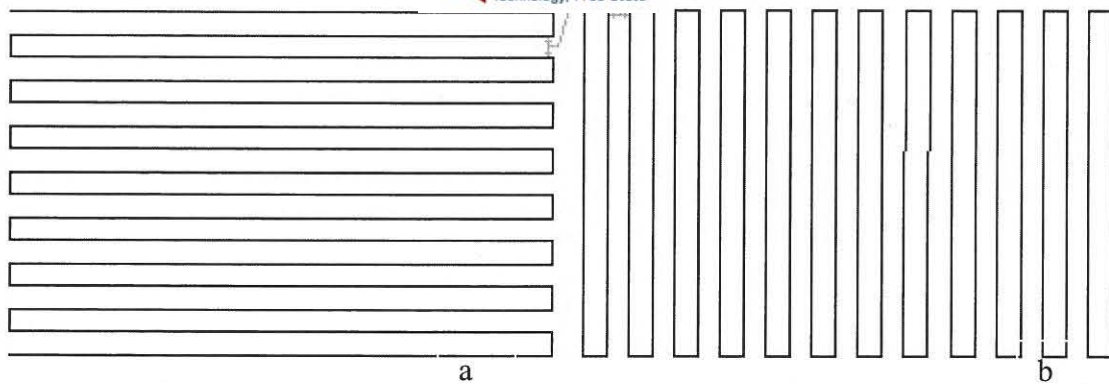
Figuur 3.9. Nabyheidsensors

Ongeag daarvan hoe akkuraat die voelpe en die motorbeweging is, bly dit nog steeds nodig om beide te kalibreer, voordat aftasting gedoen kan word. CMM's het gewoonlik 'n datumsfeer, waarvolgens die voelpe en

die motorakkuraatheid getoets word, ten einde foute in die sagteware reg te stel. Die datumsfeer word aangebring op die blad van die CMM-masjien en is sigbaar op die Renishaw® Cyclone, soos getoon in Figuur 2.5. Die dimensies en posisie van die sfeer is bekend aan die sagteware wat die voelpen beheer om sekere bewegings relatief tot die sfeer uit te voer. Sodoende word die CMM gekalibreer in reaksie op die voelpenterugvoering.

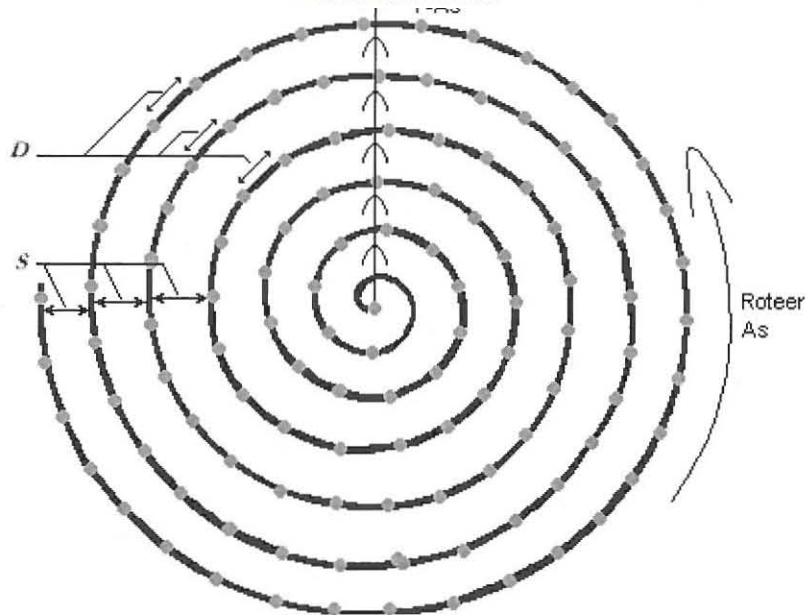
3.2 Voelpenroetes vir data-opname

Die voelpen, beheer deur die drie motore (trappiesmotore in die geval van die Max NC 10), word oor die model beweeg om die data op te neem. Hoe meer akkuraatheid met die aftasting verlang word, hoe meer data word per vierkante millimeter opgeneem. Die resolusie van die aftasting word met behulp van die sagteware deur die operateur geselekteer. Verskeie CMM's, onder andere ook die Renishaw® Cyclone, volg 'n voelpenroete waar die voelpen oor die modeloppervlak beweeg soos geïllustreer in Figuur 3.10a en b, terwyl die Z-as vertikaal beweeg, na gelang van die modelprofiel. Die rigting hang van die operateurskeuse af. Die afstand (d) tussen die lyne word dus ook deur die operateur bepaal.



Figuur 3.10. Algemene voelpenroetes

'n Ander moontlikheid wat vir data-opname oorweeg is, word in Figuur 3.11 getoon. Die metode kan gebruik word by die Max NC 10, waar die model onder die voelpen rondbeweeg. Indien die X- of Y-as vervang word met 'n horisontale, roterende as, kan 'n dataroete gevolg word, wat vergelyk kan word met die groewe van 'n musiekplaat. Die aanvangspunt vir die metode sal in die middelpunt van rotasie wees. Die resolusie hang dan van die afstand D na S af soos geïllustreer in Figuur 3.11. Data afgetas deur dié metode sal ook suksesvol vergelyk kan word. Gemene middelpunte, waar daar met aftasting begin sal word, kan vanaf die omtrekaftasting bepaal word.



Figuur 3.11 Spiraalroete vir data-aftasting

3.3 Die vergelyking van driedimensionele voorwerpe

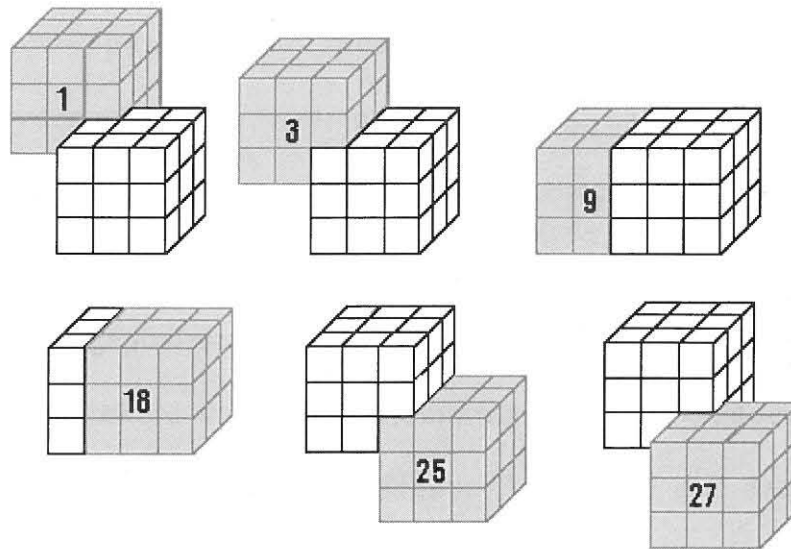
Die rede waarom die vergelyking van standaard geometriese datalêers slegs deur komplekse en moeisame programmering bewerkstellig sal kan word, word in hierdie hoofstuk verduidelik. Eerstens word die vergelyking van driedimensionele data, waar geen verwysingspunte tot mekaar beskikbaar is nie, uiteengesit, sodat die vergelykingsmetodes wat in die navorsing gebruik is, daarby aansluiting kan vind. Die afwesigheid van verwysingspunte beteken dat die data van twee modelle enige rotasie tot mekaar kan hê. Die data van een model moet wiskundig met behulp van die sagteware geroteer word, sodat alle rotasiemoontlikhede tussen die

modelle vergelyk kan word. Die beste vergelykingsposisie kan daarna geïdentifiseer word. Die vergelyking van driedimensionele data waar geen verwysingspunte teenwoordig is nie, is 'n uitgerekte proses, maar dit is teoreties moontlik. Die volgende paragrafe handel oor die passing en rotasie van stelle geometriese koördinaatdata waar geen verwysingspunte gegee word nie. Die omvang van die driedimensionele datavergelyking sal uit die bespreking duidelik word.

Die metode wat egter met die navorsingsprojek gevolg is om die modeldata te vergelyk, behels oriëntasie van modelle met mekaar, voordat met aftasting begin word. Indien modelle in verskillende posisies relatief tot mekaar afgetas word, sal data-oriëntasie met behulp van die sagteware gedoen moet word, ten einde die regte koördinaatpare met mekaar te kan vergelyk. Data-oriëntasie wat datavergelyking sonder verwysingspunte moontlik sal maak, word nou bespreek.

Daar kan aan die data van driedimensionele lêers gedink word as driedimensionele biskaarte waar elke bis uit 'n binêre waarde (1 vir die teenwoordigheid van materiaal en 0 vir 'n leemte) bestaan. Dit geld slegs waar die data vyfdimensioneel versyfer is, wat beteken dat die data omnidireksioneel opgeneem word en geen verborge aansigte bevat nie. In matriks N^3 , net groot genoeg vir al die data van 'n spesifieke model, word die binêre waardes gestoor en is gevolglik automaties aan 'n X-, Y- en Z-

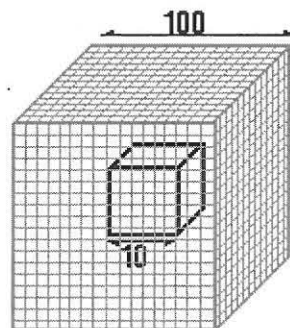
koördinaatposisie gekoppel. Wanneer twee stelle driedimensionele biskaarte met mekaar vergelyk word, moet al die moontlike posisies en rotasies waarin die data tot mekaar kan wees, vergelyk word.



Figuur 3.12 Die vergelyking van twee 3^3 -matrikse

Figuur 3.12. illustreer twee 3^3 -matrikse wat met mekaar vergelyk word d.m.v. datapassing en toon sommige van die 27 moontlike posisies wat met mekaar vergelyk kan word. In die praktyk sal so 'n datamatriks veel groter wees, wat sal beteken dat baie meer kombinasies met mekaar vergelyk sal moet word. Die resultate van die vergelykings in al die moontlike vergelykingsposisies moet gestoor word, sodat die akkuraatste vergelykingsposisie uit al die antwoorde gekies kan word.

Indien die datamatriks sodanig is dat dit net groot genoeg is om al die modeldata te akkommodeer, kan aanvaar word dat, indien die geometriese data van twee soortgelyke modelle met mekaar vergelyk word, die relatiewe middelpuntkoördinate van die twee datastelle om en by na dieselfde relatiewe punt op die model verwys, ongeag die relatiewe modelrotasies tydens aftasting. Indien aanvaar word dat die twee modelle se middelpuntkoördinate op ongeveer dieselfde posisie in die datamatriks lê, kan die data van die twee modelle met mekaar vergelyk word deur slegs die vergelykingsposisies in die area van die middelpunt te vergelyk, waar die waarskynlikheid van ooreenstemming hoog is. Sodoende word datapassing verkort en tyd bespaar. Figuur 3.13 toon die data van 'n hipotetiese model in 'n 100^3 -matriks met sy vergelykingsvenster, 'n matriks van bv. 10^3 , rondom die modelmiddelpunt.



Figuur 3.13 Datapassing rondom die datamiddelpuntarea

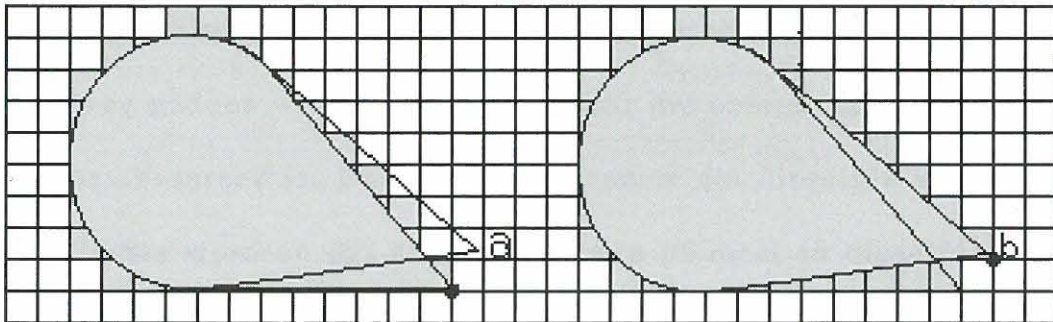
Die resultate van die vergelyking word gevolglik ook in die vorm van 'n tydelike resultaatmatriks (dieselfde grootte as die venstermatriks) gestoor,

sodat die vergelykingsposisie wat die beste ooreenkoms van die data verteenwoordig, geïdentifiseer kan word. Dié vergelykingsposisiewaarde verteenwoordig egter slegs die vergelyking van 'n model op 'n spesifieke driedimensionele rotasiehoek tot die ander, en is dus een van vele ander resultate. Elke resultaat word gekoppel aan 'n XYZ-rotasiewaarde, deur dit weereens in 'n spesifieke posisie in 'n ander resultaatmatriks te stoor. Figuur 2.2 het die X-, Y- en Z-as-rotasierigtings getoon. Die driedimensionele rotasies verteenwoordig al die moontlike posisies wat voorwerpe relatief tot mekaar kan hê. Die moontlike posisies sluit al die kombinasies van rotasie (deur 360°) wat die drie asse tot mekaar mag hê, in. Die een voorwerp, of in werklikheid die data van die voorwerp, moet dus geroteer word, sodat die data vergelyk kan word in alle moontlike posisies waarin die een model tot die ander kan verkeer.

Gestel dat die rotasies in 1° -stappe gedoen word. Die moontlike posisies wat vergelyk sal moet word is 360^3 en sal dus 46656000 wees. Uit die 46656000 moontlike antwoorde, sal die beste resultaat dus gekies word. Dit is dan ook die grootte van die resultaatmatriks vir al die rotasiemoontlikhede. Waar resolusies in die μm -meetgebied val, sal 'n rotasiehoek van 1° in werklikheid te groot wees. Hoe fyner die aftasresolusie is, hoe kleiner moet die rotasiehoek wees om die aftasresolusie te bevredig. 'n Kleiner rotasiehoek gee aanleiding tot 'n

groter aantal vergelykingsposisies. Die vergelyking van die data in alle moontlike rotasieposisies is dus 'n omslagtige proses.

Die proses word verder gekompliseer en verleng deur die berekening van nuwe datakoördinate vir elke rotasie. Figuur 3.14 toon 'n dwarsnit van 'n model en 'n geroteerde spookbeeld van dieselfde model X° . Die nuwe berekende koördinate (B) van die model moet dan sodanig bereken word, dat dit op 'n geldige koördinaatpunt van die matriks val (waar die X- en Y-lyne op die ruitkaart kruis). Dit is die koördinaatpunt wat die CMM sou waarneem, indien die aftasting in daardie spesifieke rotasieposisie gedoen is.



Figuur 3.14 Koördinaatberekennings vir datarotasie

Vergelyking van modelle by wyse van dié komplekse metode sal in baie gevalle steeds die menslike oordeel vereis, indien verwysingspunte vir die datavergelyking benodig word. Dink aan twee identiese sfere, waarvan daar van elk 'n plat vlak geskaaf word tot en met 10% van die deursnee.

Veronderstel dat die plat vlak van een sfeer aan sy bokant geleë is, terwyl dié van die ander aan die onderkant is. Vir die rekenaar sal die modelle identies wees. Dit sal dus in sommige gevalle nodig wees om 'n paslike verwysingsraamwerk te hê, sodat datavergelyking korrek geskied. Die verwysingspunte kan byvoorbeeld daargestel word deur 'n grafiese voorstelling van die modelle op die rekenaarskerm te toon en dan die een model te roteer, totdat dit min of meer ooreenstem met die ander, om sodoende 'n basiese verwysingsraamwerk te skep waarin die rekenaar sy berekenings kan doen. Indien die vergelykingsproses wat vroeër bespreek is, nou binne bv. 10° (i.p.v. 360°)-rotasies in alle rigtings gedoen word, sal dit die aantal vergelykingsmoontlikhede aansienlik verminder.

Gestel die data van twee modelle word geneem uit 'n bo-aansig, soos dit normaalweg gedoen word, in plaas van deur die omnidireksionele metode, wat so pas bespreek is. Daar word aangeneem dat dieselfde vlak van die model afgetas word en dat die modelle min of meer in dieselfde posisie tot mekaar lê. Die waarskynlikheid dat die twee modelle se X-, Y- en Z-rotasies tot mekaar identies is wanneer die modelle met die oog georiënteer word, is beperk. Gestel daar word vereis dat die rotasies in die X- en Y-asse, wat in wese die horisontale tilt van die modelle verteenwoordig, nie 10° -rotasie oorskry nie, terwyl die Z-rotasie wat die horisontale rotasie van die model verteenwoordig, onbeperk kan wees. In so 'n geval kan die Z-rotasies vooraf deur middel van omtrekaftasting met

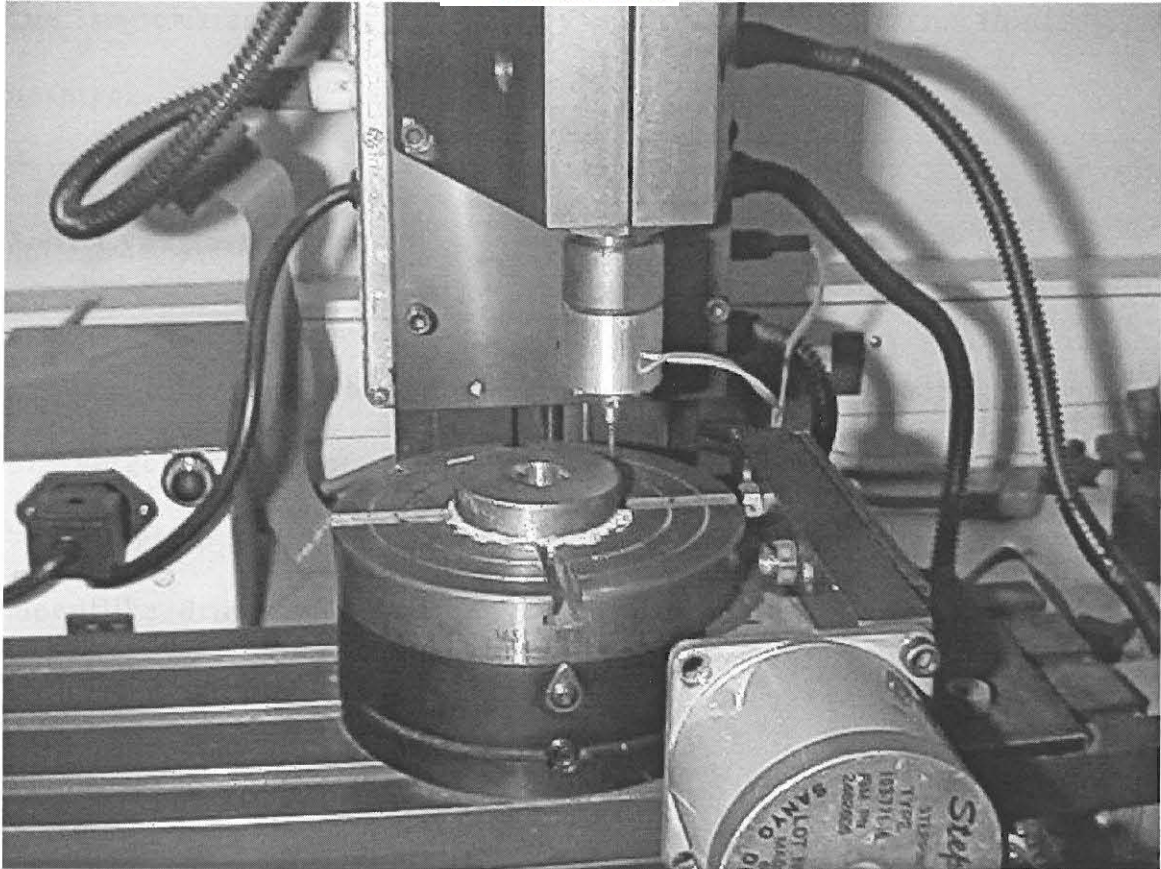
mekaar gerig word, terwyl X- en Y-datarotasië van slegs 10° in die sagteware nodig sal wees.

Die metode wat in die studie gevolg is, is een waar die modelle vooraf met mekaar georiënteer word, sodat geen datarotasië in die sagteware nodig is nie. Dataverskuiwing in die X- en Y-asse word wel deur die sagteware gedoen. Slegs een vlak van die modelle word met mekaar vergelyk. Data word vanuit die bo-aansig geneem (soos waargeneem deur die voelpe), wat beteken dat daar verborge aansigte op die model mag wees, waarvan geen data opgeneem sal word nie. Omdat die X- en Y-koördinate van een datastel met dieselfde X- en Y-koördinate van die ander gekoppel moet word (sodat die Z-koördinate met mekaar vergelyk kan word), moet daar gepoog word om die aftasting sódanig te doen dat die opname van die X- en Y-koördinate op dieselfde relatiewe plekke op die modelle gedoen word. Die voorwaardes waaraan voldoen moet word, ten einde te verseker dat die modelle op presies dieselfde relatiewe posisie lê tydens aftasting, sluit in dat daar

- geen tilt in die X-asse is nie,
- geen tilt in die Y-asse is nie en
- geen rotasië van die model in die X- en Y-asse sal wees nie.

Om dié voorwaardes te bevredig moet modelle dus met mekaar georiënteer word, voordat data opgeneem kan word.

Eerstens word die omtrekke van die twee modelle afgetas, waarna een model horisontaal geroteer (deur middel van die ekstra gemonteerde as, getoon in Figuur 3.15) word, totdat die twee modelsilhouette (wat op die rekenaarskerm vertoon word, Figuur 6.7) reg op mekaar lê. Nadat oriëntasie in die Z-as voltooi is, word bepaal of X- of Y-rotasies nodig is. Dit beteken bloot dat daar bepaal moet word of die modelle teen dieselfde hoek op die blad geplaas is. Omdat daar geen motore is om die X- en Y-as mee te roteer nie, moet dit met die hand gedoen word. Om te bepaal of daar enige ongelykhede tussen die modelle is, word 'n aantal lyne in die X- en Y-rigtings afgetas en met mekaar vergelyk, om sodoende grafies aan te toon of die modelle in dieselfde vlak lê. Die roetines om die twee modelle met mekaar te oriënteer word in Hoofstuk 6 bespreek, waar die werking van die sagteware verduidelik word. Wanneer die twee modelle in dieselfde posisies relatief tot mekaar lê, word die modelle gewoonweg afgetas, met die X- en Y-posisies vir beide modelle dieselfde. Die resolusie en die lynspasiëring van die modelle (gespesifiseer deur die gebruiker in die sagteware) moet dieselfde wees. So ook moet die X-lengte en Y-breedte van die aftasarea (ook m.b.v. die sagteware deur die gebruiker bepaal) dieselfde wees. Waar aftasting gedoen word met die oog op vergelyking met gestoorde data, moet eers bepaal word met watter verstellings die aftasting gedoen moet word. Dit word in Hoofstuk 6 bespreek, waar die sagteware verduidelik word.



Figuur 3.15 Die roteer-as gemonteer op die Max NC 10-CMM

3.4 Die stapberekeningslogika tydens aftasting.

Een van die verskille tussen CMM- en NC-masjiene, in Hoofstuk 2 bespreek, is die bewegingsroetes van die CMM wat onbekend is tydens aftasting. Die NC-masjien daarenteen voer sy bewegings uit volgens die data in die NC-programmatuur. Daar is dus sekere prosedures wat gevolg moet word wanneer besluit moet word oor die rigting waarin die volgende stap moet geskied. Die logika agter hierdie besluite word nou verduidelik.

Die sagtewareroetines wat daarby betrokke is word in Hoofstuk 6 bespreek.

3.4.1 Lynaftasting

Die bepaling van die bewegingstappe by lynaftasting is redelik eenvoudig in vergelyking met dié van omtrekaftasting. Slegs die vorige bewegingstap word tydens stapbepaling in aanmerking geneem. Die moontlike drie stande met lynskandering (van links tot regs), met die reaksie-opsies wat daarop volg, is as volg:

- Laaste beweging: Op
 - ♦ Opsie A: Regs
 - ♦ Opsie B: Op
- Laaste beweging: Regs
 - ♦ Opsie A: Af
 - ♦ Opsie B: Regs
 - ♦ Opsie C: Op
- Laaste beweging: Af
 - ♦ Opsie A: Af
 - ♦ Opsie B: Regs

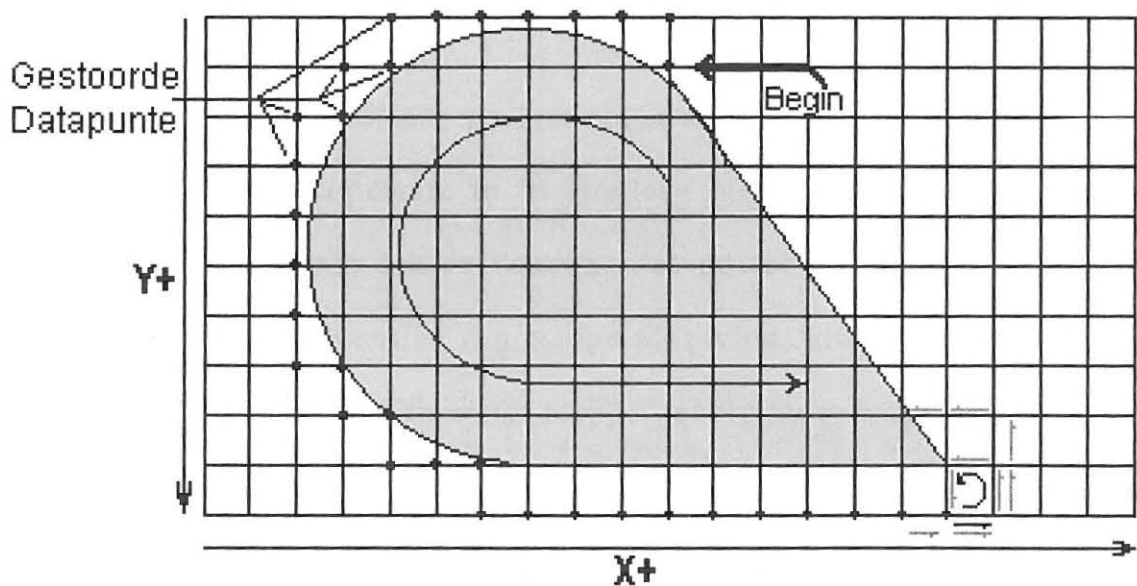
3.4.2 Omtrekaftasting

Die omtrekaftasting maak staat op die vier voorafgegaande bewegingstappe om die volgende bewegingsrigting te bepaal. Slegs die X- en die Y-asse is betrokke. Indien die omtrek teenkloksgewys afgetas word, word die besluitname as volg gedoen:

- Laaste rigting: X-
 - ♦ Opsie A: Y+
 - ♦ Opsie B: X-
 - ♦ Opsie C: Y-
 - ♦ Opsie D: X+
- Laaste rigting: Y+
 - ♦ Opsie A: X+
 - ♦ Opsie B: Y+
 - ♦ Opsie C: X-
 - ♦ Opsie D: Y-
- Laaste rigting: X+
 - ♦ Opsie A: Y-
 - ♦ Opsie B: X+
 - ♦ Opsie C: Y +
 - ♦ Opsie D: X-
- Laaste rigting: Y-
 - ♦ Opsie A: X-

- ♦ Opsie B: Y-
- ♦ Opsie C: X+
- ♦ Opsie D: Y+

Indien die vorige bewegingsrigting vir die beginposisie van die omtrekaftasting, getoon in Figuur 3.16, as X- of Y+ gegee word, kan gesien word hoe hierdie reëls geïmplementeer en gehoorsaam word tydens omtrekaftasting. In die rekenaarsagteware wat die omtrekaftasting beheer, is die voorafgaande bewegingstappe vir die aanvangspunt bekend en altyd dieselfde, omdat dié punt altyd vanuit dieselfde rigting benader word.



Figuur 3.16 Omtrekaftasting van 'n model

Die logika agter omtrekaftasting word egter gekompliseer a.g.v. voorsorgmaatreëls wat getref moet word om te verhoed dat die voelpe

die model “verloor” en die sagteware gevolglik in ‘n lus beland. As gevolg van die bewegingstraagheid van die as-aandrywingsmeganisme, en die gebrek aan volmaakte omnidireksionaliteit van voelpenne, word soms verskillende lesings verkry, wanneer ‘n sekere modelrant vanuit verskillende rigtings benader word. Omdat die beweging wat moet volg, bepaal word volgens die vier voorafgegaande bewegings, gebeur dit soms dat die voelpen in ‘n eindelose lus kan beland.

Figuur 3.16 toon die silhoeëtte van ‘n model met die moontlike koördinaatpunte wat op die posisies lê waar die lyne kruis. Die skerp punt van die figuur lê reg op só ‘n koördinaatpunt. Indien die aftasreëls wat beskryf is, gevolg word, kan dit gebeur dat, wanneer die punt vir die tweede keer bereik word, die voelpen geen aanraking waarneem nie. Om te voorkom dat die sagteware in ‘n eindelose lus (sirkelpyltjie op Figuur 3.16) beland, moet daar sekere voorwaardes gestel word, wat spesifiseer dat, indien sekere sekwensies van stappe plaasvind, uitsonderings gemaak moet word op die roetines en ander stappe gevolg moet word. In die geval van Figuur 3.16 was die uitsonderingsekwensie $X+, Y-, X-$ en $Y+$. Op die sekwensie word gereageer met $X+, Y-, Y-, X-$ en $X-$. Hierdie reaksiesekwensies kan op hul beurt ook weer in lussekwensies beland, waarvoor daar om die beurt ook getoets en op gereageer moet word. Tabel 3.1 toon hierdie sekwensies en die reaksies wat daarop moet volg. Die sagtewareroetines wat dit moontlik maak, word in Hoofstuk 7 bespreek.

Tabel 3.1 Lusvoorkomende stappe by omtrekaftasting

Lussekvensie	Reaksie
X+, Y-, X-, Y+	X+, Y-, Y-, X-, X-
Y-, X-, Y+, X+	Y-, X-, X-, Y+, Y+
X-, Y+, X+, Y-	X-, Y+, Y+, X+, X+
Y+, X+, Y-, X-	Y+, X+, X+, Y-, Y-
X+, X+, Y-, Y-, X-, X-, Y+, Y+	Beweeg na laaste raakposisie.
Y-, Y-, X-, X-, Y+, Y+, X+, X+,	Beweeg na laaste raakposisie.
X-, X-, Y+, Y+, X+, X+, Y-, Y-	Beweeg na laaste raakposisie.
Y+, Y+, X+, X+, Y-, Y-, X-, X-	Beweeg na laaste raakposisie.

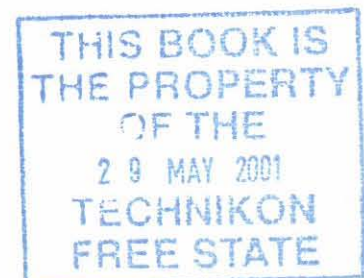
3.5 Opsomming

Uit hierdie hoofstuk behoort dit duidelik te wees waarom die vergelyking van driedimensionele voorwerpe nie so eenvoudig is as om net twee neutrale lêers met mekaar te vergelyk nie, asook hoe die afwesigheid van 'n verwysingsraam die vergelyking van twee stelle geometriesedata kompliseer. Die metode wat gevolg is om modelle te vergelyk, is bespreek, asook die vereistes vir die hardeware wat betrokke is. Die logika agter stapbepaling is ook verduidelik.

GEOMETRIESE LÊERFORMATE

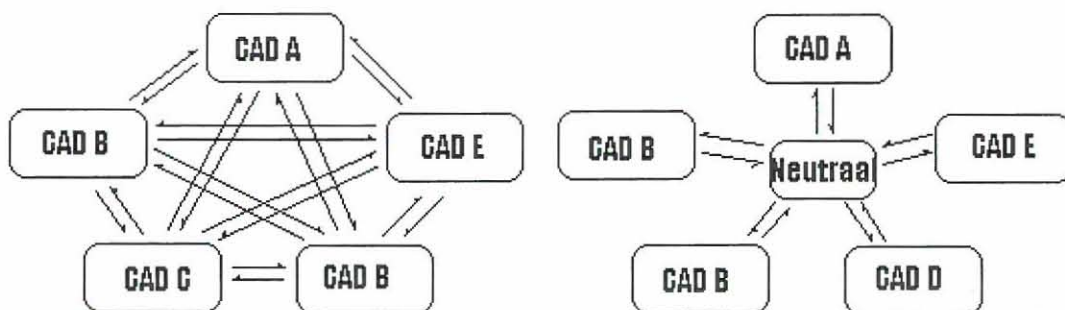
4.1 Inleiding

Die formaat waarin geometriese data gestoor word, is krities wanneer die data van een lêer met dié van 'n ander vergelyk moet word. Vektor- of “metafile”-lêerformate is beskrywende formate, wat 'n model beskryf deur lyn, reghoek, boog, driehoek en ander soortgelyke vormentiteite. Die interpretasie van die beskrywings van twee lêers, sodat die data met mekaar vergelyk kan word, is 'n moeisame en komplekse taak [4, p. 1250]. Met die oog op data-oordrag na ander CAD-sagteware en datavergelyking, is daar in die voorstudie navorsing gedoen oor verskillende neutrale-lêerformate. Die dokumentasie oor lêerformate is geweldig beperk en in baie gevalle vaag of verouderd [20, p. 5]. In die volgende paragraaf word 'n kort bespreking gevoer oor neutrale-lêerformate.



4.2 Neutrale-lêertipes

Neutrale-lêerformate is standaard lêertipes wat gebruik word om datakommunikasie tussen CAD-sagteware te bewerkstellig. Omdat daar reeds soveel lêerformate bestaan, is dit moeilik vir 'n CAD-program om alle lêer-tipes te kan vertolk [21, p. 1]. Daarom is daar besluit om sekere lêerformate te identifiseer en dit as 'n standaard te gebruik om geometriese data tussen verskillende CAD sagteware te vertolk. Dié lêerformate moet buigsaam genoeg wees sodat dit alle voorwerpvorms kan beskryf. Figuur 4.17 toon hoe neutrale-lêerformate die in- en uitvoer van lêers tussen CAD-sagteware vereenvoudig. Die aantal lyne wat elke CAD pakket verlaat verteenwoordig die aantal lêers wat die pakket moet ondersteun. Daar kan duidelik gesien word hoe die neutrale lêerformaat die las van lêervertolking verlig.



Figuur 4.1 Voordele van neutrale-lêerformate

4.2.1 Die AutoCAD *dxf*-lêerformaat

AutoCAD *.dxf* is 'n vektor-tipe lêer wat bykans enige voorwerpvorm denkbaar kan beskryf. Sy groot aantal vormelemente maak die lêerformaat egter baie kompleks [5, pp. 796-829]. In sy mees basiese vorm lyk die *.dxf*-lêer as volg:

0	: Entiteitgroepekode
SECTION	: Die begin van seksie
2	: Entiteitkode
999	: Opmerkingskode
Hierdie is net 'n lyn.	: Opmerkings
0	: Entiteitgroepekode
LINE	: Begin van element
8	: Laagnommergroepekode
0	: Laagnommer
10	: Xa-koördinaatkode
1.1	: Koördinaat
20	: Ya- koördinaatkode
3.42	: Koördinaat
30	: Za- koördinaatkode
2.5	: Koördinaat
11	: Xb- koördinaatkode
1.4	: Koördinaat
21	: Yb- koördinaatkode
4.3	: Koördinaat
31	: Zb- koördinaatkode
6.43	: Koördinaat
0	: Entiteitgroepekode
ENDSEC	: Einde van seksie
0	: Entiteitgroepekode
EOF	: Einde van DXF-lêer

Die kort gedeelte beskryf 'n lyn tussen twee stelsel XYZ-koördinate en kom redelik eenvoudig voor. Wanneer voorsiening gemaak moet word om al die voorwerpbeskrywings te kan interpreteer, word dit 'n groot taak. Die rede waarom die lyn-entiteit getoon word, is omdat dit die entiteit is

wat gebruik word in die sagteware om die data om te skakel na die *.dxf*-lêerformaat, sodat dit deur ander CAD-programme gebruik kan word. Indien verbindingslyne tussen al die datakoördinate getrek word, kan 'n grafiese beeld van die model op die rekenaarskerm vertoon word. Dit is waarom slegs die lyn-entiteit benodig word om data sonder baie moeite na die *.dxf*-lêerformaat oor te skakel. Die spasiëring van elke letter en syfer moet presies op die regte plek in die datalêer geskryf word, met die onsigbare ASCII-karakters op die regte plekke. As die formaat nie presies reg is nie, kan ander CAD-sagteware nie die data lees nie, of dit sal verkeerd gelees word.

4.2.2 Die *.iges*-lêerformaat

Hierdie lêerformaat is ook baie kompleks en veral algemeen gebruik by hoog aangeskrewe CAD-programme. Die dokumentasie van geometriese entiteite wat die data beskryf in *.iges* v4, sal meer as 500 bladsye kan vul [20, p. 484]. Dit is 'n neutrale vektor, sowel as 'n 3D-lêersoort, wat deur alle platforms ondersteun word. In die navorsing is die moontlikheid oorweeg om *.dxf*- sowel as *.iges*-lêersoorte, wat albei deur die Renishaw® Cyclone CMM gegenereer word, met mekaar te vergelyk. Die beskrywing van die geometriese data maak dit egter baie moeilik om die lêer te interpreteer as datapunte wat vergelyk kan word. Selfs wanneer 'n model

vir 'n tweede keer op dieselfde posisie afgetas word, word die data in die lêers verskillend gedefinieer.

4.2.3 Die *.stl*-lêerformaat

Die *.stl* of stereolitografie wat in 'n ASCII of binêre lêerformaat kan wees, is ontwerp en word gebruik vir modelvervaardiging en is die standaard uitset van baie snelprototiperingsmasjiene [29, p. 1]. Die data word gestoor in die vorm van 'n menigte aantal driehoekies wat gesamentlik die modeloppervlak opbou, sonder dat daar enige openinge of oorvleuling tussen die driehoekies bestaan. Data gestoor in die ASCII-formaat, kan maklik geles word, maar dié formaat neem baie geheuespasie in beslag. Al die driehoekdata kom tussen die “solid” en “endsolid” sleutelwoorde voor en elke driehoeksgroep bestaan uit 'n normale posisiewaarde en drie Cartesiese koördinate vir die betrokke driehoek [1, p. 1]. Kodelys 4.1 toon die beskrywing van 'n enkele driehoek in 'n *.stl*-lêer.

Die omskrywing van data na 'n *.stl*-lêer is baie ingewikkeld. Daar bestaan selfs sagteware wat foutsporing op *.stl*-lêers doen [30, p. 1]. Data moet sodanig bereken word, dat alle grensende driehoeksyte dieselfde grootte het en daar geen opening tussen enige van die driehoeke bestaan nie. 'n Toename in resolusie van die data vereis 'n toename in die aantal driehoeke wat groter datalêers tot gevolg het.

Die *.stl*-lêerformaat bied geen eenvoudige metode om datavergelyking te doen nie. Selfs die skep daarvan vir uitvoering na ander CAD-sagteware, is baie ingewikkeld.

```
solid
...
facet normal 0.00 0.00 1.00
  outer loop
    vertex 2.00 2.00 0.00
    vertex -1.00 1.00 0.00
    vertex 0.00 -1.00 0.00
  endloop
endfacet
...
endsolid
```

Kodelys 4.1 'n .stl-lêer

4.3 Spesiale lêertipes vir afgetaste data

Dit is dus duidelik waarom dit nodig was om 'n lêerformaat te skep wat geskik is vir die vergelyking van data. Die meeste CMM-sagteware gebruik eie lêerformate om tydens aftasting data te stoor. Eers daarna word dit omskep in 'n neutrale-lêerformaat. Die formaat en doel van die lêers wat ontwerp is met die navorsingsprojek, word nou bespreek.

4.3.1 Die *.pos*-lêerformaat

Die *.pos*-lêer is verantwoordelik vir die data met betrekking tot die posisionele plasing van die model en bestaan uit twee dele. Die eerste deel bevat die omtrek-koördinate, terwyl die tweede deel die data van die

kruislyne vir die gelykheidsbepaling bevat. Dit word gestoor om as verwysing vir ander vergelykings te dien, wanneer die model vir die aftasting op dieselfde relatiewe posisie geplaas moet word as een wat voorheen afgetas is. Die formaat is as volg:

X-koördinaat	:Omtrekdata
Y-koördinaat	:Omtrekdata
X-koördinaat	:Omtrekdata
Y-koördinaat.	:Omtrekdata
(Ensovoorts)	:Omtrekdata
Round End	:Omtrekdata eindig
Aantal kruislyne (N)	:Lyndata begin
Lines	:Kodewoord
A	:Rigtingskode
Eerste lynnommer	:Lynnommer
Line	:Lyn-beginkode
X-koördinaat	:Koördinaatdata
Y-koördinaat	:Koördinaatdata
Z-koördinaat	:Koördinaatdata
(Ensovoorts)	:Koördinaatdata
Lines	:Kodewoord
A	:Rigtingskode
Nde lynnommer	:Lynnommer
Line	:Lyn-beginkode

X-koördinaat	:Koördinaatdata
Y-koördinaat	:Koördinaatdata
Z-koördinaat	:Koördinaatdata
(Ensovoorts)	:Koördinaatdata
End A	:Rigtingeinde
Lines	:Rigtingskodewoord
B	:Rigtingskode
Eerste lynnommer	:Lynnommer
Line	:Lyn-beginkode
X-koördinaat	:Koördinaatdata
Y-koördinaat	:Koördinaatdata
Z-koördinaat	:Koördinaatdata
(Ensovoorts)	:Koördinaatdata
Lines	:Rigtingkodewoord
B	:Rigtingskode
Nde lynnommer	:Lynnommer
Line	:Lyn-beginkode
X-koördinaat	:Koördinaatdata
Y-koördinaat	:Koördinaatdata
Z-koördinaat	:Koördinaatdata
(Ensovoorts)	:Koördinaatdata
End B	:Rigting einde
Max Z-diepte	:Maksimum tasdiepte

4.3.2 Die .hvs-lêerformaat

Die .hvs-lêer is die lêer wat al die geometriese koördinate van die data-aftasting bevat. Inligting insake die resolusie, afstand tussen die lyne, lengte en breedte, die aftasmetode en maksimum diepte van die Z-as word ook ingesluit. Die formaat is as volg:

Resolusie	:Tasresolusie
In X- of Y- rigting	:Tasrigting
Lynresolusie	:Afstand tussen lyne
X-lengte	:X-dimensie
Y-lengte	:Y-dimensie
Line	:Geometriese datakode
X-koördinaat	:Koördinaatdata
Y-koördinaat	:Koördinaatdata
Z-koördinaat	:Koördinaatdata
Line	:Geometriese datakode
(Ensovoorts)	:Koördinaatdata
End	:Lêereinde

4.3.3 Die *.set*-lêerformaat

Die *.set*-lêer is 'n lêer wat sekere omgewings- en masjienverstellings wat deur die sagteware opsies gesellekteer is, stoor. Wanneer die lêer geaktiveer word, word die betrokke sagteware-opsies verander na die betrokke waardes in die lêer. Die verstellings sluit motorspoed, resolusie, aftasmetode en voelpentipe in. Die formaat is as volg:

Resolusie	:Tasresolusie
Spoed	:Aftasspoed
Voelpensensitiwiteit	:Voelpensensitiwiteit
Voelpentipe	:Soort voelpe
Aftasroete	:Tipe aftasroete

4.3.4 Die *.cal*-lêerformaat

Dié lêerformaat stoor die kalibrasiewaardes wat bereken word tydens die kalibrasieroetines. Dit bestaan uit drie desimale waardes van masjien A gevolg deur die masjien B-waardes.

4.4 Opsomming

In hierdie hoofstuk is die navorsing wat gedoen is oor verskillende geometriese lêerformate met datavergelyking as motivering bespreek. Dit behoort duidelik te word waarom die vergelyking van neutrale-lêerformate 'n moeisame taak kan wees. Omdat hierdie lêerformate die voorwerpe beskryf, moet die data geïnterpreteer word in terme van geometriese koördinate, voordat dit vergelyk sal kan word. Die spesiale lêerformate wat ontwerp is vir die stoor van afgetaste voorwerpdata, ten einde dit met mekaar te vergelyk, is ook bespreek.

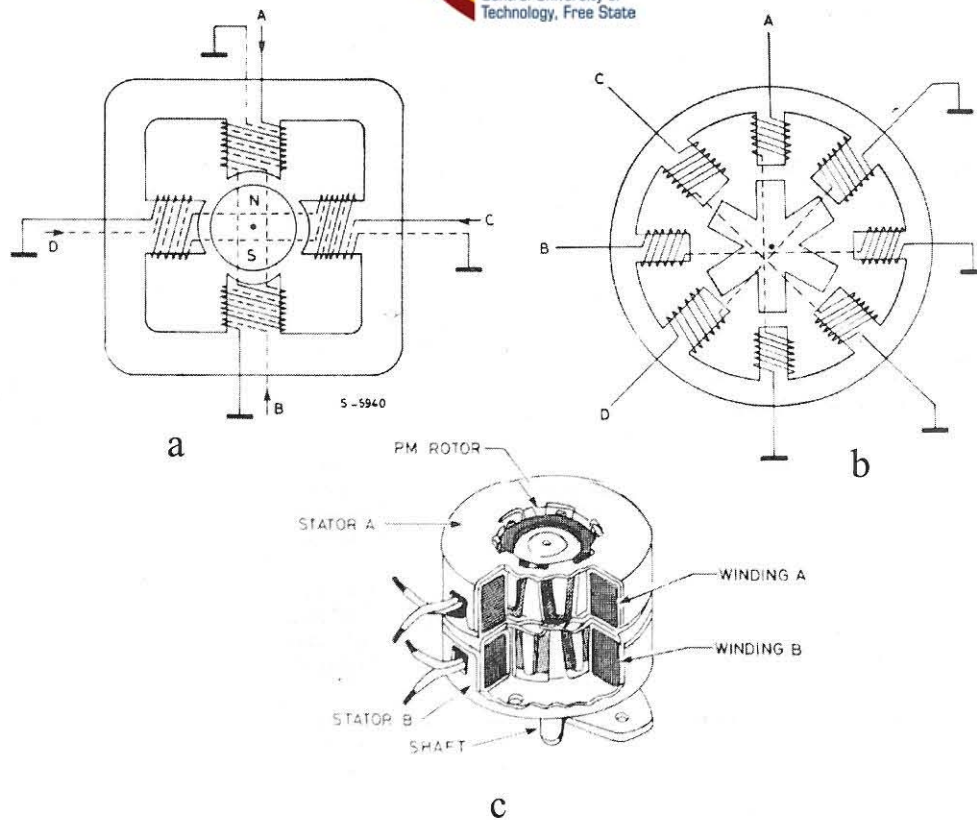
HOOFSTUK 5

DIE ELEKTRONIESE KOPPELVLAK

In dié hoofstuk word gekyk na die elektroniese koppelvlak tussen die rekenaar en die CMM-masjiene. Die elektroniese koppelvlak maak dit moontlik om twee Max NC 10 CMM-masjiene gelyktydig te beheer, asook twee ekstra trappiesmotore in te sluit vir die rotasie van modelle. Sonder die elektroniese koppelvlak is intelligente voelpenne ook nie moontlik nie. Omdat trappiesmotore verantwoordelik is vir die asbewegings van die aftasmasjiene, word in die eerste afdeling van hierdie hoofstuk sekere algemene punte rakende trappiesmotore bespreek. Die beheerlogika en mikrobeheerdersagteware kom ook later ter sprake.

5.1 Trappiesmotore

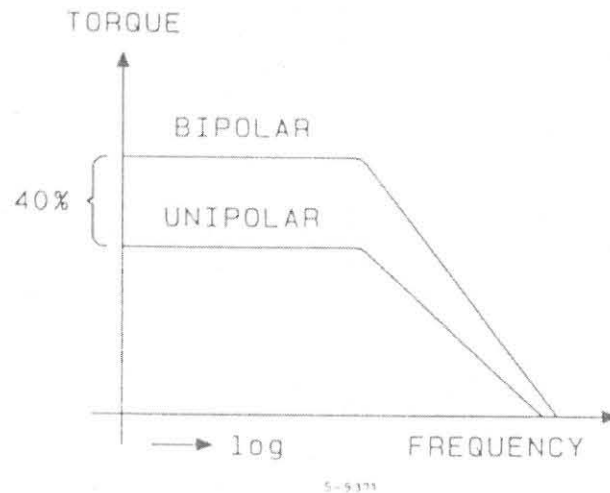
Soos die naam suggereer, beweeg trappiesmotore, in stede van aanhoudende rotasie, slegs 'n paar grade per stap. Dit is wat die trappiesmotor so geskik maak vir presisiebeheer wanneer 'n hoë wringkrag benodig word.



Figuur 5.1 Voorstelling van trappiesmotore

Figuur 5.1c stel 'n trappiesmotor voor en Figuur 1.2a en b toon die interne werking daarvan. Die geskikte rotasie-staphoek word verkry deur veelvoudige pole op die motors te implementeer [28, p. 72]. Die Sanyo Step Sym-trappiesmotore wat gebruik word op die Max NC 10, het 'n rotasie-staphoek van 1.8° per stap teen 79.23 g/mm , maar is ook beskikbaar teen 164.12 g/mm wringkrag [18, p. 1]. Twee metodes om trappiesmotore aan te dryf is die tweepool- en eenpool-koppelmetodes. Die tweepool-koppelmetode is vir die navorsing gebruik, omdat dit die meeste wringkrag lewer. Wringkrag word gedegradeer deur te hoë sekvensie-frekwensie, asook 'n tekort aan stroomtoevoer [28, p. 40]. 'n

Kromme van spoed teenoor wringkrag kan uitgestip word om die degradering te visualiseer [15, p. 69]. Figuur 5.2 toon die wringkragafname met 'n frekwensietoename vir beide een- en tweepool-koppelmetodes.



Figuur 5.2 Wringkragafname by hoëstapfrekwensies

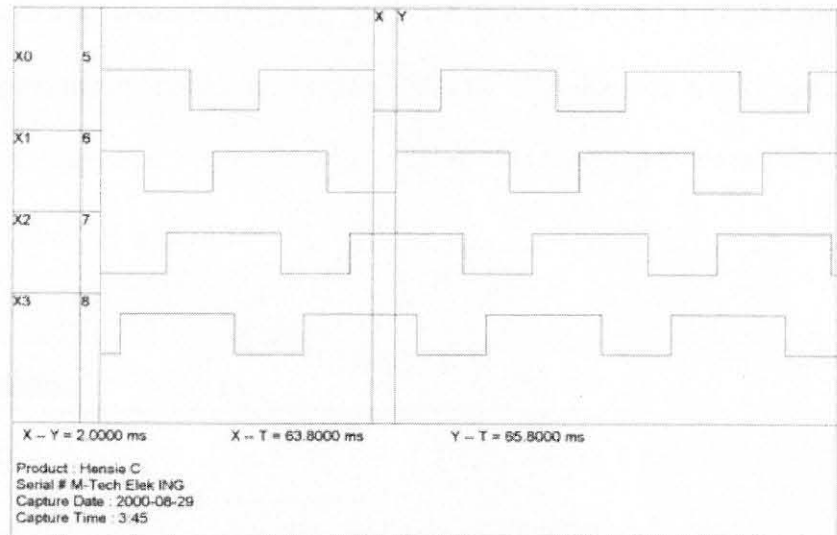
NPN-Si Darlington-kragtransistors met 'n maksimum kollektorstroom van 8 A is gebruik om aan die trappiesmotore voldoende stroom te voorsien. Die polariteit van die magnete moet in die regte sekwensie verander word om beweging te verkry. Die sekwensie word in Figuur 5.3a getoon soos verkry vanaf die Jobmatch Logic 1-logiese analiseerder. Die seindiagram word in Figuur 5.3b getoon.

Product : Hensie B
Serial # M-Tech Elek ING
Capture Date : 2000-08-29
Capture Time : 3:34

```

1495      1 1 0 1
1496      1 1 0 1
1497      1 0 0 1
1498      1 0 0 1
1499      1 0 1 1
1500      1 0 1 1
1501      0 0 1 1
1502      0 0 1 1
1503      0 1 1 1
1504      0 1 1 1
1505      0 1 1 0
1506      0 1 1 0
1507      1 1 1 0
1508      1 1 1 0
1509      1 1 0 0
1510      1 1 0 0
1511      1 1 0 1
1512      1 1 0 1
1513      1 0 0 1

```



a

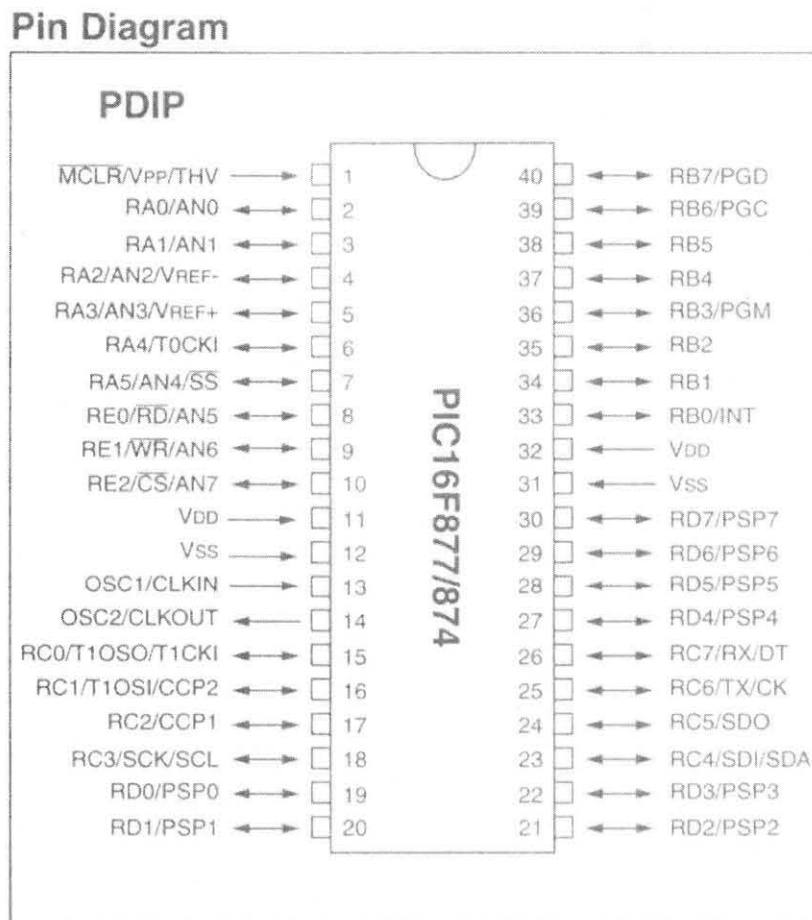
b

Figuur 5.3 Rekenaar gegenereerde trappiesmotorseine soos verkry vanaf 'n logiese analiseerder

5.2 Microchip® PIC-mikrobeheerders

Die Microchip® PIC-mikrobeheerder wat vir die elektroniese koppelvlak gebruik is, is 'n gebruikersvriendelike en 'n ekonomiese produk, wat vryelik beskikbaar is. Die mikrobeheerder wat gebruik is, is die PDIP PIC16F877 wat 'n 40-pen geïntegreerde stroombaan (GS) is. Die PIC16F877, hierna in die hoofstuk verwys as die PIC, beskik oor vyf inset/uitset(I/U)-poorte wat in totaal 33 penne beskikbaar stel vir in- of uitsette. Agt van die penne kan as analooginsette gebruik word, wat dit gerieflik en ekonomies maak om die analoogsensors op die voelpenne te interpreteer, sodat die voelpenstande na die rekenaar versend kan word. Die PIC16F877-mikrobeheerder beskik oor baie ander funksies, wat nie in

hierdie toepassing gebruik word nie [19, p. 1]. Omdat soveel I/U-penne benodig word vir trappiesmotorbeheer, is die PIC16F877 die ideale keuse. Die pen-diagram van die PIC16F877 wat vanaf Microchip®-datavelle verkry is, word in Figuur 5.4 getoon.



Figuur 5.4 Die pen-diagram van die PIC16F877

5.3 Die parallelpoort

Die parallelpoort wat geleë is aan die agterkant van rekenaars in die vorm van 'n D-tipe 25-pen "female" verbinder, laat die lees van 'n maksimum van nege bisse en die skryf van twaalf bisse per geleentheid toe. Die 1994 IEEE 1284 standaard spesifiseer die vyf moontlike modes van werking [3, p. 1] as die:

- "Compatibility"-mode,
- die "Nibble"-mode,
- die "Byte"-mode,
- die EPP-mode (Enhanced parallel port) en
- die ECP-mode (Extended capabilities port).

Die "Compatibility"-mode (ook bekend as die "Centronics"-mode) wat vir die kommunikasie tussen die rekenaar en die beheerkring gebruik is, is aanpasbaar met die standaard-parallelpoort (SPP)-mode, wat bestaan uit vier beheerlyne, vyf statuslyne en agt datalyne. Tabel 5.1 toon die penbeskrywing van die D-tipe 25-pen-verbinder.

Tabel 5.1 Die beheerseine van die standaard-parallelpoort

Pen-nommer	SPP-Sein	Rigting: In/Uit	Register	Hardware omgekeerd
1	Strobe (I)	In/Uit	Beheer	Y
2	Data0	Uit	Data	N
3	Data1	Uit	Data	N
4	Data2	Uit	Data	N
5	Data4	Uit	Data	N
6	Data4	Uit	Data	N
7	Data5	Uit	Data	N

8	Data6	Uit	Data	N
9	Data7	Uit	Data	N
10	Ack(I)	In	Status	N
11	Busy	In	Status	Y
12	Paper-Out/End	In	Status	N
13	Select	In	Status	N
14	Auto-Feed(I)	In/Uit	Beheer	Y
15	Error(I)	In	Status	N
16	Initialize(I)	In/Uit	Beheer	N
17	Printer Sel(I)	In/Uit	Beheer	Y
18-25	Ground	Gnd		

Die SPP-sein wat 'n I in hakies het, is logies omgekeerd. Die spanningsvlakke is met die TTL-standaard aanpasbaar, maar die voorsiening- en ontvangstroomvlakke hang van die rekenaarhardeware af. Om dié rede is dit raadsaam om, ter wille van beskerming van die hardeware, buffers te gebruik, wanneer digitale kommunikasie met 'n parallelpoort gedoen word. Die basis-poortadres van LPT1 lê gewoonlik op adres 378h en die van LPT2 op 278h, maar dit kan wissel. Die rede daarvoor is dat, wanneer die rekenaar aangeskakel word, die aantal beskikbare poorte bepaal word deur die rekenaar BIOS en dan toegewys word aan LPT1, LPT2, LPT3 ens. Indien by adresposisie 3BCh (waar die BIOS begin met poortsporing) 'n parallelpoortkaart gevind word, sal LPT1 aan die adres gekoppel word [3, p. 1]. Die basis-adres verteenwoordig die agt datalyne, basisadres + 1 die statuslyne en basisadres + 2 die beheerlyne. Dit is dus ter wille van ouer rekenars beter om die basis-adres deur sagteware te laat bepaal, as om dit in die sagteware 'n vaste waarde te maak.

5.4 Die koppelvlakborduitleg

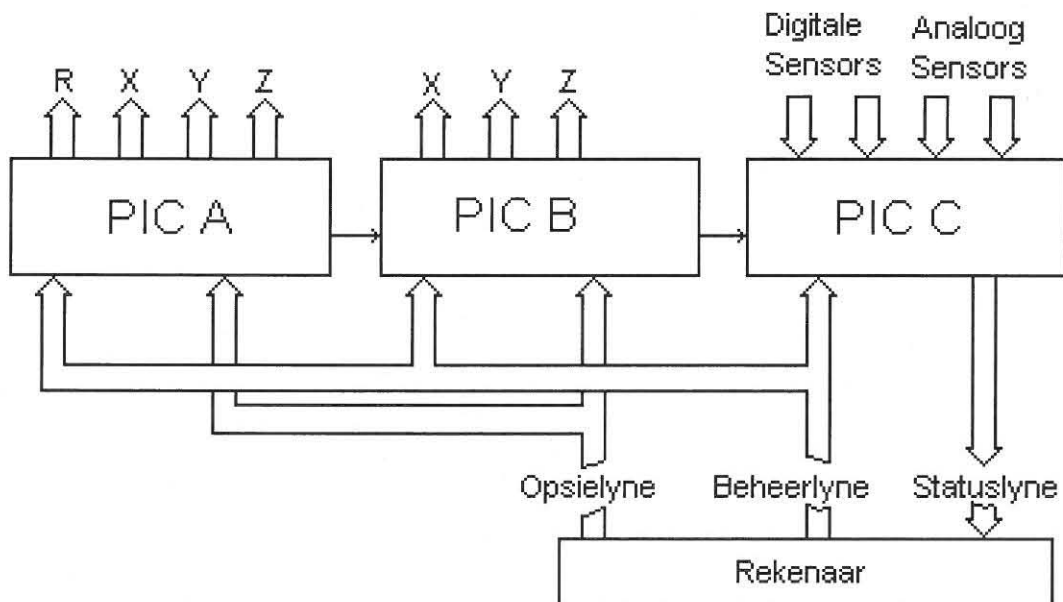
Soos reeds genoem, word twee Max Nc 10-masjiene gelyktydig vanaf 'n enkele rekenaar beheer. Elke masjien het drie trappiesmotore, asook 'n vierde, wat op een van die masjiene gemonteer is vir horisontale rotasie. Elke trappiesmotor benodig vier datalyne vir beheerseine. Verder is daar vir die bepaling van aanvangspunte drie as-sensors op die asse geïnstalleer. Die gebruik van driedimensionele voelpeppe maak dit noodsaaklik dat 'n addisionele vyf datalyne na die rekenaar beskikbaar is, sodat al die moontlike posisies van driedimensionele bewegings gekodeer kan word. Die totale moontlike posisies vir die 3^3 -matriks is 27, wat binêr uitgedruk kan word in 'n minimum van vyf bissyfers. Die totale aantal datalyne benodig, is die volgende:

$$\begin{aligned} &2 \text{ masjiene} \times [3 \text{ trappiesmotore} \times 4 \text{ datalyne} \\ &\quad + 4 \text{ limietsensors} \\ &\quad + 5 \text{ voelpepkodering}] \\ &\quad + 1 \text{ rotasietrappiesmotor} \times 4 \text{ datalyne} \\ &= 46 \text{ datalyne.} \end{aligned}$$

Die noodsaaklikheid vir 'n elektroniese koppelvlak is dus duidelik, aangesien die parallelpoort oor slegs 17 I/U-penne beskik. Multiplexseerders is aanvanklik gebruik om die nodige beheerlyne daar te stel, maar is later uitgefaseer deur sagteware wat programmeerbare

geïntegreerde stroombane (PIC) beheer. Figuur 5.5 toon 'n blokdigram van die elektroniese koppelvlak met sy drie stelle datalyne.

Twee PIC's is verantwoordelik vir die trappiesmotorbeheer, terwyl 'n derde die sensor en ander terugvoering na die rekenaar behartig. Daar is slegs een datalyn vanaf PIC 1 en PIC 2 onderskeidelik na PIC 3 vir kommunikasie tussen PIC's 1 en 3, asook tussen PIC's 2 en 3. Geen beheer-datalyne is aan PIC 3 gekoppel nie. Die rekenaarontwerp van die gedrukte stroombaan kan in Bylaag F gesien word.



Figuur 5.5 Blokdigram van die elektroniese koppelvlak

Seinkodering word gedoen deur die mikrobeheerder se beheerseine vanaf die rekenaar te multiplekseer, terwyl die trappiesmotorseine deur die mikrobeheerder gegenereer word. Die beheerseine is driedimensioneel van aard wat beteken dat dit een van 27 moontlike bewegingsrigtings in die Cartesiese vlak spesifiseer. Die beheerwoord moet dus minstens vyf

datalyne beslaan. Die beheerwoord word ook gebruik om die spoed en resoluksie van die aftasting te spesifiseer. Die funksie van die beheerlyne word deur die opsielyne bepaal. Masjienseleksie, motorspoed-seleksie, resoluksie-seleksie, voelpestandaanvraag en kommunikasiebord-deteksie word deur die opsieregister geselekteer.

Tabel 5.2 toon die verbindingspunte op die D-tipe verbinders tussen die rekenaar, die beheerbord en die masjiene. Die voelpe sensordata word apart van die 25-pen D-tipe-verbinders na die beheerbord gevoer.

Tabel 5.2 Die pen-uitleg van die D-tipe 25-penverbinders tussen die elektroniese koppelvlak, die Max NC-masjiene en die rekenaar

Pennummer	Max Nc-kant	Rekenaar-kant
1	Y0	Opsie 0
2	Z0	Beheer 0
3	Z1	Beheer 1
4	Z2	Beheer 2
5	Z3	Beheer 3
6	X0	Beheer 4
7	X1	Herstel
8	X2	-
9	X3	-
10	X sensor	Status 4
11	Y sensor	Status 3
12	Z sensor	Status 2
13	Voelpe	Status 1
14	Y1	Opsie 1
15	-	Status 0
16	Y2	Opsie 2
17	Y3	Opsie 3
18-21	Grond	Grond
22	R0	Grond
23	R1	Grond
24	R2	Grond
25	R3	Grond

5.5 Die beheerlogika

Die opsie-register wat in Tabel 5.3 uiteengesit word, bepaal watter aksie uitgevoer word. In sommige gevalle word die waarde wat op die beheer-datalyn is, in die betrokke register op die PIC-mikrobeheerder as 'n nuwe parameterwaarde gestoor. In die geval van motorbeweging sal dit egter die kode bevat wat die bewegingsrigting spesifiseer. Die opsie-datalyn kan ook inligting verskaf oor die limietsensors of voelpeestand. In so 'n geval sal die terugvoer via die status-datalyn na die rekenaar geskied. Inligting soos “gereed” of “besig” word ook deur die status-datalyn na die rekenaar gevoer. Die betekenis van die waardes in die status- sowel as die beheer-datalyn wissel na gelang van die waarde in die opsie-datalyn.

Tabel 5.3 Die opsie-registerkode gekoppel aan mikrobeheerder reaksie

Bin- waar de	Hex- waar de	Masjien-A-beheer PIC	Masjien-B-beheer PIC	Terugvoer en sensorbeheer PIC
0000	0	Lees beheerwaarde, voer beweging uit	Passief	Stuur “besig”. Indien gereed, stuur “gereed”
0001	1	Passief	Lees beheerwaarde, voer beweging uit	Stuur “besig”. Indien gereed, stuur “gereed”
0010	2	Lees beheerwaarde, voer beweging uit	Lees beheerwaarde, voer beweging uit	Stuur “besig”. Indien gereed, stuur “gereed”
0011	3	Lees beheerwaarde as nuwe resolusie	Lees beheerwaarde as nuwe resolusie	Stuur “besig”. Indien gereed, stuur “gereed”
0100	4	Lees beheerwaarde as nuwe spoed	Lees beheerwaarde as nuwe spoed	Stuur “besig”. Indien gereed, stuur “gereed”
0101	5	Passief	Passief	Diskonnekteer

0110	6	Passief	Passief	Passief
0111	7	Herstelwaarde vir resoluksie en spoed	Herstelwaarde vir resoluksie en spoed	Hestel analoog sensor sensitiwiteit waarde
1000	8	Passief	Passief	Alle sensorstande ?
1001	9	Passief	Passief	Bepaal geen tas waarde van analoë-sensors
1010	A	Passief	Passief	3D voelpe 1-stand ?
1011	B	Passief	Passief	3D voelpe 2-stand ?
1100	C	Passief	Passief	Limietsensors XYZ en BXY ?
1101	D	Passief	Passief	Limietsensors BXYZ en AXY ?
1110	E	Passief	Passief	Lees waarde as nuwe sensitiwiteit
1111	F	Konnekteer, inisieer registers	Konnekteer, inisieer registers	Konnekteer; inisieer registers

Die eerste kolom van Tabel 5.4 stel die driedimensionele kodering voor. Dit is die beheer-registerkode, wat bepaal in watter rigting die motore sal beweeg tydens waardes 0h, 1h en 2h op die opsie-register. Dit stel verder ook die terugvoerwaarde op die status-register voor tydens waarde Ah en Bh op die opsie-register, wat die driedimensionele voelpestandnavraag is. Vir die opsie-registerwaarde 3h en 4h, wat die resoluksie en spoed bepaal, word die beheerwaarde geïnterpreteer as 'n skaalwaarde tussen 1 en 32. Die status-terugvoerwaardes vir die relevante opsie-registerposisies word ook in Tabel 5.4 uiteengesit. Soos reeds genoem, geld die kode in die eerste kolom ook vir die beheer-registerwaardes tydens opsiewaarde 0h, 1h en 2h.

Tabel 5.4 Die status-register kodering

Binêre- waarde	Hex- waarde	Betekenis tydens opsie Ah en Bh	Betekenis tydens opsie 0h, 1h, 4h
00000	00	Z+, Y+	Geen betekenis
00001	01	Z+, X+, Y+	Geen betekenis
00010	02	Z+, X+, Y-	Geen betekenis
00011	03	Z+, X+	Geen betekenis
00100	04	Z+	Geen betekenis
00101	05	Z+, X-	Geen betekenis
00110	06	Z+, X-, Y+	Geen betekenis
00111	07	Z+, X-, Y-	Geen betekenis
01000	08	Z+, Y-	Geen betekenis
01001	09	Y+	Geen betekenis
01010	0A	X+, Y+	Geen betekenis
01011	0B	X+, Y-	Geen betekenis
01100	0C	X+	Geen betekenis
01101	0D	Geen beweging	Geen betekenis
01110	0E	X-	Geen betekenis
01111	0F	X-, Y+	Geen betekenis
10000	10	X-, Y-	Geen betekenis
10001	11	Y-	Geen betekenis
10010	12	Z-, Y+	Geen betekenis
10011	14	Z-, X+, Y+	Geen betekenis
10100	14	Z-, X+, Y-	Geen betekenis
10101	15	Z-, X+	Geen betekenis
10110	16	Z-,	Geen betekenis
10111	17	Z-, X-	Geen betekenis
11000	18	Z-, X-, Y-	Geen betekenis
11001	19	Z-, X-, Y+	Geen betekenis
11010	1A	Z-, Y+	Geen betekenis
11011	1B	Geen betekenis	Sensor geraak
11100	1C	Geen betekenis	Geen betekenis
11101	1D	Geen betekenis	Geen betekenis
11110	1E	Geen betekenis	Besig
11111	1F	Geen betekenis	Gereed

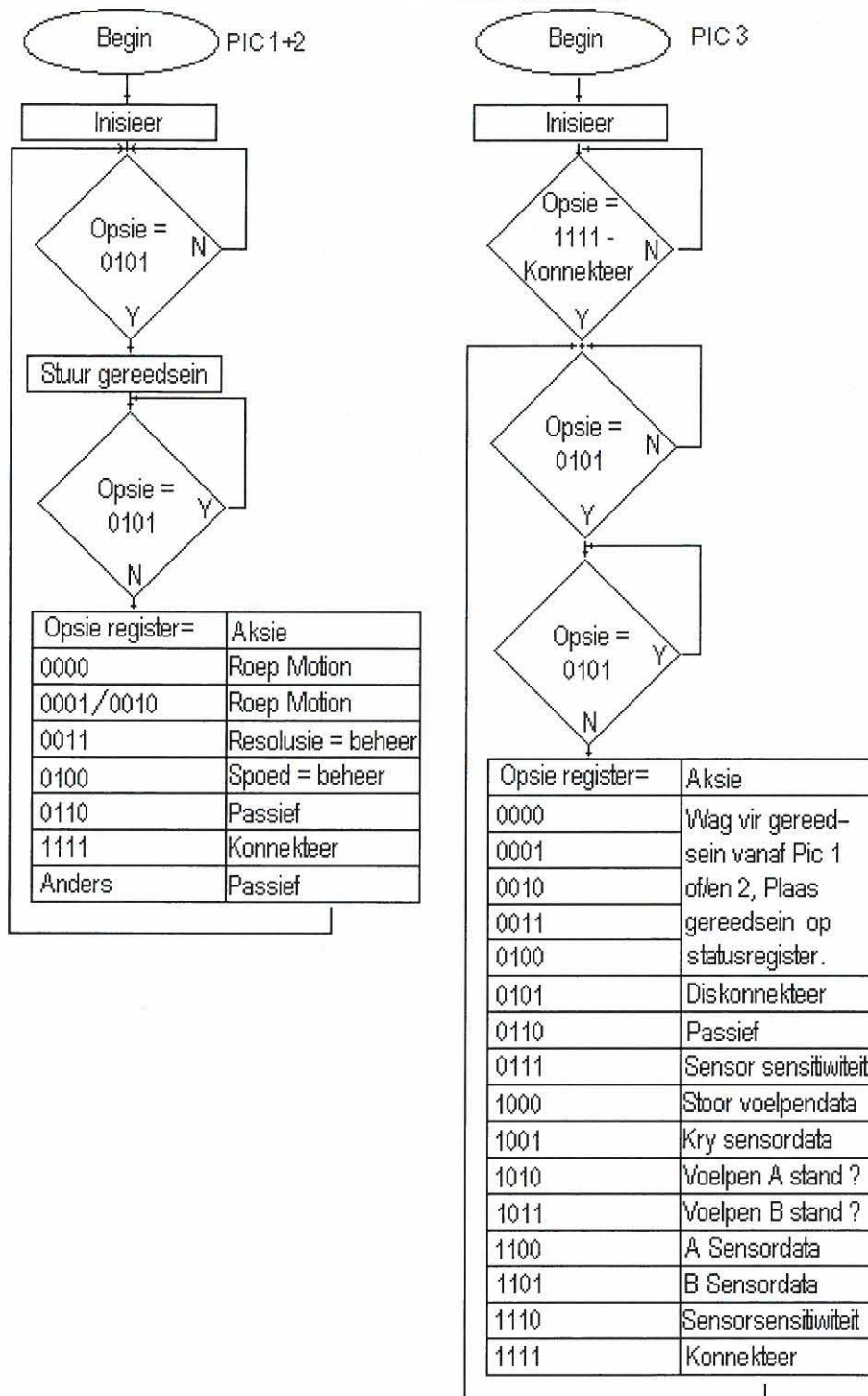
Die kodering van die status-register om die sensorstande te bepaal (opsie-registerwaarde Ch of Dh) word in Tabel 5.5 uiteengesit.

Tabel 5.5 Uiteensetting van sensorvertolking op die status-register

Opsie	Beskrywing	S0	S1	S2	S3	S4
7h	Alle sensor stande	Lim X, Y en X	Voel-pen A	Lim X, Y en Z	Voel-pen B	Nie gebruik
Ch	Masjien A sensors	Sensor X-Lim	Sensor Y-Lim	Sensor Z-Lim	Voel-pen A	Alle B sensors
Dh	Masjien B sensors	Sensor X-Lim	Sensor Y-Lim	Sensor Z-Lim	Voel-pen B	Alle A sensors

5.6 Mikrobeheerdersagteware

Die mikrobeheerdersagteware is in C geskryf, wat die lees en skryf van komplekse sagteware eenvoudig maak in vergelyking met die PIC-saamsteltaal [5, pp 33-39][19, p. 152]. Die vereenvoudigde vloei-diagramme van die “Main”-roetines van die mikrobeheerdersagteware word in Figuur 5.5 getoon. Die “Main”-roetines van die onderskeie PIC’s monitor die opsie-datalyn en voer die nodige roetines uit volgens die waarde wat ontvang word. Die sagteware van die motorbeheer PIC’s (PIC 1 en 2) wat die trappiesmotorkodes genereer, is so te sê identies. Figuur 5.5a toon die vloei-diagram van die “Main”-roetine. Die vloei-diagram van die “Main”-roetine van die derde PIC, wat alle terugvoering na die rekenaar behartig, word in Figuur 5.5b getoon. Die terugvoering sluit die sensors, voelpenne, “besig” en “gereed” seine in. Die volgende twee afdelings beskryf die belangrikste roetines van die mikrobeheerdersagteware.



Figuur 5.6 Vloeiagramme van die twee "Main"-roetines van die mikrobeheerdersagteware

5.6.1 Motorbeheersagteware (PIC 1 en 2)

Die twee motorbeheer-PIC's reageer slegs op vyf verskillende opsie-dataseine. Resolusie en spoed word bepaal deur die waarde op die beheer-datalyn, wanneer die waardes 3h en 4h op die opsie-datalyn verskyn.

Die “beweeg”-kodewaardes 0h en 2h in die geval van PIC 1 en 1h en 2h in die geval van PIC 2 inisieer die “Motion”-roetine. Die “Motion”-roetine inisieer op sy beurt weer een van 29 ander roetines, afhangende van die waarde op die beheer-datalyn. Een van die roetines word getoon in Kodelys 5.1. Let op hoe die resolusie en spoedparameters in die roetine geïmplimenteer word.

```
void DwnB(int Resl,int Spd)
{int i;
for (i=1 ;i <= (2*Resl);i++)
{Out('Y',Spd);
Out('Z',Spd);
Out('X',Spd);
delay(Spd);}
}
```

Kodelys 5.1 Een van 29 roetines wat die trappiesmotor-bewegingsrigtings

beheer

Die roetines roep op hul beurt weer 1 of 2 ander roetines op, afhange van die rigting van die driedimensionele beweging. Kodelys 5.2 toon 'n deel van een van hierdie roetines. Dit is die roetines wat fisies die trappiesmotor-kodes genereer. Die bewegingsrigtings word deur twee roetines naamlik "In" en "Out" verwesenlik. Die bestemde as vir beweging asook die spoed waarteen beweging moet plaasvind, word bepaal deur twee parameters nl. PortNr (wat 'X', 'Y', 'Z' of 'R' kan wees) en Spd ('n waarde tussen 1 en 31).

```
void Out(char PortNr,int Spd)
{switch (PortNr)
{   case 'X': BIT_CLEAR(PORTD,4);
        delay(Spd);
        BIT_SET(PORTD,2);
        delay(Spd);
        BIT_CLEAR(PORTD,0);
        delay(Spd);
        BIT_SET(PORTD,4);
        delay(Spd);
        BIT_CLEAR(PORTD,1);
        delay(Spd);
        BIT_SET(PORTD,0);
        delay(Spd);
        BIT_CLEAR(PORTD,2);
        delay(Spd);
        BIT_SET(PORTD,1);
        break;
    case 'Y':
```

Kodelys 5.2 Een van twee soortgelyke roetines wat trappiesmotorkodes genereer

Wanneer die beweging voltooi is, word 'n sein na die derde PIC gestuur, wat op sy beurt die “gereed” sein na die rekenaar stuur.

Die “delay”-roetine maak die regte tydsvertraging tussen die trappiesmotor-kodes moontlik, terwyl die “Initiate”-roetine die poortopstelling doen.

5.6.2 Beheersagteware vir voelpen- en sensorterugvoering (PIC 3)

Voordat die elektroniese koppelvlak aktief raak, moet die “konnekteer”-sein vanaf die rekenaar ontvang word. Die derde PIC sal in 'n lus bly, totdat dié kode ontvang word, voordat enige verdere terugvoering kan plaasvind. Op hierdie wyse sal die rekenaar weet dat die elektroniese koppelvlak aanwesig is. Soos gesien kan word aan sy “Main”-roetine in Figuur 5.6, reageer die derde PIC op bykans al die kodes van die opsie-datalyn. Tydens kodes 0 tot 4 word daar gewag op die sein van die motorbeheer PIC's, voordat die “gereed”-sein na die rekenaar gestuur word. Die analoogsensorkodering word deur die interne A/D-omskakelaars gedoen. Die “geen raak”-posisie word bepaal tydens inisiasie van die voelpen. Die rigting van die defleksie word bepaal deur die “geen raak”-waarde te vergelyk met 'n nuwe waarde van die voelpen. Na die vergelyking word die defleksie-rigtingskodering (getoon in Tabel 5.4) gedoen, sodat data op die status-datalyn versend kan word. Die roetines wat die A/D-omskakeling doen, word “AdnewA” en “AdnewB” genoem.

Indien die nuwe waarde meer afwyk as wat die sensitiwiteit toelaat, word die rigting van die afwyking bepaal deur roetines “RVALA” en “RVALB”. Die kodering in die status-woord word in roetine “CVAL” gedoen. Kodelys 5.3 toon ‘n gedeelte van die “Main”-roetine, vanwaar al die betrokke roetines geroep word.

```
case 0x09: {BIT_CLEAR(PORTC,5);
            ADmem();                //get no touch states off
            PORTC = 0x1f;
            goto Ack;}             //analog sensors
case 0x0a: {BIT_CLEAR(PORTC,5);
            ADnewA();              //get I-Probe 1 state
            XD = ABS(XAN-XAM);
            YD = ABS(YAN-YAM);
            ZD = ABS(ZAN-ZAM);
            if (XD > Sens) RVALA('X');
            else XDif = 0;
            if (YD > Sens) RVALA('Y');
            else YDif = 0;
            if (ZD > Sens) RVALA('Z');
            else ZDif = 0;
            PORTC = CVAL(XDif, YDif, ZDif);
            break;
            goto Ack1;}
case 0x0b: {BIT_CLEAR(PORTC,5);
            ADnewB();              //get I-Probe 2 state
```

Kodelys 5.3 Roetines betrokke by voelpenkodering

Kodelys 5.3 is ‘n gedeelte van die “case”-roetine en toon die geval van opsie-waarde 09, 0A en 0B. Geval 09 roep die “Admem”-roetine op, wat die "geen raak"-waardes van die voelpenne in die mikrobeheerder se geheue stoor. Dit is met hierdie waardes wat die nuwe waardes, verkry in

die “AdnewA” en “AdnewB”-roetines, vergelyk word. Geval 0A en 0B is die sensorstandnavraag-roetines wat reeds beskryf is.

5.7 Opsomming

Deur gebruik te maak van drie PIC16F877 mikrobeheerders kon ‘n elektroniese koppelvlak daargestel word, wat twee Max NC 10-CMM’s beheer, asook om tasrigtings bewuste voelpenne en limietsensors te akkommodeer. Die trappiesmotorseine wat deur die PIC’s behartig word, verminder verder die werkklas van die rekenaar, sodat die proseseringstyd gewy kan word aan grafikaroetines. Die elektroniese koppelvlak tree ook op as ‘n anti-kopieerprop, sodat die sagteware nie net bloot gekopieer kan word nie.

REKENAARSAGTEWARE

In hierdie hoofstuk word die Reverse3D-sagteware, wat geskryf is, verduidelik en word veral klem gelê op die prosedures wat gevolg moes word om model-aftasting en datavergelyking te doen. Die sagteware is vir die Windows 95 en -98 platforms geskryf en is dus in die grafiese gebruiker-koppelvlak (GUI) formaat. Die programmeertaal waarin die sagteware geskryf is, is Microsoft Visual Basic V 6.0. Die programmatuur word in Hoofstuk 7 bespreek.

6.1 Installering van die sagteware

Die sagteware word geïnstalleer deur die “Setup”-ikoon deur middel van die muis te aktiveer. Die installasie-roetine soek eenstens na die teenwoordigheid van die nodige selflaailêers, sodat dit geïnstalleer of op datum gebring kan word. Die opstel-lêers, bestaande uit die dinamiesebinding-programmateek (*dll*) (gestoor in die Windows\System gids) en die Reverse3D-sagteware (gestoor in ‘n gids, gespesifiseer deur die gebruiker) word hierna geïnstalleer. Die program kan geaktiveer word vanuit “Programs”- Hvs3D – Reverse3D onder die "Start"-menu in Windows 95/98.

6.2 Inisiasie

Wanneer die Reverse3D-sagteware in werking tree, is van die eerste take wat verrig moet word ten einde die sagteware bruikbaar te maak, die opsporing van die elektroniese koppelvlak. Dit word gedoen deur die enigste aktiewe hierargiese menu-item, naamlik “Connect” onder die “Options”-menu (getoon in Figuur 6.1a) te selekteer. Indien die hardeware nie gevind word nie, sal die boodskap dat die poging misluk het, vertoon word. Slegs indien die elektroniese koppelvlak teenwoordig is, sal die menu’s geaktiveer word om die sagteware bruikbaar te maak. Die sagteware is dus nutteloos by die afwesigheid van die elektroniese koppelvlak. Hierna word die program-opsies (soos dit geselekteerd was tydens die vorige werksessie) na die elektroniese koppelvlak versend.

Wanneer daar suksesvolle kommunikasie bewerkstellig is, sal die sagteware voortgaan om die aanvangspunte van die twee Max NC 10-masjiene te bepaal. Nadat die aanvangspunte bereik is, beweeg die asse na hul onderskeie middelpunte. Hierna kan voortgegaan word om die sagteware te gebruik.

6.3 Die menu

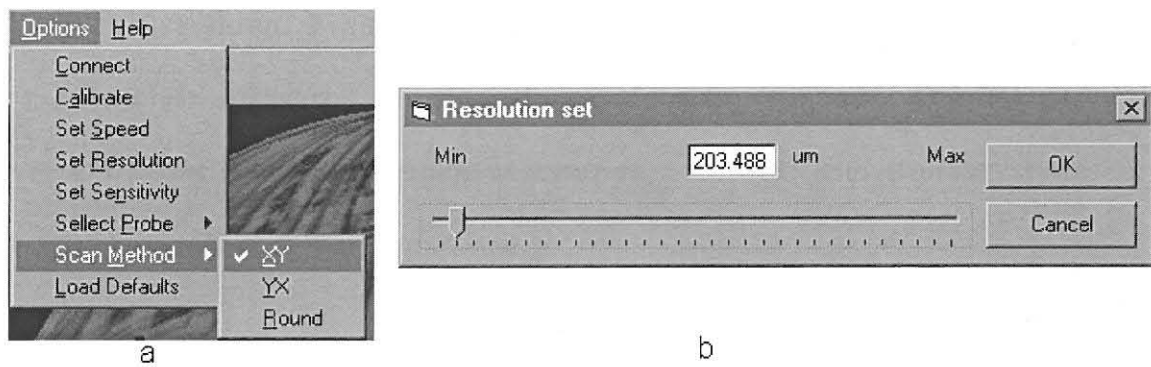
Die menu is in die standaard formaat vir Windows-sagteware uitgelê ter wille van gebruikersvriendelikheid vir diegene wat daarmee vertrou is. Algemene-menu-items wat in die sagteware voorkom, is “File”, “View” en “Help. Die items werk op dieselfde wyse as enige ander GUI-sagteware en bespreking daarvoor is onnodig. Vier ander menu-items, eie aan die Reverse3D-sagteware, is ingesluit en heet “Options” , “Jog”, “Copy” en “Scan”.

6.3.1 “Options”

Die “Options”-menu-item met sy onderafdelings verskyn in Figuur 6.1a. Die laaste ses items word gebruik om verskillende opsies vir aftasting te selekteer word gedoen deur aftekenmerke langs die sekondêre aksies te maak, of deur ‘n beheerraampie op te roep.

Figuur 6.1b toon ‘n voorbeeld van die beheerraampie vir resoluieverstelling en is soortgelyk aan die spoed- en sensitiwiteit-verstellings.

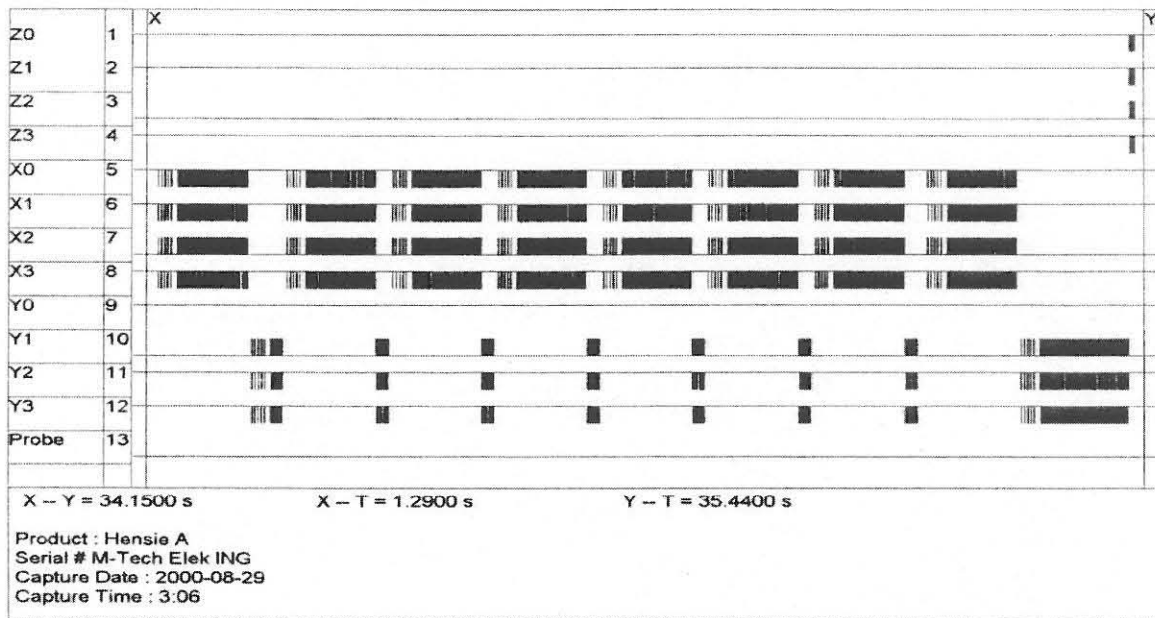
Die “Connect”-item, in Hoofstuk 6.3 bespreek, is verantwoordelik vir die koppeling met die hardeware en is een van die enigste menu-items wat aktief is, nadat die sagteware in werking gestel is.



Figuur 6.1 Die “Options”-menu uitgerol en resolusiebeheerraampie

‘n Meganiese probleem wat ondervind is met die Max NC-masjiene, en gevolglik in die sagteware gekorrigeer moes word, is dat daar bewegingstraagheid op die asse is, wanneer die trappiesmotore van rotasierigting verander. Die oorsaak hiervoor is die glybaan-en-skroef-aandrywingsmeganisme (in Figuur 2.6 getoon) wat vir die asse gebruik is. As gevolg hiervan is die geometriese koördinate, wat in die datalêer verskyn, óf meer of minder is as die werklike waarde, afhangende van die bewegingsrigting. Dit was dus nodig om kalibrasie van die asse te doen, deur te bepaal wat hierdie traagheid is, en om ‘n aantal bykomende stappe, gelykstaande aan die traagheid, te neem wanneer die motore van rigting verander. Nog ‘n faktor wat bydra tot die noodsaaklikheid vir kalibrasie van die motore, is voelpenne wat soms nie omnidireksioneel dieselfde sensitiwiteit het nie. Indien die voelpenposisie met ‘n afstand gelykstaande aan die voelpentraagheid (vir die betrokke defleksierigting) verskuif word, kan hierdie probleem geëlimineer word. Hierdie korreksie

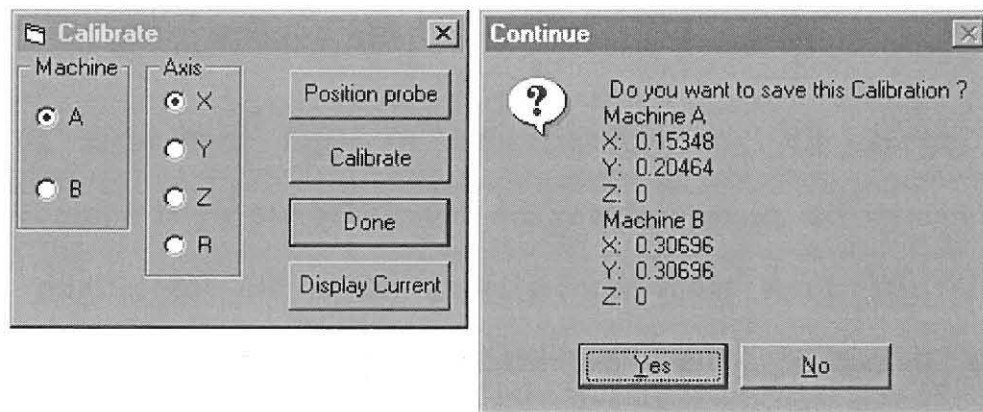
word in toevoeging tot die voelendiktekompensasie (in Hoofstuk 3.1 bespreek) gedoen. Figuur 6.2 is die seindiagram van modelaftasting, oor 'n baie klein aftasarea om die bykomende stappe te toon. Die dun strepies verteenwoordig die kompensasiestappe, terwyl die donker lyne die werklike beweging verteenwoordig.



Figuur 6.2 Traagheidskompensasie vir die trappiesmotore

Koördinaat-kalibrasie word gedoen deur die “Calibrate”-aksie te selekteer. In Hooftuk 2.1.1 is die Renishaw® Cyclone bespreek en is daar onder andere van die datumsfeer wat vir kalibrasiedoeleindes gebruik word, melding gemaak. Die afwesigheid van 'n datumsfeer by die Max NC 10 het daartoe aanleiding gegee dat alternatiewe metodes gebruik moes

word om kalibrasie te bewerkstellig. Figuur 6.3 toon die beheervenster wat in reaksie op die “Calibrate”-menu-item gelaai word asook die “Continue”-beheerraampie wat in reaksie op die “Done”-drukknop gelaai word. Die kalibrasie word gedoen deur in die rigting van die voorwerp te beweeg totdat aanraking waargeneem word. Die motor beweeg dan ‘n verdere 10 stappe, voordat dit in die teenoorgestelde rigting beweeg word. Die aantal stappe gegee tussen die geentas-kondisies, is die aantal korreksiestappe wat geneem moet word, wanneer die bewegingsrigting verander.



Figuur 6.3 Die beheerraampies betrokke by kalibrasie van die trappiesmotore

Van die gebruiker word verwag om die as vir kalibrasie te selekteer en dan die voelpe te posisioneer op ‘n sekere posisie vanaf ‘n voorwerp. Dit word gedoen deur die “Position Probe”-drukknop in die raampie te selekteer en die daaropvolgende instruksies te volg. Wanneer al die asse gekalibreer is, word die “Done”-drukknop geselekteer. Dit sal die

“Continue”-inligtingsraampie met die onderskeie kalibrasies vertoon, sodat dit verwerp of aanvaar kan word in reaksie op die gebruiker-terugvoering. Kalibrasie van die sagteware, sowel as van die hardeware, moet gereeld gedoen word - áltyd wanneer voelpenne geruil word. Kalibrasie van die hardeware behels dat die blad-klemskroewe styf genoeg vasgedraai word, sodat speling op die blad beperk is, maar beweging steeds gemaklik kan geskied.

6.3.2 “Jog”

Die “Jog”-menu-item roep ‘n beheerraampie na die skerm wat verantwoordelik is vir die beheer van die voelpenposisie, bv. wanneer die voelpen relatief tot die model geposisioneer moet word. Die “Jog”-beheerraampie kan ook gerieflikheidshalwe vanuit ‘n aantal ander beheerraampies opgeroep word. Figuur 6.4 toon die skermaftreksel van die “Jog”-beheerraampie. Sy werking is voor die hand liggend, indien na sy beheeritems gekyk word.



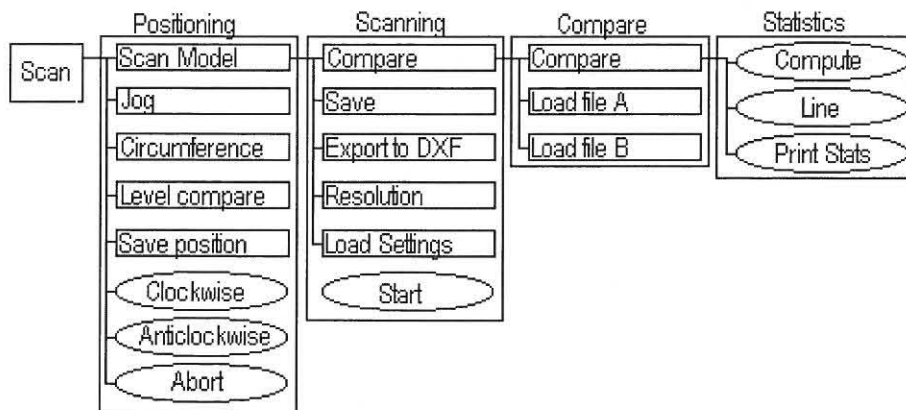
Figuur 6.4 Die “Jog”-beheerraampie

6.3.3 “Copy”

Die “Copy”-menu-item word gebruik om ‘n model te masjineer. Van die gebruiker word verwag om die korrekte beitel of boor op die masjien aan te bring, aangesien daar geen magasynfasiliteite op die Max NC-10 is nie. Dié menu-item voorsien die gebruiker van twee opsies, naamlik modelmasjinerig vanaf ‘n dataleêr (afgetas op ‘n vroeër stadiuim), asook modelkopiëring vanaf ‘n modelprofiel. In laasgenoemde geval word aftasting op een masjien gedoen, terwyl die ander die model slaafs masjineer. Die beheerraampies wat in reaksie op die respektiewe seleksies getoon word, beskik ook oor die nodige kontroles om skaalveranderings op die kopie toe te laat.

6.3.4 “Scan”

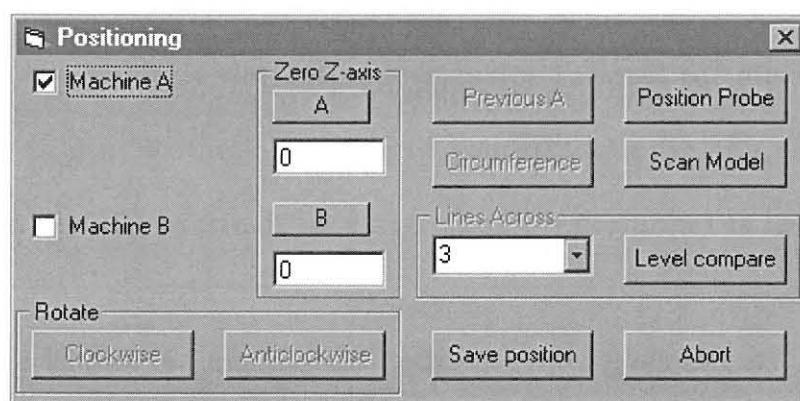
In Hoofstuk 3.3 is die model-oriëntasie, indien datavergelyking beoog word, bespreek. Om twee stelle data te kan vergelyk, moet die modelle voor aftasting relatief tot mekaar georiënteer word. Om hierdie rede roep die “Scan”-menu-item eers die beheerraampie wat model-oriëntasie beheer na die skerm. Geen model-oriëntasie is verpligtend nie en die gebruiker kan die opsie uitoefen om direk voort te gaan met modelskandering. Aftasting word vanuit die “Scan Options”-beheerraampie gedoen, wat op sy beurt vanuit die oriëntasie-beheerraampie geïnisieer word. Om data van twee modelle te vergelyk, word ‘n beheerraampie op sy beurt vanuit die “Scan Options”-beheerraampie geroep. Die skakelingshiërargie van die beheerraampies word in Figuur 6.5 getoon. Die groot vierkante stel die hoofbeheerraampies voor en die kleiner vierkante die sekondêre beheerraampies, wat vanuit die hoofbeheerraampies geïnisieer word. Die ovale stel die drukkoppe voor, wat sekere funksies verrig, sonder om beheerraampies te gebruik.



Figuur 6.5 Die skakelingshiërargie van beheerraampies onder die “Scan”-menu-item

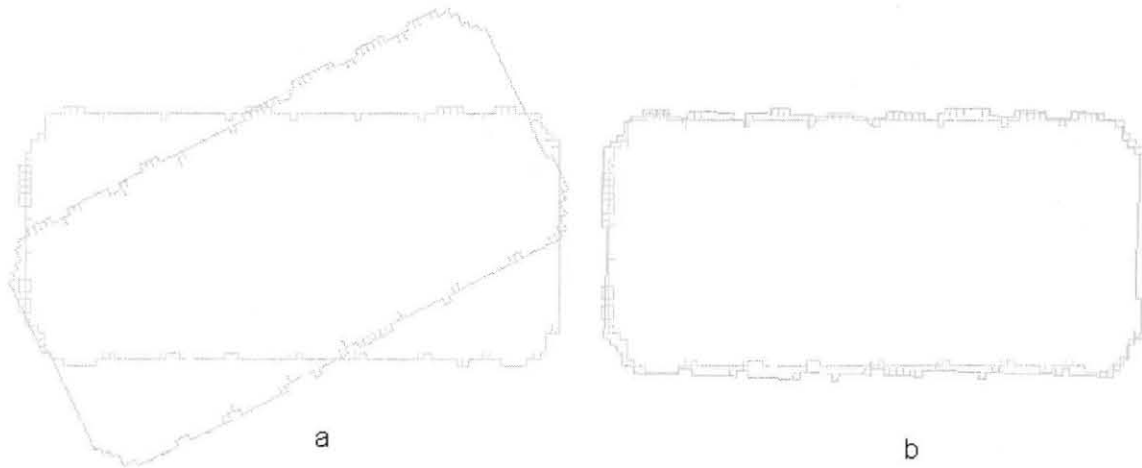
6.4 Die skandering en vergelyking van modelle in Reverse3D

Figuur 6.6 is die skermaftreksel van die “Positioning”-beheerraampie. Soos baie van die ander kontroleraampies is al die funksies nie vanuit die staanspoor aktief nie en maak dit staat op voorafgaande gebeure, wat vereis word om korrek te funksioneer.



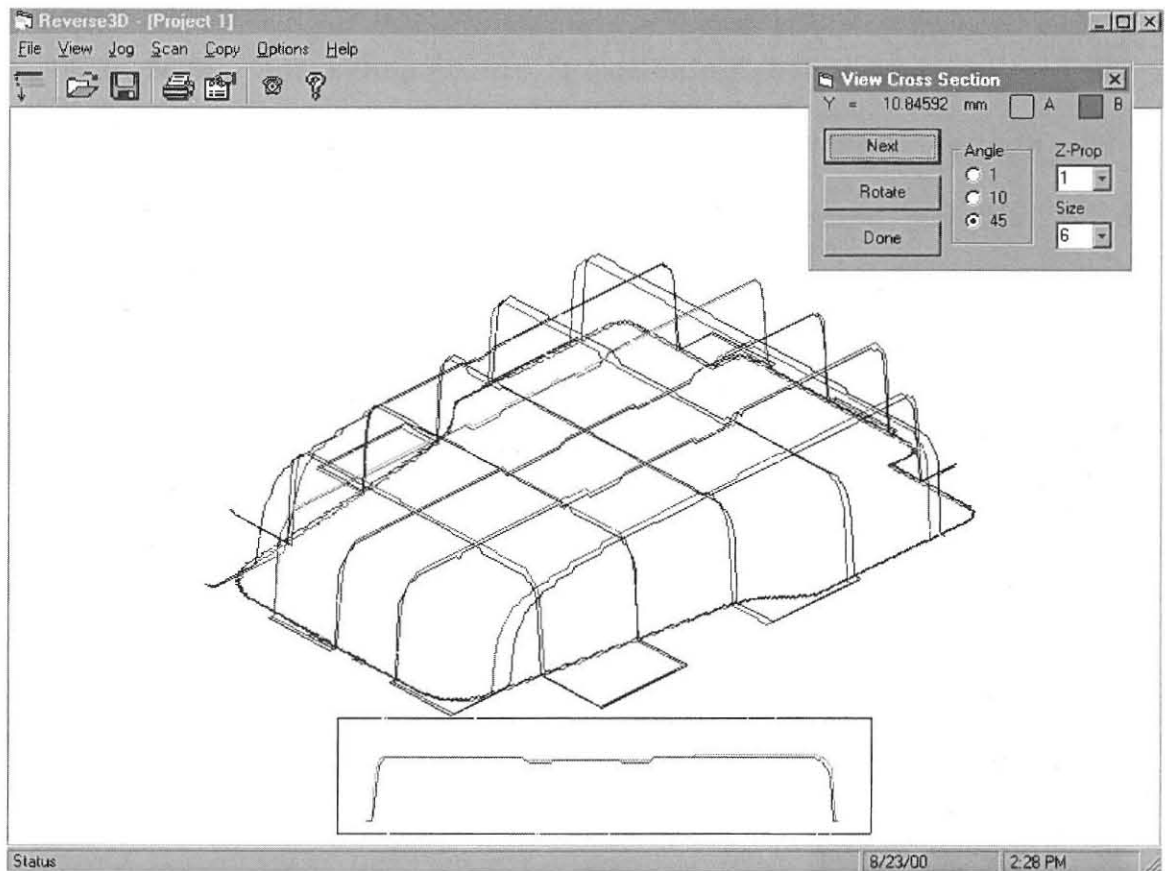
Figuur 6.6 Die “Positioning”-beheerraampie

Voordat omtrekaftasting gedoen kan word, moet die voelpenne bokant die modelle in posisie gebring word deur middel van die “Position Probe”-drukknop wat die “Jog”-beheerraampie vertoon. Die diepte waarteen die aftasting gedoen word, word deur die gebruiker gespesifiseer deur “Zero”-drukknoppe wat teenwoordig is in die “Jog”- sowel as die “Positioning”-beheerraampie. Wanneer die “Circumference”-drukknop gedruk word, word die gebruiker ingelig van die resolusie-waarde waarteen aftasting gedoen gaan word, sodat (indien dit onaanvaarbaar is) dit verander kan word voordat die proses in aanvang neem. Wanneer met aftasting begin word, sal die voelpen afwaarts beweeg, totdat die model gevind word. Daarna sal die voelpen na die modelrand op die gekose diepte begin soek. Indien geen aanraking gevind word voordat die minimum diepte bereik is nie, sal die proses gestop word en die gebruiker ingelig word van die fout. Wanneer die aftasdiepte bereik is, sal die omtrekaftasting gedoen word, totdat die eind-koördinaat die begin-koördinaat bereik. Figuur 6.7 toon ‘n lae resolusie omtrekaftasting van twee modelle. In Figuur 6.7a kan gesien word hoe twee modelle verskillende rotasies relatief tot mekaar het. Deur die knoppe in die “Rotate”-raampie te gebruik, kan die een model geroteer word, sodat die silhouette ooreenstem soos dit die geval is in Figuur 6.7b.



Figuur 6.7 Omtreksilhoeët van modelle soos vertoon op die rekenarskerm

Nadat die modelle georiënteer is in die Z-as moet die X- en Y-as-rotasies tot mekaar gerig word, indien nodig. Die ‘Level Compare’-drukknop en lyntal word aktief, nadat die omtrekaftasting gedoen is. Die lyntal bepaal die aantal lyne in die X- en Y-rigtings wat afgetas moet word, ten einde die gelykheid of ongelykheid van die modelle visueel teenoor mekaar te kan bevestig. Die proses kan herhaal word, totdat die model-posisies bevredigend ooreenstem. Hierna kan voortgegaan word met aftasting. Figuur 6.8 toon die “View Cross Section”-beheerraampie met die beeld van die modelle wat ná posisiebepaling gegenereer word, sodat daar bevestig kan word of hulle relatiewe posisies in die Cartesiese vlak met mekaar ooreenstem.



Figuur 6.8 Die beelde van die twee modelle om posisionele gelykheid te bevestig

Die vierkant onderaan die skerm toon die sy-aansig van die individuele afgetaste lyne, sodat elke lyn afsonderlik besigtig word. Lyne word beskou deur die “Next”-drukknop in die beheerraampie te druk. Deur middel van hierdie hulpmiddels word die vlak waarop een model lê, verstel, ten einde te verseker dat beide op dieselfde relatiewe posiesie lê. Daar moet egter tydens oriëntasie na die modelle as ‘n geheel gekyk word, omdat verwag kan word dat daar sekere fisiese verskille op die modelle teenwoordig mag wees.

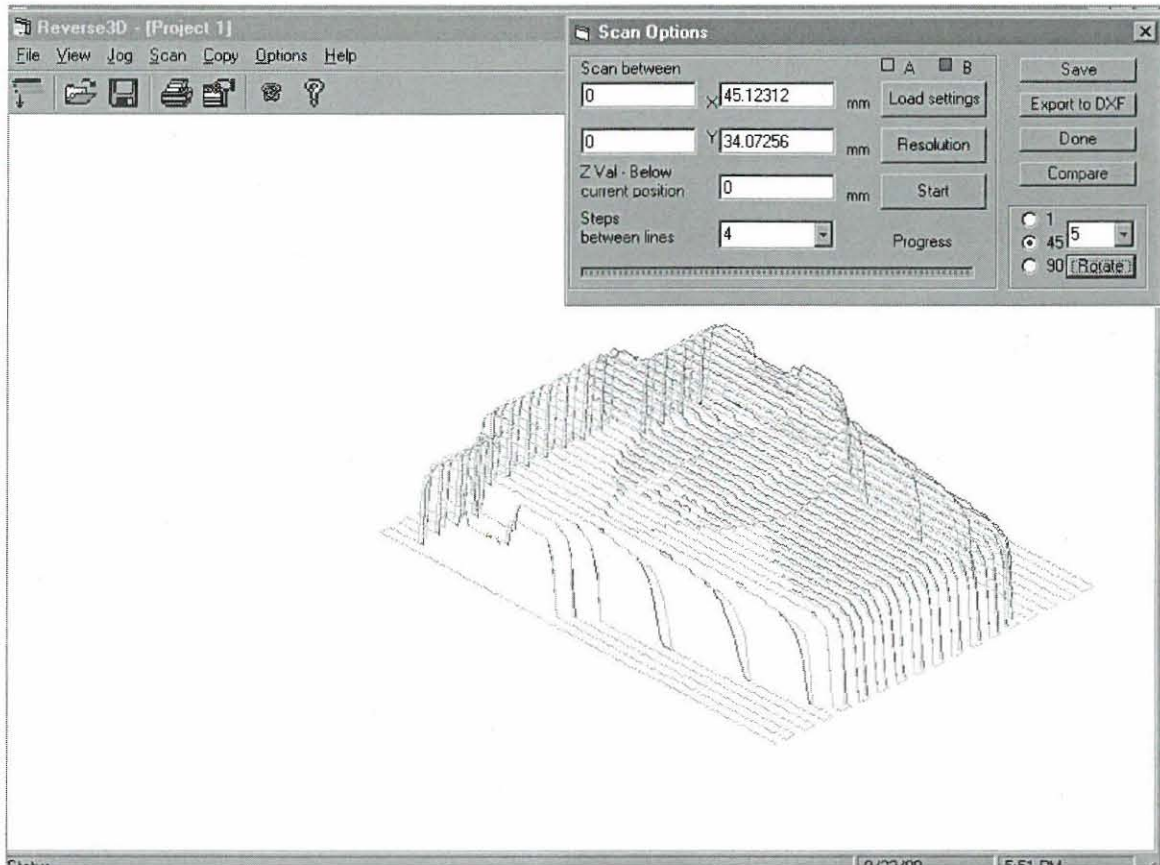
Indien die posisies ooreenstem, kan voortgegaan word met aftasting, deur die “Scan Model”-drukknop in die “Positioning”-beheerraampie te druk. Dit sal op sy beurt die “Scan Options”-beheerraampie vertoon. Die skermafbeelding van ‘n model wat geskandeer is, tesame met die “Scan Options”-beheerraampie, word in Figuur 6.9 getoon. Indien posisionele vergelyking gedoen is, sal die aftasdimensies reeds in die respektiewelike spasies ingevul wees, maar dit kan ook verander word indien nodig.

Die ander opsies wat geselekteer word, sluit in: die afstand tussen die parallelle lyne, diepte, en resoluksie. Daar word ook voorsiening gemaak vir die laai van opsies vanaf vorige aftastings.

Indien aftasting van ‘n model gedoen word met die oog op datavergelyking met modeldata wat gestoor word, is dit nodig om presies dieselfde opsies te selekteer as dié van die rekorddata. Indien nie aan die vereistes voldoen word nie sal die aantal datapunte verskil en die vergelyking nie deur die sagteware toegelaat word nie. Die data word deur die “Save”-drukknop in *hvs*-lêerformaat gestoor, maar kan in die *dxf*-lêerformaat gestoor word deur die “Export”-drukknop te gebruik.

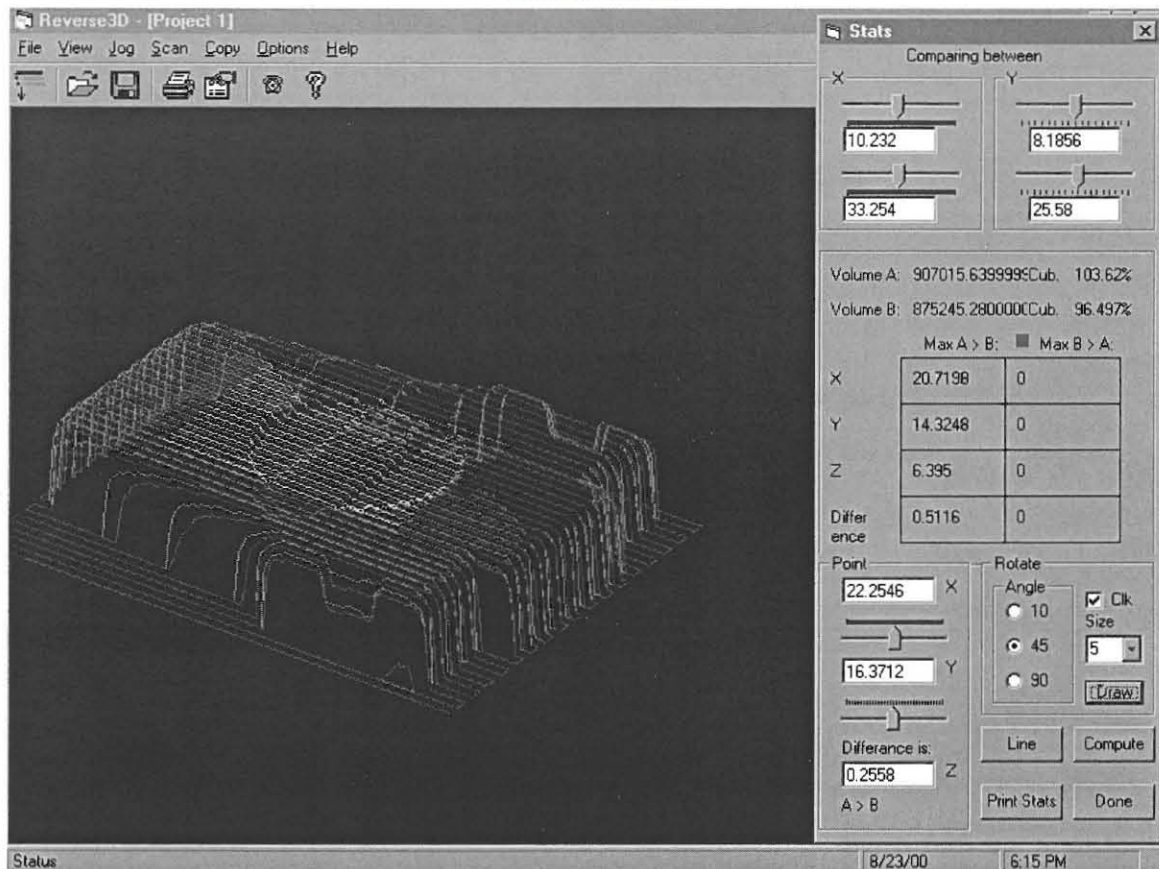
Om data van modelle te vergelyk is daar ‘n kortpad drukknoop, naamlik “Compare” wat die “Compare”-beheerraampie vertoon. Die “Compare”-beheerraampie is verantwoordelik vir die laai en toets van die data-lêers.

Indien die datatoets aandui dat die data met dieselfde opsies opgeneem is, sal die “Statistics”-beheerraampie vertoon word, wat die data met mekaar vergelyk. Figuur 6.10 toon ‘n voorbeeld van modelvergelyking.



Figuur 6.9 Die “Scan Options”-beheerraampie met die beeld van die afgetaste modelle

Datavergelyking kan geskied deur gebruik te maak van al die data-koördinate, of slegs die data tussen sekere limiete te gebruik. Die maksimum afwyking in die Z-as, asook die volume- en persentasie-afwyking word bereken. Data kan ook lyn vir lyn bestudeer en vergelyk word.



Figuur 6.10 Die vergelyking van modeldata

6.5 Opsomming

Die uitvoering van die vergelykingstegnieke (bespreek in Hoofstuk 3) is in hierdie hoofstuk verduidelik, asook die werking van al die beheerraampies betrokke by model-oriëntasie en -aftasting. Daar is ook getoon hoe die sagteware kompenseer vir van die tekortkominge van die Max NC CMM-masjien. Funksies wat algemeen voorkom by die meeste CAD-sagteware, is egter nie bespreek nie.

REKENAARPROGRAMMATUUR

In hierdie hoofstuk word die sleutelmodules van die sagteware wat verband hou met data-aftasting en -vergelyking, oppervlakkig bespreek. 'n Bespreking van die programkode is vir die doel van hierdie studie onodig. Die programmeertaal wat gebruik is, is die objek-georiënteerde Visual Basic 6. Die betrokke dinamiesebinding-programmateek (*dll*-lêers) wat gebruik is, word ook kortliks bespreek.

7.1 Visual Basic 6 en Borland C++ 5

Met die aanvang van die navorsing is verskeie persone geraadpleeg oor juis watter programmeertaal die geskikste sou wees vir die skryf van die rekenaarsagteware. Soos dit by die skrywers van verskeie rekenaartaalboeke wat bestudeer is die geval is, was elkeen daarvan oortuig dat die taal waarin hý of sý programmeer, die alfa en die omega is. Daarom is besluit om 'n studie te loods oor rekenaartale, ten einde die geskikste taal vir die navorsing te identifiseer. Navorsing is gedoen op die Delphi, C++ Builder, Borland C++ 5 en Visual Basic 6. Dit het geblyk dat Delphi, wat op die Pascal-rekenaartaal gebaseer is, nie dieselfde aanhang het as ander tale nie en, dat dit sekere beperkinge het wat nie teenwoordig

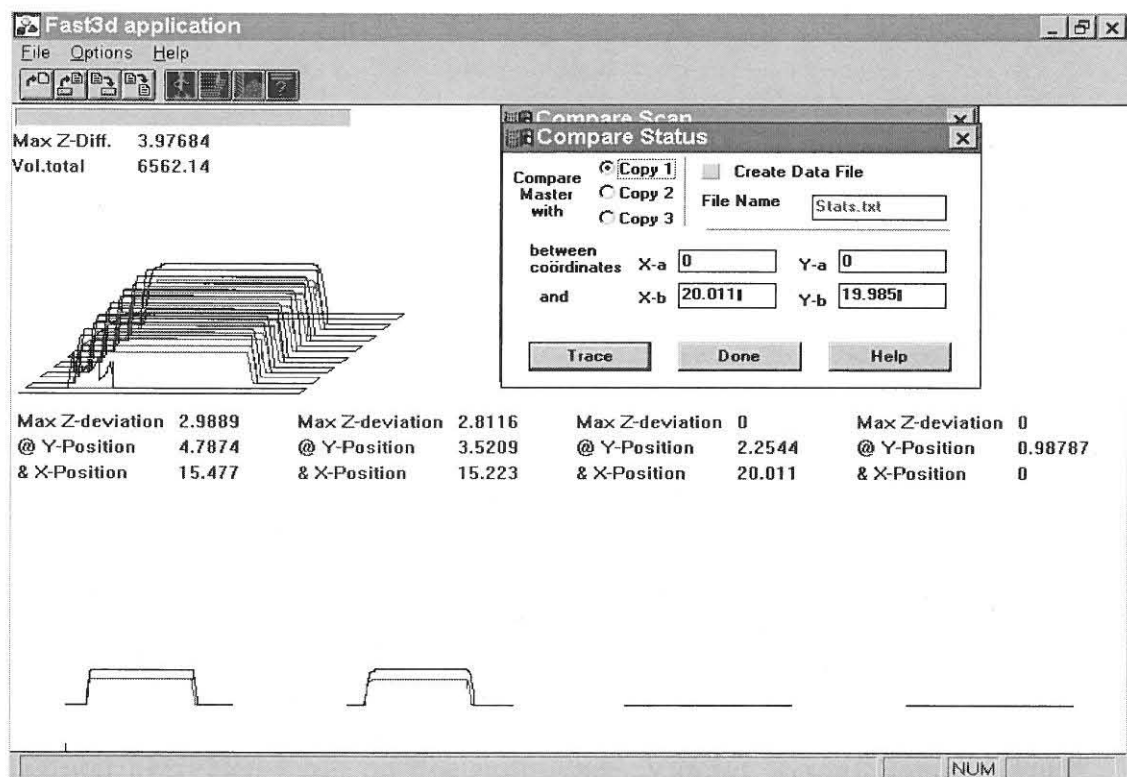
is by Borland C++5 en C++Builder, wat van die Borland-saamsteleenheid gebruik maak om masjienkode te genereer. Daar is dus besluit om die sagteware te skryf in Borland C++ 5 en Visual Basic 6, sodat 'n vergelyking getref kan word tussen die tale en om dan met die geskikste taal die sagteware daar te stel. Die eindproduk is in Visual Basic 6 geskryf vir redes soos in die volgende paragrawe verduidelik word. Vergelyking tussen die twee tale word ook in die onderafdelings getref.

7.1.1 Borland C++ 5

C++ wat op die C-programmeertaal gebou is, is 'n middelvlakprogrammeertaal en ook die oorspronklike taal waarin Windows geprogrammeer is [22, p. 4]. C++ is waarskynlik die kragtigste programmeertaal en word beskou as 'n middelvlakprogrammeertaal, omdat dit slegs 'n beperkte aantal sleutelwoorde bevat, wat dit toelaat om bykans enige taak te verrig, maar om nog steeds die effektiwiteit van laevlaksaaamsteltale te behou [22, p. 5]. Soms raak die gebrek aan laevlak programmeeringsvryheid in hoëvlaktale 'n struikelblok, wanneer sekere take verrig moet word. Nog 'n groot voordeel van C++ is sy uitvoerspoed, wat vergelykbaar is met die spoed van saamsteltale.

Omdat C++ objekgeoriënteerd is, kan modules ook maklik oorgedra word na ander platforms, sonder om baie veranderings aan die kode te doen.

Die C++ programmatuur word gekompileer, wat beteken dat die bron-kode eenmalig omgeskakel word na die objek- of masjien-kode - wat rekenaars direk kan interpreteer. Sommige tale soos die Visual Basic 4, en ouer weergawes, het interpreteerders nodig gehad wat die bron-kode lyn vir lyn na masjien-kode omskakel elke keer wanneer die sagteware in werking tree word. Die nadeel van geïnterpreteerde sagteware is die degradering in uitvoerspoed [27, p. 7]. Figuur 7.1 toon die C++-weergawe van die Reverse3D-sagteware, wat tydens die navorsing geskryf is.



Figuur 7.1 Die C++-weergawe van Reverse3D

Die prys wat betaal word vir programering in C++ is tyd, a.g.v sy kompleksiteit en moeilikheid om te ontfout. Omdat die program, ten spyte

van modulariteit, uit duisende lyne programkode kan bestaan, raak dit moeilik om te volg. Die programmeerder word opgehou met baie laevlak-programmering, wat soms in wese niks met die spesifieke sagteware te doene het nie. In C++Builder word die probleem egter tot 'n groot mate opgelos. Omdat die skryf van Windows-sagteware in C++ so baie tyd in beslag neem, is besluit om nie dié taal te gebruik tot en met die voltooiing van die navorsing nie.

7.1.2 Visual Basic 6.

Visual Basic is spesifiek ontwerp om die ontwikkeling van grafiese Windows-sagteware eenvoudig te maak. Omdat daar soveel oorhoofse programmering betrokke is by Windows-programmering, was dit voorheen slegs ervare programmeerders wat in staat was om toepassings vir dié bedieningsplatform te skryf. Wat Visual Basic so gewild maak, is sy eenvoud en verkorte programmeringstyd vir die ontwerper. Visual Basic genereer baie van die kode namens die programmeerder en hanteer baie lae-vlakwerking in Windows, sodat gekonsentreer kan word op spesifieke roetines wat betrekking het op die sagteware. Vandat Visual Basic (Visual Basic 5 en later) in masjienkode saamgestel kan word en nie meer die *VBRUN600.dll*-lêer benodig om die P-kode (soos dit bekend staan) te interpreteer nie, het dit 'n baie gewilde taal geword [14, p. 755]. Die vermoë om derdeparty-beheeritems te kan gebruik bespaar ook baie tyd en

maak die sagteware aantreklik. Die drie basiese ontwerpstappe in Visual Basic is as volg [22, p. 3]:

- Plaas die beheeritems op die beheerraampie.
- Voorsien die beheeritems van die nodige metodes en eienskappe.
- Skryf die Basic-programkode.

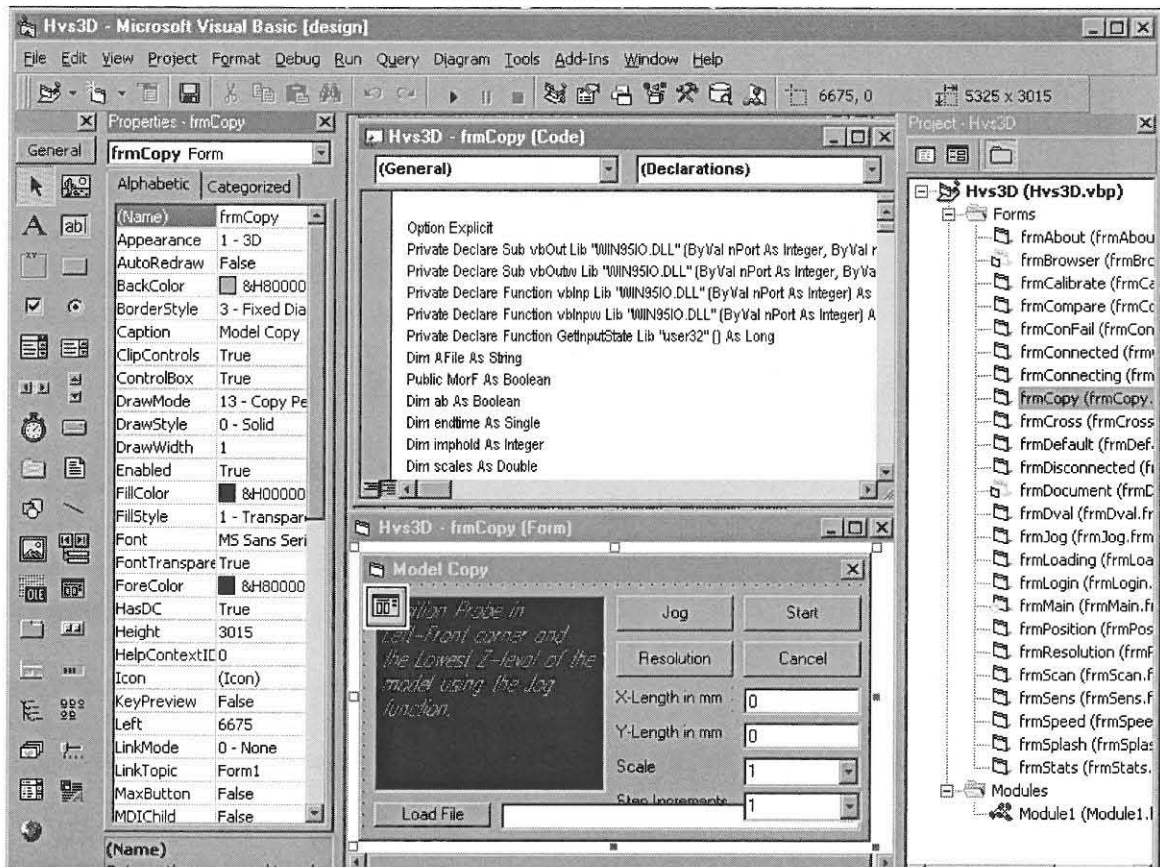
Voorbeelde van beheeritems is drukknoppe, klinkknoppe, opsieknoppe, teksraampies en talle ander waaruit gekies kan word. Die metodes en eienskappe waarvan melding gemaak is, spesifiseer voorkomsverwante karakteristieke van die onderskeie beheeritems [6, p. 22]. In Figuur 7.2 word die “Integrated Development Environment” (IDE) getoon met die beheeritems heel links langs die metode- en eienskapraampies geleë. In die middel van die skerm is die ontwerp van ‘n beheerraampie; onder, die koderaaampie.

In Windows word daar vir alle gebeure of voorvalle boodskappe gegenereer wat, indien nie dadelik aandag kan geniet nie, hiërargies gestoor word, totdat dit bedien kan word. Die voorvalle sluit sleutelbord knoppies gedruk, muisklik, dubbele muisklik, muisbeweging, muistrek en baie ander in. Alle aktiewe programme in die Windowsomgewing monitor dié boodskappe. Waar ‘n roetine gekoppel is met ‘n spesifieke boodskap, sal dit uitgevoer word wanneer die boodskap sy opwagting maak.

In Visual Basic word 'n lys van moontlike boodskappe gegee, waarop 'n spesifieke beheeritem kan reageer. Die Basic-kode word vir die betrokke voorval waarop gereageer moet word, geskryf. Vanuit 'n spesifieke voorvalroetine kan ander gespesialiseerde roetines geïnisieer word. In Borland C++ vind die monitor van Windows-boodskappe op 'n baie laer vlak plaas. Boodskappe moet herken word om boodskap-parameters te isoleer en te interpreteer, voordat dit bruikbaar sal wees. In Engels sal daar gesê word: "The message is cracked".

Behalwe die effense degradering van uitvoerspoed van 'n Visual Basic-program, is die programmeerder beperk daarmee om dieselfde te doen as dit wat moontlik is met die C++-programmeertaal. Om dié rede laat Visual Basic die programmeerder toe om die Win32 "Application Program Interface" (API) te benut en dieselfde roetines as C-programmeerders te gebruik. Sodoende word 'n groot deel van die probleem opgelos [22, p. 4].

Vir die sagteware wat in die navorsing geskryf is, het Visual Basic volkome voldoen aan die vereistes. Waar dit nodig was, is die nodige API-roetines gebruik, of die nodige *dll-lêer* is vanaf die internet verkry. Visual Basic is baie gebruikersvriendelik en 'n baie vinnige manier om Windows sagteware te skep.



Figuur 7.2 Die Visual Basic IDE

7.2 Module-uitleg van die programmatuur

Die Reverse3D bestaan uit 24 beheerraampies ("forms" genoem, tydens ontwerptyd) wat getoon word in die projek-uitlegraampie (regs geleë op die IDE in Figuur 7.2), en 'n aantal algemene-beheerraampies wat nie getoon word nie. Elkeen van die modules het 'n groot aantal roetines om hulle funksioneel te maak. Sommige van die roetines word in reaksie op gebeure (bv. knoppies wat gedruk word) opgeroep, terwyl sommige deur ander roetines geaktiveer word.

Die funksies van die kernroetines word vervolgens bespreek. Al die roetines met 'n `_Click` agtervoegsel word in reaksie op 'n muisklik op 'n beheeritem geaktiveer. Die naam van die betrokke beheeritem staan voor die "`_Click`".

In Bylaag A word die vloeiagramme van dele van die sagteware wat betrekking het op die oriëntasie, aftasting en vergelyking van die data getoon.

- **FrmMain**

- ♦ "FrmMain" hanteer al die menu-itemvoorvalle van die sagteware. Daar is 63 verskillende roetines wat, in reaksie op menu-itemvoorvalle, die beheerraam na die skerm roep of ander roetines inisieer. Individuele bespreking van die roetines is onnodig.

- ♦ **Homing**

"Homing" word in reaksie op die `mnuConnect`-menu-item se "`Click`"-voorval uitgevoer. Dit is dié roetine wat die aanvangspunte bepaal.

- ♦ **Toolbar1_ButtonClick**

Die "`Toolbar1_ButtonClick`"-roetine reageer op die nutsbalkknoppies, om dit met die nodige roetines te koppel. Een van die knoppies roteer die groepe items deur spesifieke groepe sigbaar te maak, wanneer die knoppie gedruk word.

- **FrmCalibrate**

Die “Calibrate”-beheerraampie is verantwoordelik vir die kalibrasie van die twee Max NC-masjiene.

- ♦ **position_Click**

Verkry die opsieknoppie-status, genereer die relevante boodskap aan die gebruiker en roep die “Jog”-beheerraampie na die skerm.

- ♦ **Calibrate_Click**

Dit is die roetine wat ‘n spesifieke as kalibreer.

- ♦ **Done_Click**

Vertoon die kalibrasie en maan die gebruiker om dit te verwerp of te aanvaar.

- **FrmJog**

Die "Jog"-beheerraampie laat die gebruiker toe om die voelpe op die masjiene te posisioneer en is ook verantwoordelik daarvoor om tred te hou van die Cartesiese-koördinaatposisie.

- ♦ **JogX_Click, JogY_Click, JogZ_Click, JogRot_Click**

Wanneer ‘n opsieknoppie aktief raak word die “JogPos”-en “JogNeg”-knoppie se opskrifwaarde verander om die rigting van beweging te reflekteer.

- ♦ **JogPos_Click, JogNeg_Click**

Die twee knoppies verkry die betrokke stande van die opsieknoppies en roep dan die “JogDirA_Click” of “JogDirB_Click”-roetines op.

- ♦ **JogDirA_Click, JogDirB_Click**

Die roetine bepaal die as vir beweging en roep die regte bewegingsroetine in werking. Dit monitor ook die voelpen-stand tydens as-bewegings.

- ♦ **ZA, ZB**

Dit merk die posisie vir die relatiewe nul op die Z-as.

- **FrmPosition**

- ♦ **Sscan_Click**

Hierdie is die beheerroetine wat stapsgewys die nodige roetines in werking stel om omtrekaftasting van ‘n model te doen. Dit is ook verantwoordelik om foutboodskappe te genereer, indien daar iets verkeerd loop in die proses. Kyk na Bylaag A-5 vir die vloeddiagram.

- ♦ **FindCircStart**

Die eerste stap van omtrekaftasting is om die model op te spoor en ‘n beginpunt te bepaal waar die aftasting in aanvang kan neem. Indien geen model opgespoor word voor die gespesifiseerde

minimum nie, sal die roetine 'n foutboodskap genereer en die proses halt.

♦ **Mcircmf**

Hierdie roetine is verantwoordelik vir motorbeheer by omtrekaftasting en besluit watter een van die "Last (A,B,C of D)"-roetines op 'n gegewe oomblik geaktiveer moet word. Die roetine eindig wanneer enige punt 'n herhaling is van die eerste koördinaat. Kyk na Bylaag A-3 vir die vloediagram.

♦ **LastA, LastB, LastC, LastD**

Omdat alle bewegings afhang van die vier voorafgegaande bewegings, word die vier roetines gebruik om, volgens wat voorafgaande gebeur het, die volgende beweging te bepaal. Op hul beurt roep hulle weer een van die agt bewegingsroetines op. Kyk na Bylaag A-4 vir die vloediagram.

♦ **Up, Down, Left, Right, Forward, Backward, RotMotAC, RotMotC**

Dit is hierdie roetines wat fisies die motorbewegingskodes genereer wat na die elektroniese koppervlak gestuur word sodat dit op hul beurt die trappiesmotor-kode kan genereer. Kyk na Bylaag A-8 vir die vloediagram van die "Backward"-roetine.

♦ **Level_Click**

Die roetine beheer die aftasroetines om modeloriëntasie tot mekaar te bepaal. Kyk na Bylaag A-9 vir die vloediagram.

♦ **LevScan**

Dié roetine bereken die aftasroetes vir gelykheidsaftasting volgens die lyntal (geselekteer deur die gebruiker) en modeldimensies (verkry vanaf die omtrekaftasting).

♦ **LineScan4Pos**

Hierdie roetine monitor die voelpe en roep die nodige bewegingsroetines op om die aftasting van 'n lyn te bewerkstellig. Die roetine word herhaaldelik met die nodige parameters geroep, totdat genoegsame lyne vir vergelyking verkry is.

♦ **ToStart**

Hierdie roetine keer die voelpe terug na die gestoorde aanvangsposisie van die aftasting.

♦ **Rotate, Crss, InCircDraw, RotDispl**

Hierdie roetines is verantwoordelik vir die omskakeling van die geometriese data na 'n visuele beeld op die rekenaarskerm. Dit bevat komplekse wiskundige formules, wat onder andere die skaal en rotasie-hoek-parameters vertolk om die beeld te genereer. Die lysing van die "Rotate"-roetine word in Bylaag B getoon.

♦ **InCircDraw**

Hierdie roetine, soos die vorige drie roetines wat bespreek is, is verantwoordelik vir die omskakeling van die geometriese data na 'n visuele beeld op die rekenaarskerm, maar in dié geval word dit tydens aftasting gedoen.

Tydens omtrekaftasting is geen modeldimensies bekend nie. Dit is om hierdie rede dat die "InCircDraw"-roetine geen skaalwaarde in sy parameterlys kan bevat nie. Die skaal moet voortdurend aangepas word om die beeld op die rekenaarskerm in te pas. Die lysing van die "InCircDraw"-roetine word in Bylaag C getoon.

- ♦ **Touched**

Hierdie roetine kommunikeer met die elektroniese koppelvlak, om die voelpen-stand te bepaal. Kyk na Bylaag A-7 vir die vloedidiagram.

- **FrmCross**

Hierdie beheerraampie se primêre doel is om aan die gebruiker 'n visuele beeld te gee van die gelykheidsaftasting, sodat verseker kan word dat die modelle dieselfde oriëntasie in die Cartesiese vlak het.

- ♦ **DrawThing**

Die "DrawThing"-roetine is soortgelyk aan die grafiese roetines wat reeds bespreek is, maar laat individuele lyn-inspeksie teen verskillende vergrotings toe.

- **FrmScan**

Hierdie beheerraampie is verantwoordelik vir die aftas van modelle.

- ♦ **ScanningXY, ScanningYX**

Hierdie roetines bepaal die aftas-roetes, afhangende van die gesellekteerde aftasmetode. Lynroetes parallel aan die X-as word deur ScanningXY bereken, terwyl lyne parallel aan die Y-as deur

ScanningYX bereken word. Kyk na Bylaag A-1 vir die vloeddiagram van ScanningXY.

♦ **LineScann**

Hierdie roetine word deur die ScanningXY- en ScanningYX-roetines geroep en is ook verantwoordelik vir die stoor van data. Verder is dit ook verantwoordelik vir die vertoning van die modelbeeld tydens die aftastingsproses. Kyk na Bylaag A-2 vir die vloeddiagram.

♦ **DrawThing2**

Hierdie roetine is verantwoordelik vir die teken van die model beeld op die rekenaarskerm by voltooiing van die aftasting.

♦ **Sve_Click**

Hierdie roetine inisieer die algemene “Save”-beheerraampie om die data met ‘n sinvolle naam te stoor.

♦ **LSet_Click**

Hierdie roetine verander alle verstellings vir ‘n spesifieke aftasting na die verstellings wat gegeld het tydens ‘n vroeëre aftasting van ‘n model, waarvan die data gestoor word.

♦ **Export**

Hierdie roetine omskep die *.hvs*-lêerformaat in ‘n *.dxf*-lêerformaat.

- **FrmCompare**

Die beheerraampie word as voorganger gebruik, wanneer die data van twee modelle vergelyk word. Die twee hoofroetines open twee lêers wat vergelyk moet word. Voordat die frmStats-beheerraampie geroep word, word daar gekyk of die data met mekaar vergelyk kan word. Indien nie die vergelyking onuitvoerbaar is, sal 'n foutboodskap vertoon word en die vergelyking gekanselleer word.

- **FrmStats**

- ♦ **Draw, DrawLine**

Hierdie twee roetines se funksies is om data grafies te vertoon.

- ♦ **Volume**

Hierdie roetine doen die vergelyking van die twee stelle data.

7.3 Roetines in DLL-leërs

Die *.dll*-leërs stoor algemene roetines wat op 'n gereelde basis deur verskeie rekenaarsagteware gebruik word. So word baie beheeritems in *.dll*-leërs gestoor, wat op sy beurt in die Windows/System-gids gestoor word. Die teenwoordigheid van sulke lêers in die Windows bedieningstelsel het tot gevolg dat baie geheuespasie gespaar kan word as gevolg van nie-herhalende programkodes in toepassingsagteware. Wanneer die lêers opgegradeer word, word die sagteware ook in der

waarheid opgegradeer. Die volgende lêers, wat almal met die beheeritems te doene het, is ingesluit in die sagteware:

- *VB6STKIT.dll*
- *STDOLE2.dll*
- *ASYCFILT.dll*
- *OLEPRO32.dll*
- *OLEAUT32.dll*
- *MSVBVM60.dll*

Al die lêers is in die System-gids gestoor. Tydens installasie word daar dan na die teenwoordigheid van die lêers gesoek. Wanneer hulle gevind word, word hulle, indien nodig, met die goedkeuring van die gebruiker op datum gebring

Verder word daar 'n *.dll* wat nie normaalweg in die System-gids gevind word nie, gebruik. Die *Win96IO.dll* lêer is verkry vanaf die internet [24, p. 1] en is 'n produk van SoftCircuits. Die doel van die *.dll* is om in- en uitset via die parallelpoort in Windows 95 en 98 te bewerkstellig. Visual Basic kan nie direk met die rekenaar se hardeware kommunikeer nie, weens van die beskermende voorsorgmaatreëls wat deur Microsoft daargestel is. Die *.dll*-lêer maak in- en uitset moontlik en word ook in die System-gids gestoor.

7.4 Opsomming

In dié hoofstuk is 'n opsomming verskaf van programmeertale wat oorweeg is vir die skryf van die navorsingsagteware. 'n Vergelyking is getref tussen twee programmeertale, naamlik Borland C++ 5 en Visual Basic 6. Die roetines betrokke by die funksies waarvan in Hoofstuk 6 melding gemaak is, is bespreek, asook die *.dll*-lêers wat gebruik is.

STELSELEVALUASIE

Na afloop van die projek was dit nodig om die stelsel op die proef te stel, ten einde die akkuraatheid daarvan te bepaal, asook om ontfouting te doen. Die eksperimente wat gedoen is, het behels dat modelle met bekende verskille afgetas word en dan om die teoretiese resultate met die praktiese te vergelyk. Faktore wat aanleiding gee tot die afname in akkuraatheid, word ook in hierdie hoofstuk bespreek.

8.1 Eksperiment A: Metingsakkuraatheid

Daar is op 'n vroeë stadium van die navorsing toetse gedoen om te bepaal of die afstande wat masjiene beweeg, ooreenstem met dié afstande wat deur die rekenaar verwag word, aangesien daar geen posisionele terugvoering is by die Max NC-aftasmasjiene nie. Die noodsaaklikheid van kalibrasie op die motore vanweë hulle bewegingstraagheid, het daaruit aan die lig gekom. Tabel 8.1 toon die werklike afstande wat beweeg is, teenoor die afstande wat die rekenaar tydens beweging op die Y-as verwag. Die tabel vergelyk rekenaarwaardes wat opgeneem word met werklike afstande wat beweeg is (verkry van 'n elektroniese meetpasser) met kalibrasie, asook sonder kalibrasie.

Tabel 8.1 Akkurater metings d.m.v. kalibrasie

	Sonder kalibrasie		Gekalibreerd		
	Rekenaar waarde	Meet- passer	Kalibrasie afstand	Rekenaar waarde	Meet- passer
Masjien A Y+	2.5584	2.42	0.204672	2.5584	2.56
Y-	2.5584	2.36	0.204672	2.5584	2.55
Y+	2.5584	2.4	0.204672	2.5584	2.55
Y-	2.5584	2.4	0.204672	2.5584	2.55
Masjien B Y+	2.5584	2.29	0.307008	2.5584	2.54
Y-	2.5584	2.3	0.307008	2.5584	2.54
Y+	2.5584	2.29	0.307008	2.5584	2.55
Y-	2.5584	2.29	0.307008	2.5584	2.54

Vanuit die data in Tabel 8.1 kan gesien word hoe die akkuraatheid verbeter het as gevolg van die kalibrasiestappe, wat tydens rigtingsveranderinge geneem word.

8.2 Eksperiment B: Rotasie-akkuraatheid

Die kleinste staphoek vir horisontale rotasie wat moontlik is met die ekstra gemonteerde trappiesmotor op die Max NC-CMM, is 0.025° /stap. Daar is eksperimenteel bevestig dat 'n volledige rotasie vir elke 14400 stappe voltooi word.

Om binne die akkuraatheidssperke te val moet die modelgrootte waarvan omtrekaftasting gedoen word, en die aftasresolusie in 'n spesifieke verhouding tot mekaar wees. Die afstand wat die modelrand beweeg vir 'n gegewe rotasiehoek, vergroot met die toename in afstand tussen die modelrand en die rotasiemiddelpunt. Die verhouding word as volg gedefinieer:

$$d = r/\tan (0.025)^\circ, \quad (8.1)$$

waar d = afstand tussen die modelrand en die rotasiemiddelpunt

r = aftasresolusie

Daarom sal die maksimum afstand d by die minimum r van $6.48 \mu\text{m}$ gelyk wees aan die volgende:

$$6.48\mu\text{m}/\tan (0.025)^\circ = 14851.056\mu\text{m} = 14.85 \text{ cm}.$$

Die rotasiestaphoek is dus klein genoeg ten einde modelrandafstande wat kleiner is as 'n aftasresolusie van $6.48\mu\text{m}$ te kan beweeg, indien die afstand na die rotasiemiddelpunt nie 14.85cm oorskry nie. Teen groter aftasresolusies is horisontale rotasie van groter modelle moontlik.

8.3 Eksperiment C: Volumeberekeninge

'n Aantal modelle waarvan die volume bekend is, is afgetas, sodat bepaal kon word hoe akkuraat die aftasting gedoen word. Daar is ook modelle waarvan die verskille bekend is, afgetas, sodat bepaal kon word hoe akkuraat die rekenaar die verskille identifiseer. Figuur 3.1 toon die

skermaftreksel van 'n deel van 'n model wat ondersoek word vir vergelyking. Die diepte van die gat is 5.31 mm en sy deursnee is 5.98 mm.

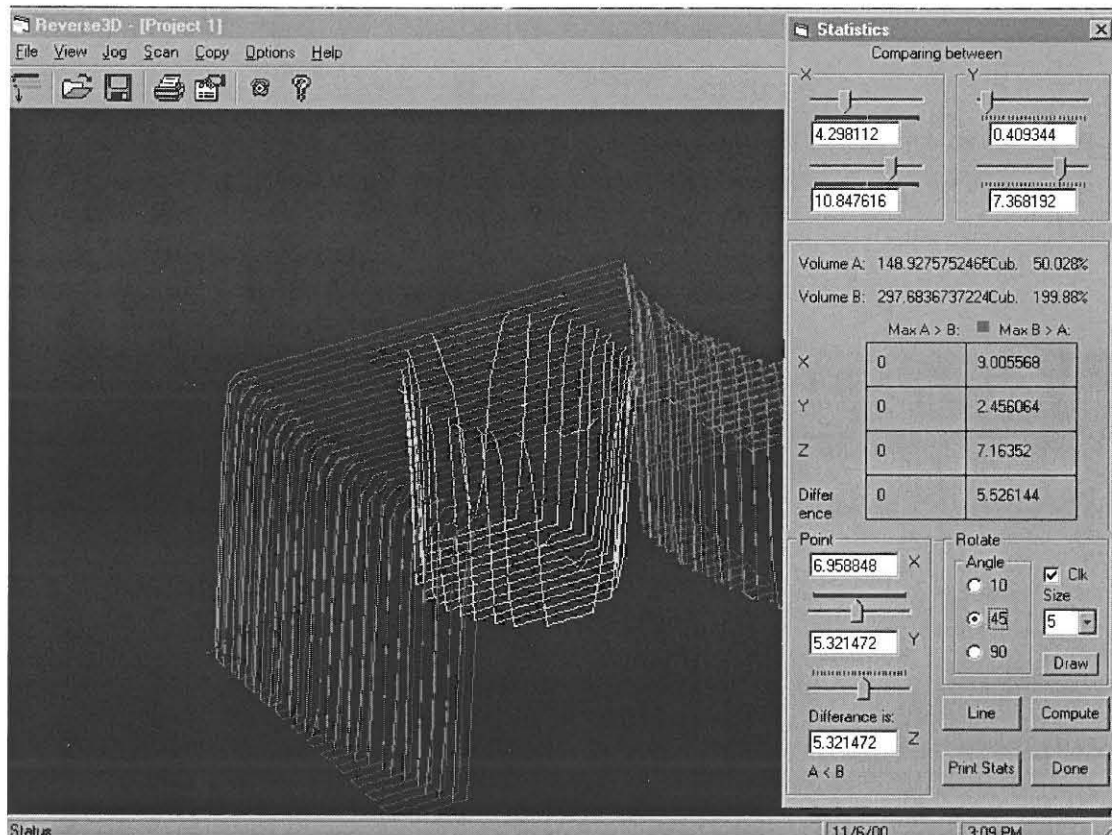
Die wiskundige volume (V) van die gat is:

$$V = \pi \times (\text{radius})^2 \times \text{diepte van gat}$$

$$V = \pi \times (5.98 \div 2)^2 \times 5.31$$

$$V = 149.137 \text{ mm}^3.$$

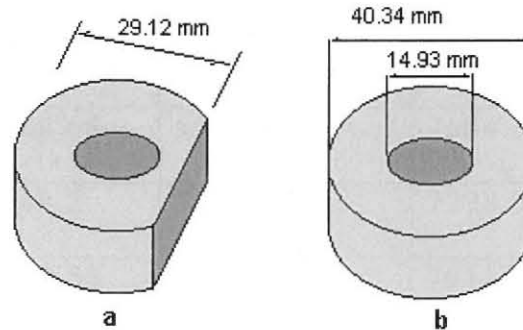
Die volumeverskil volgens die rekenaar is 148.77 mm^3 en vergelyk goed met die teoretiese waarde.



Figuur 8.1 Berekening van volume

8.4 Eksperiment D: Vergelyking van modelle

Twee identiese modelle, waarvan een verander is soos getoon in Figuur 8.2., is gebruik om die vergelyking van modelle d.m.v. die Reverse 3D-sagteware te evalueer.



Figuur 8.2 Dimensies van die modelle wat vergelyk is

Die teoretiese volume, V_b , van model b word soos volg bereken:

$$V_b = \pi \times \text{Hoogte} \times (20.17^2 - 7.435^2) \quad (8.2)$$

Die teoretiese waarde van die segmentvolume (V_{Seg}) wat verwyder is, word as volg bereken:

$$V_{\text{seg}} = \frac{1}{2} \times \text{radius}^2 \times [\alpha - \text{Sin} \alpha]$$

$$\text{Waar } \alpha = 2 \text{Cos}^{-1}(\text{Aangrensende sy} / \text{Skuins sy})$$

$$\therefore V_{\text{seg}} = \frac{1}{2} \times (20.17)^2 \times \text{Hoogte} \times [2 \text{Cos}^{-1}(8.95 \div 20.17) - \text{Sin}(2 \text{Cos}^{-1}(8.95 \div 20.17))] \quad (8.3)$$

Volume V_a word as volg bereken.

$$V_a = \pi \times \text{Hoogte} \times (20.17^2 - 7.435^2) - V_{\text{seg}} \quad (8.4)$$

Tabel 8.2 toon die vergelyking van teoretiese met die eksperimentele waardes.

Tabel 8.2 Resultate van Eksperiment 3

Z-Diepte	Model A		Model B			
	Bereken	Rekenaar	Bereken	Rekenaar	%T	%P
6.958848	5664.68	5547.462	7684.33	7620.616	73.7	72.7
7.061184	5749.2	5582.0929	7798.55	7655.776	73.7	72.9
7.16352	5832.53	5817.795	7911.58	7888.428	73.7	73.7

Die teoretiese, sowel die praktiese resultate van al drie aftastings, vergelyk redelik goed met mekaar. Die praktiese resultate was in twee gevalle'n fraksie minder as die berekende waardes.

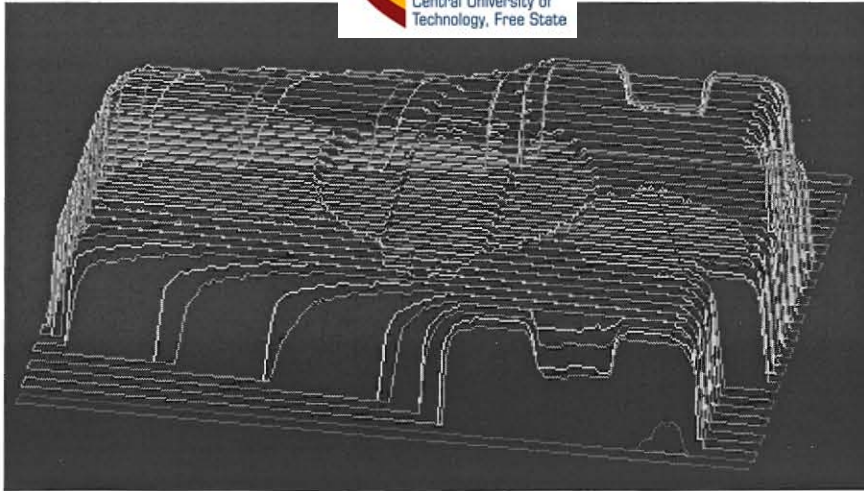
8.5 Eksperiment E: Ewekansige steekproefvergelyking

Om aan te toon hoe akkuraat aftasting van twee identiese modelle gedoen word, is tien datapunte ewekansig getrek en die data vergelyk. Tabel 8.3 toon die tien datapunte wat vergelyk is.

Tabel 8.3 Datapunte van ewekansige steekproef

Nr	X-koördinaat	Y-koördinaat	Z-koördinaat verskil
1	32.9521	21.2858	0
2	1.2280	18.0111	0.2046
3	9.0055	27.8353	0
4	18.2158	23.7419	0
5	33.5662	21.6952	0
6	35.5895	19.6485	0
7	27.4260	15.5550	0
8	20.0578	11.4616	0
9	13.1054	2.4550	0.2046
10	15.5530	4.4027	0

Die data van agt datapunte was identies en die van twee datapunte het op die Z-as verskil. Figuur 8.2 is 'n skermafbeelding van modelle wat vergelyk is. Daar kan duidelik gesien word hoe groot die verskille is tussen die Z-koördinate, wat op die modelrand geleë is. Hierdie verskille dra by tot die onakkuraatheid van die vergelykings wat getref word. Dit is 'n blote toeval dat daar met die steekproef geen datakoördinate getrek is wat op die modelrand voorkom nie. Hierdie verskynsel skep egter steeds 'n probleem tydens datavergelyking.



Figuur 8.3 Groot koördinaatverskille op vertikale modelrande

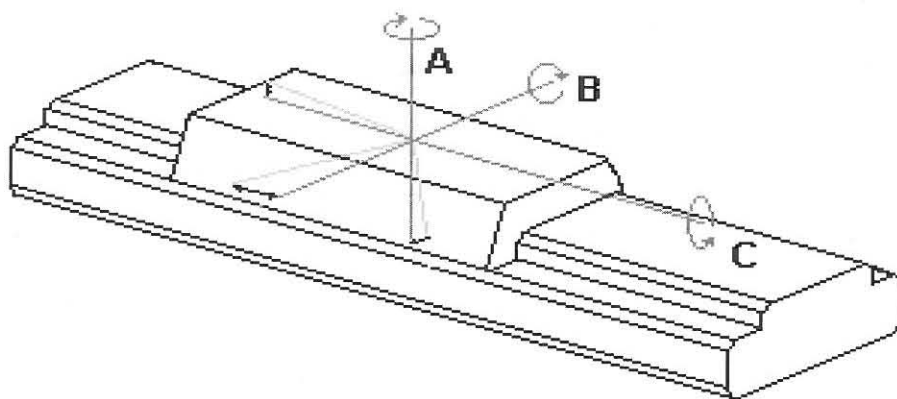
8.6 Akkuraatheidsdegradeering

Die meganiese konstruksie van 'n CMM speel 'n baie groot rol in die akkuraatheid van die aftasting. Sekere van die beperkings kan in die sagteware gekorrigeer word soos dit bespreek is in Hoofstuk 7. Die invloed van die motorresolusie en voelpensensitiwiteit op die akkuraatheid is ook voor die hand liggend.

Die voelpe wat gebruik is vir data-opname was die tassakelaar voelpentipe en is toegerus met 'n baie kort penpunt van ongeveer 2cm lank. Die rede hiervoor is om die voelpensensitiwiteit te optimaliseer. Die meganiese aspek van die voelpe wat tydens die navorsing ontwerp is kon egter, weens 'n gebrek aan befondsing, nie verwesenlik word nie.

Daarom kon daar geen vergelyking tussen die verskillende voelpenne getref word nie.

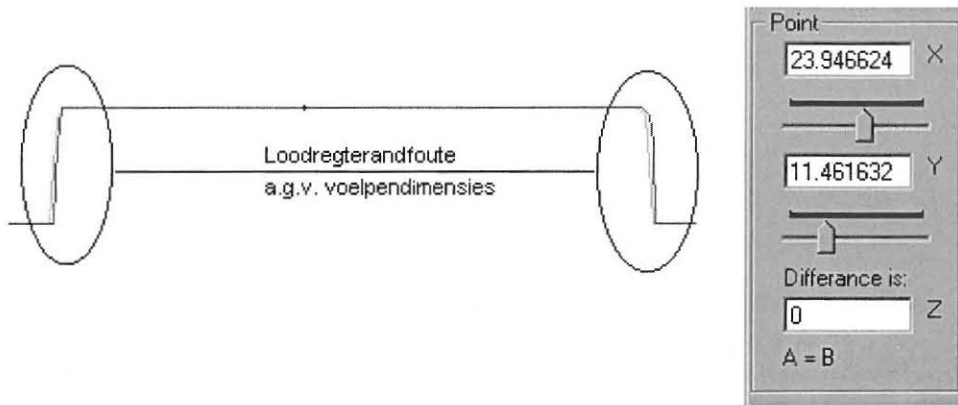
Daar is egter ook ander faktore by CMM's wat aanleiding gee tot foutiewe metings. CMM's wat van die skroef-en-glybaanmetode gebruik maak vir driedimensionele beweging, ondervind klein rotasies in die al drie asse wanneer die bewegingsrigting verander. Dié rotasies veroorsaak dat die Z-koördinaatlesings wat geneem word, nie dié is van die betrokke XY-koördinaat ter sprake nie [15, p. 72]. Die rotasie beïnvloed veral groter modelle, waar die koördinaatverskuiwing toeneem volgens die afstand vanaf die rotasiepunt. Om die grootte van die rotasies van die drie asse te bepaal vereis baie fyn instrumentasie. Voorts sal die sagteware aangepas moet word om die foute in berekening te bring. Figuur 8.2 toon die rotasieprobleme wat ondervind word met die skroef-en-glybaanmetode. Rotasie-A skep die grootste probleem, omdat dit die X- en Y-assisies verskuif.



Figuur 8.4 Foute by glybaan-bewegingsmetode

Daar word in die praktyk baie navorsing gedoen om die meganiese probleme wat ondervind word met CMM's en soortgelyke masjiene uit te skakel. Liniêre laers met balskroefaandrywingsmeganismes op dubbel bane waarop die blad kan gly, is reeds van so 'n aard dat die speling op die asse weglaatbaar klein word [16, p. 50]. Die speling, resolusie, spoed en torsie van trappiesmotore wat tans beskikbaar is, is ook baie beter as dié waarmee die Max NC CMM toegerus is. Wat eksterne sensors vir die meting van koördinate betref, kan verplasings al van tot so klein as $0.0086\mu\text{m}$ deur hologramtegnologie bepaal word [16, p. 32].

Die groot koördinaat verskille op vertikale modelrande waarvan in afdeling 8.5 melding gemaak is, lewer ook 'n groot bydrae tot onakkurate vergelykings. Nog 'n faktor wat aanleiding kan gee tot foutiewe lesings op vertikale modelrande is onsilindriese voelpenne. Voelpeursneekompensasie kan slegs suksesvol toegepas word, indien die voelpen volkome silindries en die punt rond is. Voelpeursneekompensasie (in Hoofstuk 3 bespreek) voorkom die volumetoename wat gepaard gaan met foutiewe lesings as gevolg van voelpenpuntradius. Indien die voelpenpunt taps afgewerk is, sal die volume-aanpassing foutief wees. Voelpenne wat nie volkome silindries is nie, dra ook by tot die afname in akkuraatheid. Figuur 8.3 toon foutiewe lesings op loodregte modelrande.



Figuur 8.5 Foutiewe lesings by vertikale modelrande

Uit die eksperimente is dit duidelik hoe die teoretiese volumes die rekenaarvolumes oorskry. 'n Moontlike oorsaak hiervoor is dat die resulterende volumevermindering groter is as wat benodig word.

SAMEVATTING

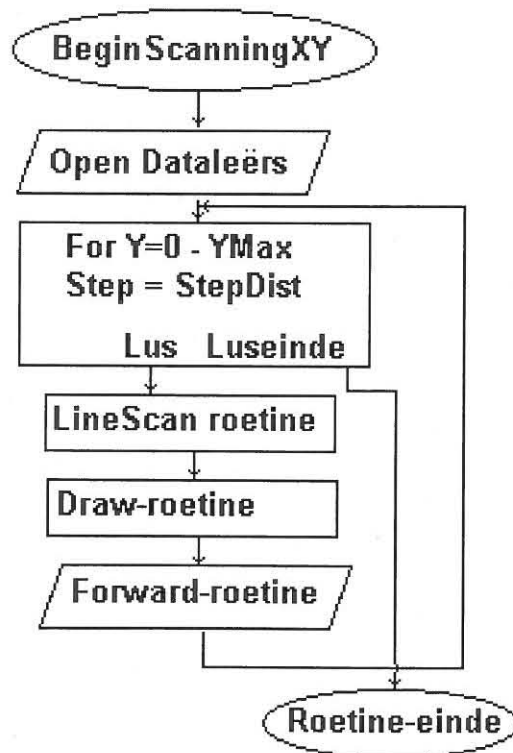
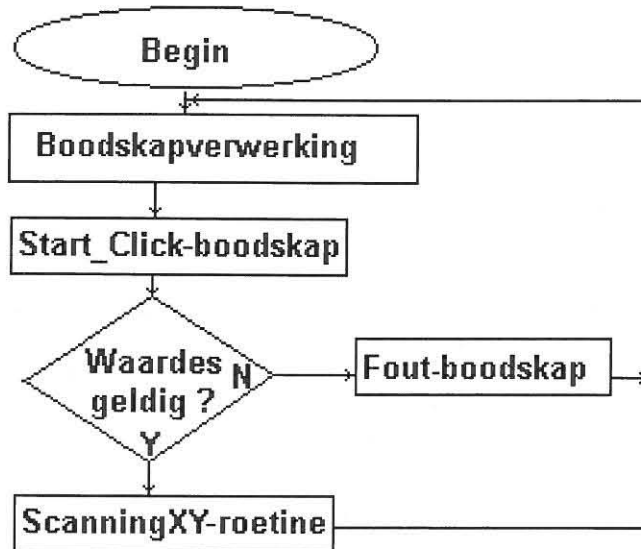
Die studie het ten doel gehad die daarstelling van 'n stelsel wat dit moontlik sal maak om driedimensionele voorwerpe met mekaar te vergelyk. Daar moes ook gekonsentreer word op metodes wat gevolg kan word om die tekortkominge van die gebruikte CMM's te oorkom, asook om akkurate voelpenne te ontwerp. Die vergelyking van driedimensionele voorwerpe deur die vergelykingsmetode wat daargestel is, word suksesvol uitgevoer, maar baie akkurater vergelykings kan verwag word, indien hierdie beginsels toegepas kan word op 'n meer akkurate aftasmasjien, met akkurater voelpenne.

In Hoofstuk 2 is inleidende inligting verskaf oor NC-, CNC- en CMM-masjiene om as grondslag te dien waarop die daaropvolgende hoofstukke bou. Hoofstuk 3 het die voelpenne en die vergelykingsmetodes wat ontwerp is, bespreek. Hoofstuk 4 het die hardeware wat ontwerp is vir die projek toegelig, terwyl die betrokke lêerformate in Hoofstuk 5 aan die orde gekom het. Hoofstukke 6 en 7 het gekyk na die sagteware en programmatuur daaragter. Die evaluering van die stelsel en gevolgtrekkings daarop is in Hoofstuk 8 aangebied.

Verdere navorsingsvoorstel

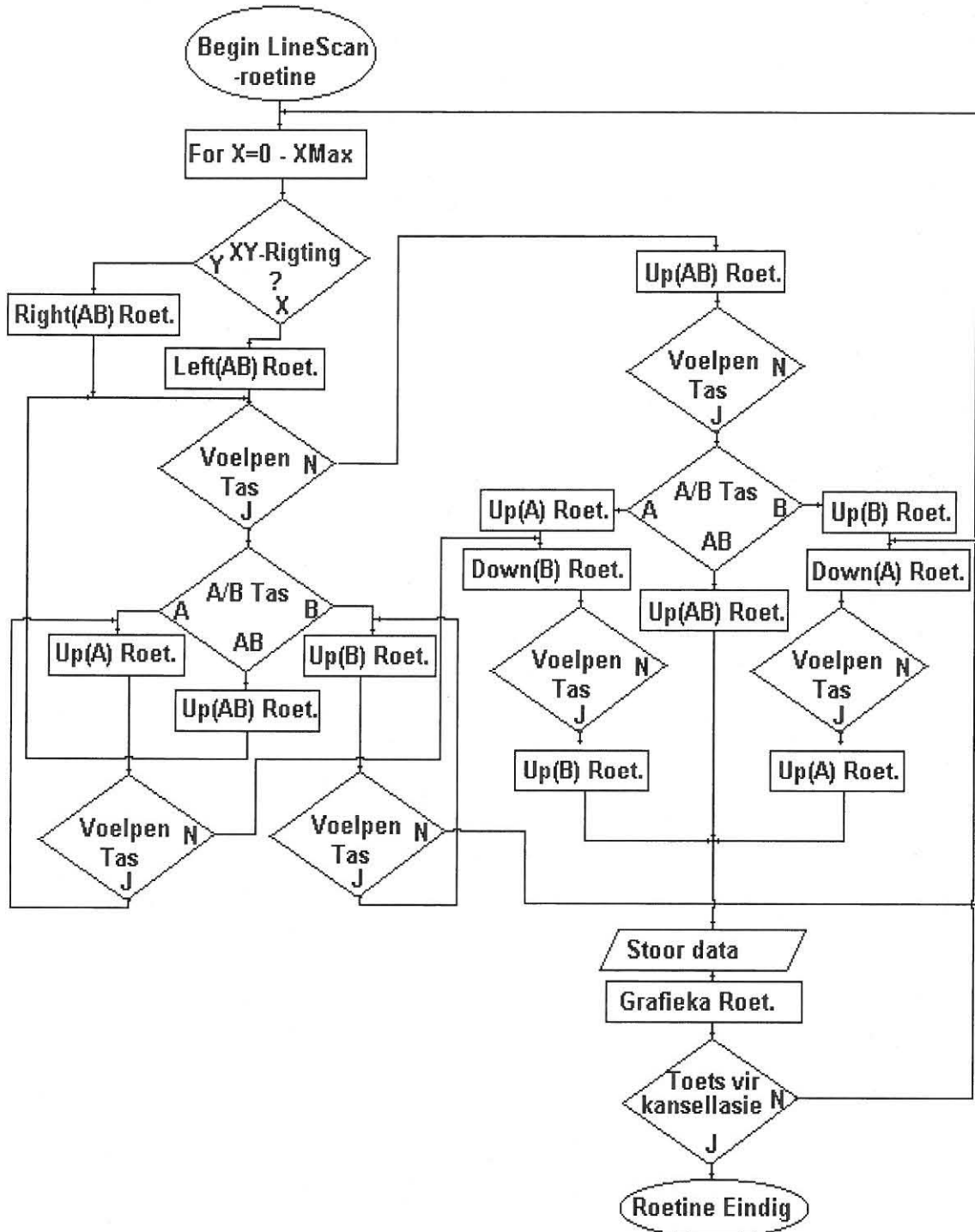
Daar word reeds intensief navorsing gedoen op CMM-masjiene vir dataopname. Daar kan dus eerder navorsing gedoen word om die vergelykingsmetodes wat uit hierdie navorsing na vore gekom het op ander CMM-masjiene te implementeer, sodat 'n baie akkurate vergelykingstelsel daargestel kan word. Daar kan ook verder navorsing gedoen word oor die vergelykingsmetodes wat in Hoofstuk 3 bespreek is. Navorsing oor die vergelyking van neutrale lêerformate, bv..*stl*, met mekaar, kan ook van groot nut wees in hierdie veld van ingenieurswese.

Vloediagramme

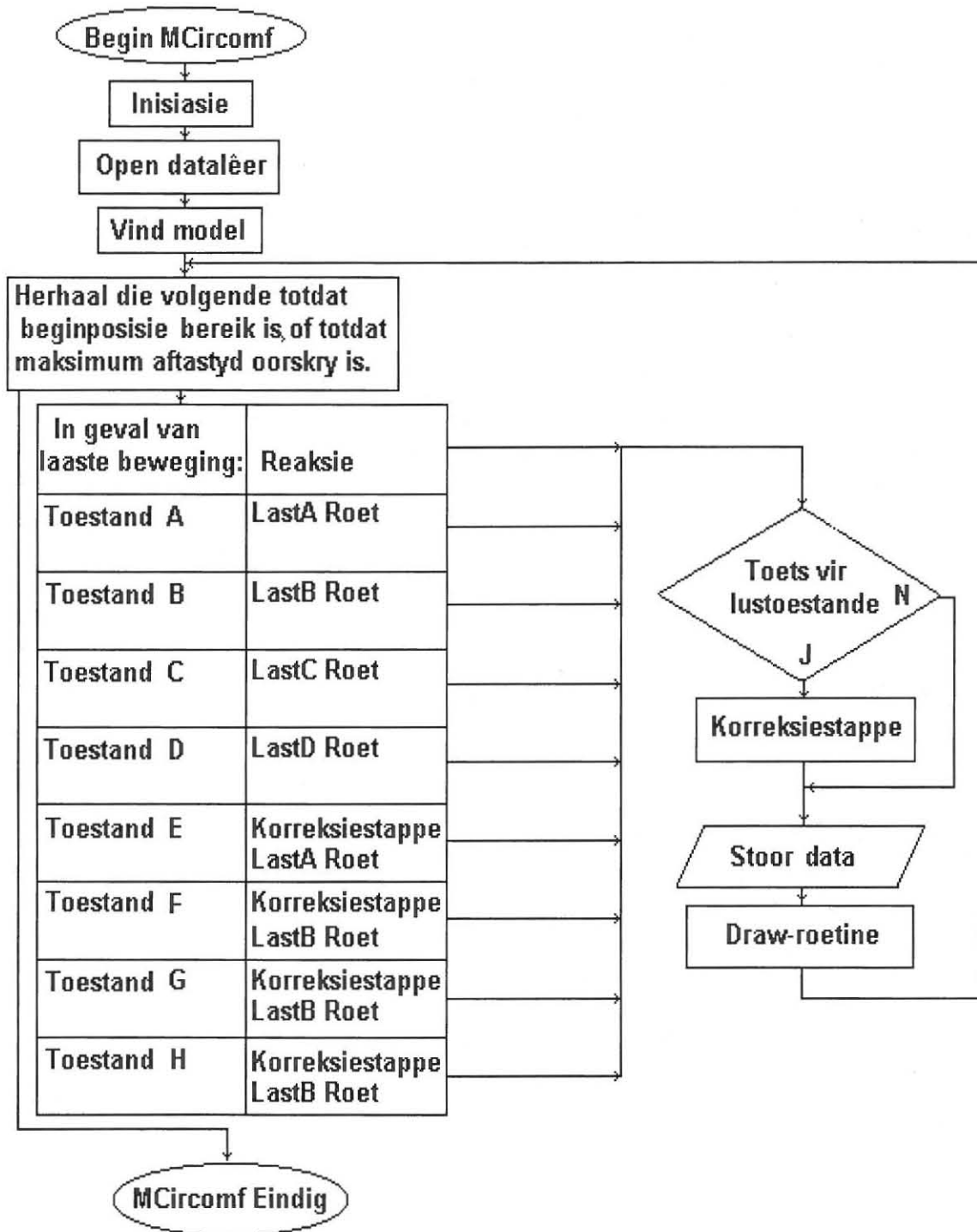


A1.

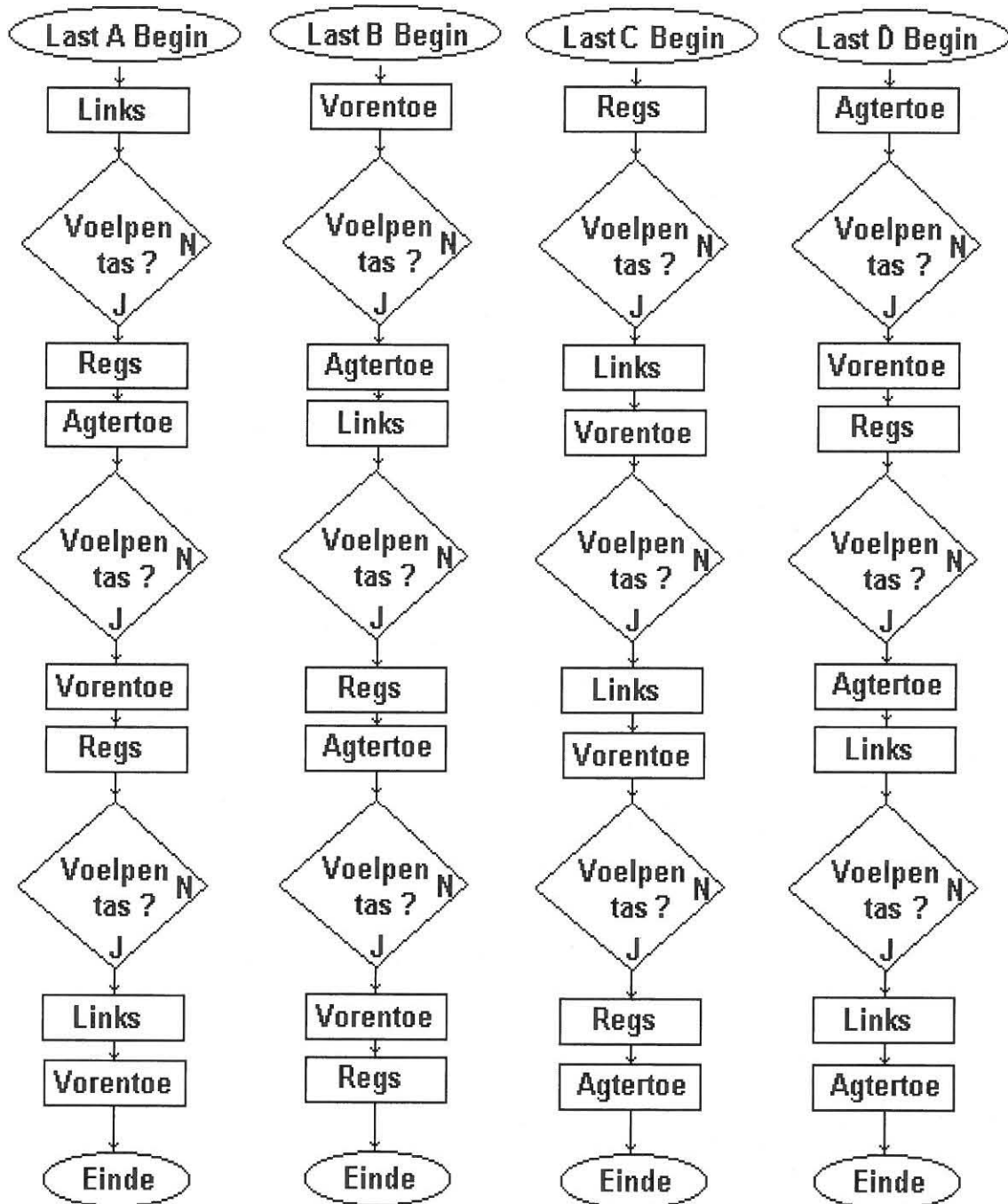
A2.



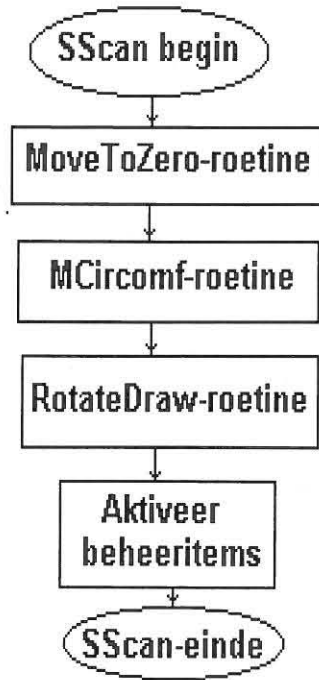
A3.



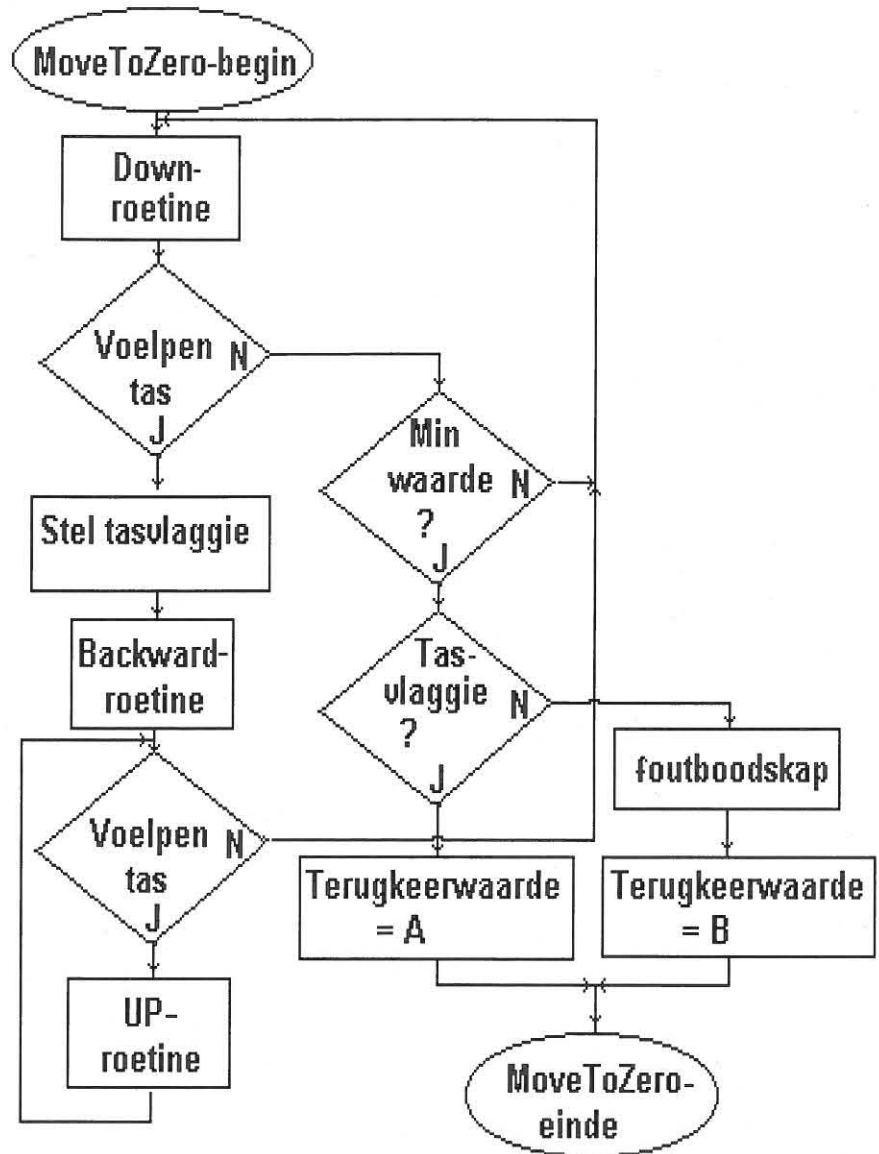
A4.



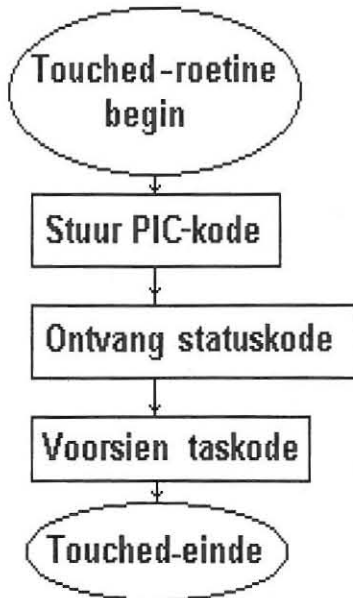
A5.



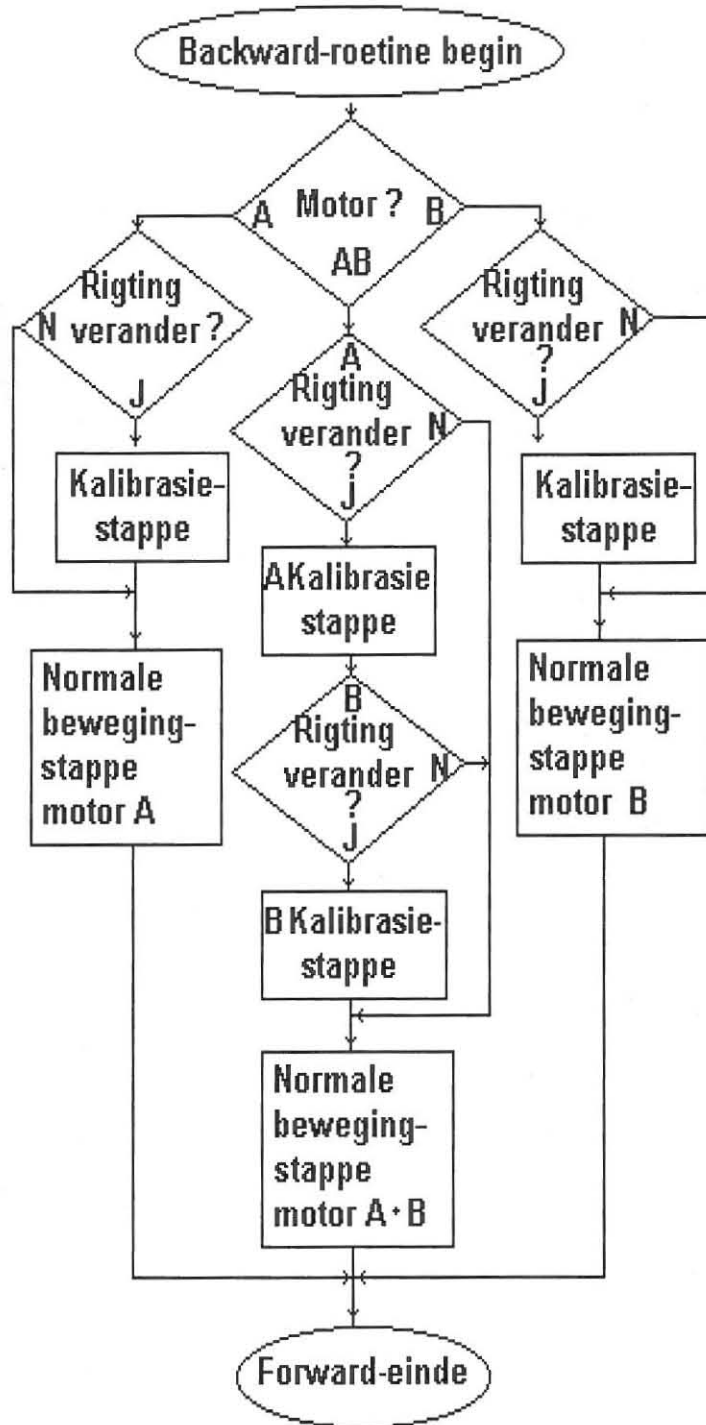
A6.

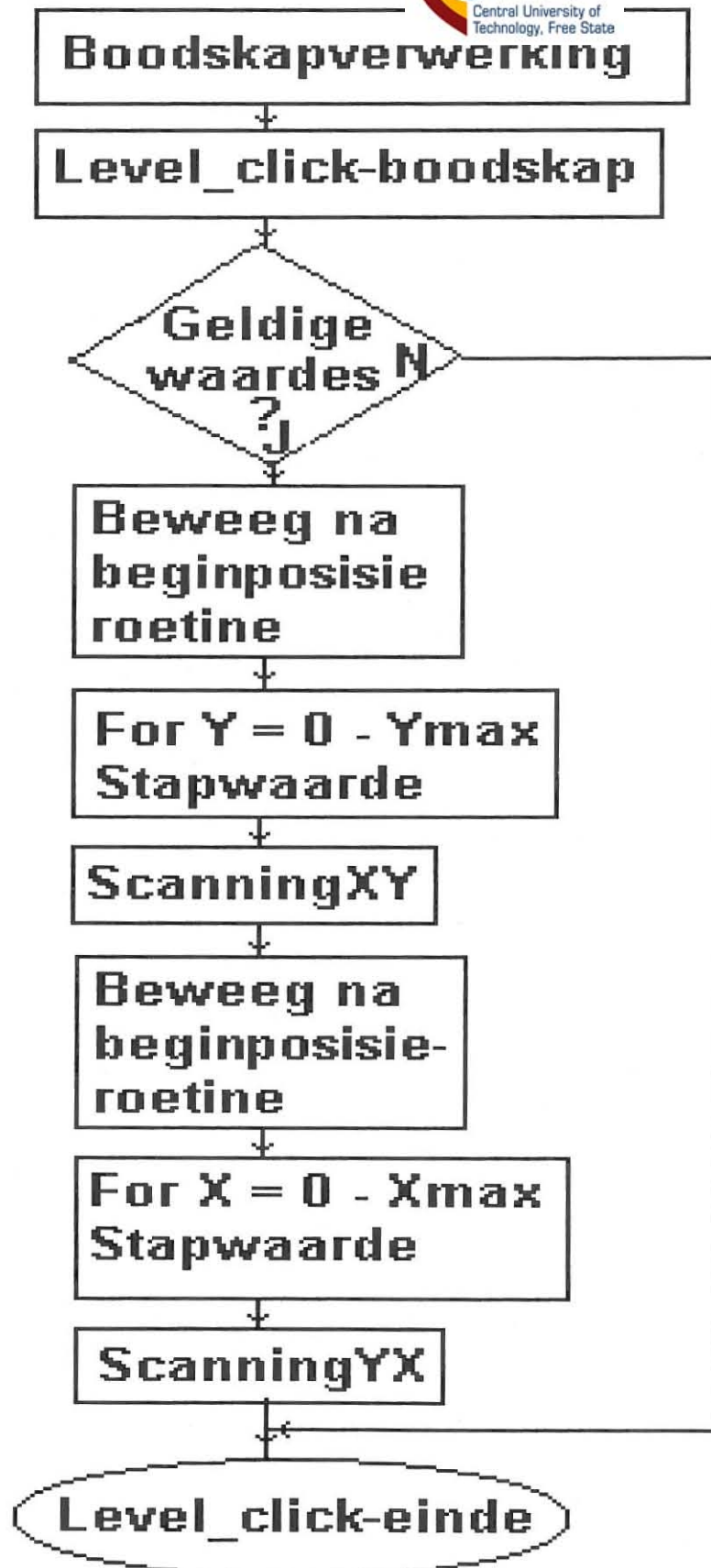


A7.



A8.





Die Rotate-roetine se programlys

```
Public Sub Rotate(Rotv As Integer, AFile As String, PColor As  
Integer, size As Double)  
Dim XBuff As String  
Dim YBuff As String  
Dim XBuffv As Double  
Dim YBuffv As Double  
Dim XLMax As Integer  
Dim YLMax As Integer  
Dim XLMin As Integer  
Dim YLMin As Integer  
Dim XW As Integer  
Dim YW As Integer  
Dim XMS As Integer  
Dim YMS As Integer  
Dim XDims As Integer  
Dim YDims As Integer  
Dim XYDims As Integer  
Dim XYW As Integer  
Dim XPVal As Double  
Dim YPVal As Double  
Dim ind As Integer 'remove  
Dim Rotvv As Double  
Dim Theta As Double  
Dim SVal As Double  
Dim endtime As Double  
Rotvv = Rotv * Pi / 180  
ind = 0  
On Error GoTo Error_Handler  
Open AFile For Input As #1  
Do While Not EOF(1)  
Line Input #1, XBuff  
If XBuff = "Round End" Then  
Exit Do  
End If  
Line Input #1, YBuff  
XBuffv = val(XBuff)  
YBuffv = val(YBuff)  
If XBuffv > XLMax Then  
XLMax = XBuffv  
End If
```

```

If XBuffv < XLMin Then
XLMin = XBuffv
End If
If YBuffv > YLMax Then
YLMax = YBuffv
End If
If YBuffv < YLMin Then
YLMin = YBuffv
End If
Loop
Close #1
XW = XLMax - XLMin
YW = YLMax - YLMin
If XW > YW Then
XYW = XW
Else: XYW = YW
End If
XMS = XW / 2
YMS = YW / 2
XDims = 490 * size
YDims = 310 * size
If XDims > YDims Then
XYDims = XDims
Else
XYDims = YDims
End If
XSent = 490 / 2
YSent = 310 / 2
'YNN = Sin(Rotvv)
'XNN = Cos(Rotvv)
If SS = True Then
scales = XYDims / XYW
End If
frmDocument.Line -(XSent, YSent), QBColor(0)
Open AFile For Input As #1
Open "s" For Output As #2
Do While Not EOF(1)
Line Input #1, XBuff
If XBuff = "Round End" Then
Exit Do
End If
Line Input #1, YBuff
XBuffv = val(XBuff)
YBuffv = val(YBuff)
SVal = Sqr(XBuffv * XBuffv + YBuffv * YBuffv)

```



```

If SVal > Abs(YBuffv) Then

  If Sgn(XBuffv) = 1 Then
    Theta = Atn((YBuffv / SVal) / Sqr(-(YBuffv / SVal) * (YBuffv /
SVal) + 1))
  Else
    If Sgn(XBuffv) = -1 Then
      Theta = Pi + Atn((YBuffv / SVal) / Sqr(-(YBuffv / SVal) *
(YBuffv / SVal) + 1))
    End If
  End If

  Else
    If Sgn(YBuffv) = 1 Then
      Theta = Pi / 2      '90 * Pi / 180
    Else
      If Sgn(YBuffv) = -1 Then
        Theta = -Pi / 2   '-90 * Pi / 180
      Else
        If Sgn(YBuffv) = 0 Then

          If Sgn(XBuffv) = 1 Then
            Theta = 0
          Else
            If Sgn(XBuffv) = -1 Then
              Theta = Pi
            End If
          End If

        End If
      End If
    End If

  End If

  If (Theta > (Pi / 2)) Or (Theta < (-Pi / 2)) Then
    YPVal = 310 - (-SVal * Sin(Theta - Rotvv) * scales + YSent) '+
YMS * scales
    XPVal = (SVal * Cos(Theta - Rotvv) * scales + XSent) '- XMS *
scales
  Else
    YPVal = 310 - (SVal * Sin(Theta + Rotvv) * scales + YSent) '+
YMS * scales
    XPVal = (SVal * Cos(Theta + Rotvv) * scales + XSent) '- XMS *
scales
  End If

```

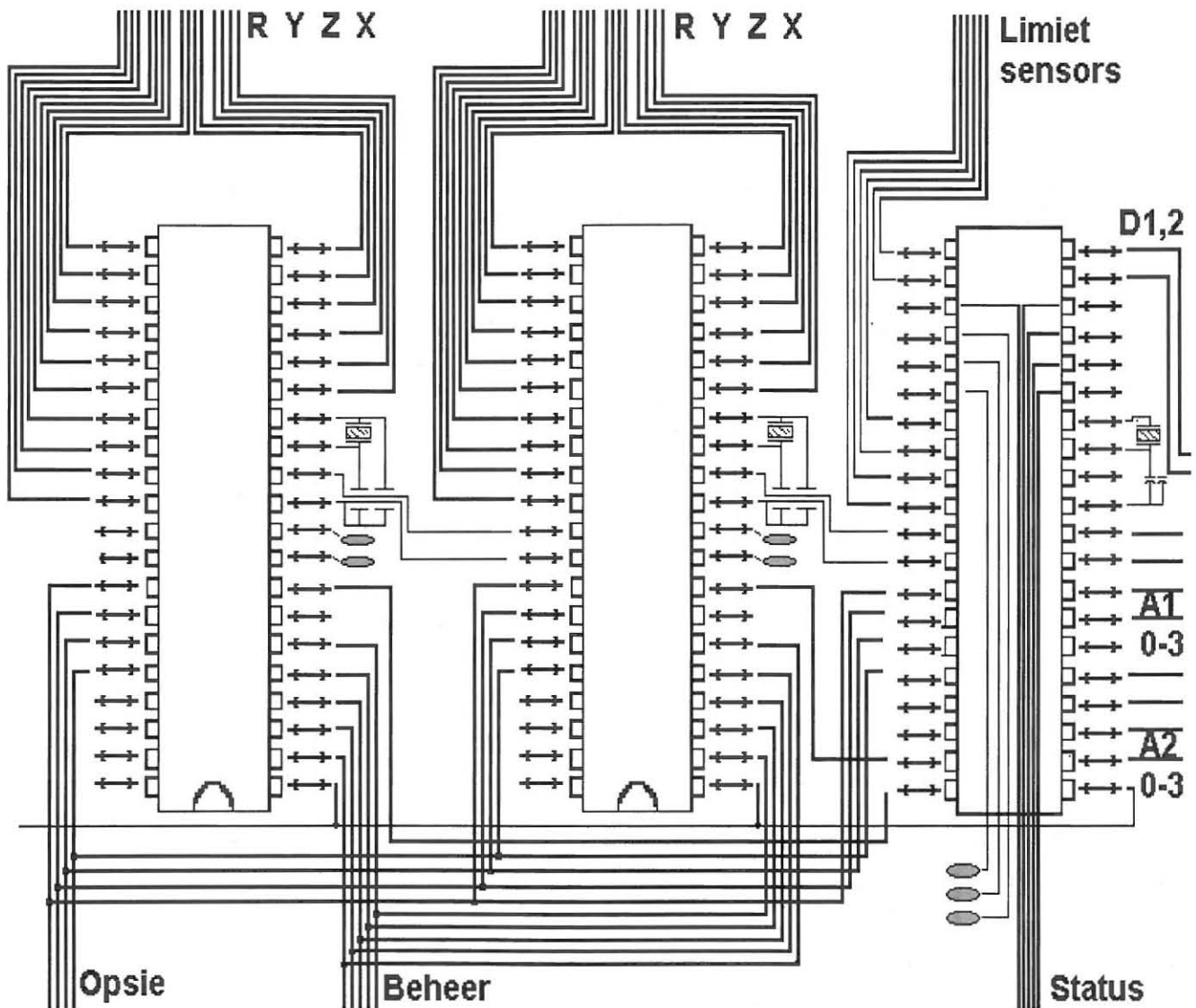
```
ind = ind + 1
Print #2, XPVal
Print #2, YPVal
If ind = 1 Then
frmDocument.Line -(XPVal, YPVal), QBColor(0)
Else
frmDocument.Line -(XPVal, YPVal), QBColor(PColor)
End If
Loop
Close #1
Close #2
Exit Sub
Error_Handler:
MsgBox "Error number " & Err & vbCrLf & Err.Description, 48,
"Whoops"
Close #1
Close #2
Exit Sub
End Sub
```

Die InCircDraw-roetine se programlys

```
Public Sub INCircDraw(XSTT As Long, YSTT As Long, XLMax As
Long, YLMax As Long)
Dim XYMAX As Long
Dim XBuff As String
Dim YBuff As String
Dim XBuffv As Integer
Dim YBuffv As Integer
Dim Col As Integer
Close #1
Redraw = 0
Col = 0
On Error GoTo Error_Handler
Open "Templ" For Input As #1
XYMAX = Abs(XLMax)
If Abs(YLMax) > XYMAX Then
XYMAX = Abs(YLMax)
End If
Line Input #1, XBuff
Line Input #1, YBuff
XBuffv = val(XBuff)
YBuffv = val(YBuff)
Xft = XBuffv
Yft = YBuffv
frmDocument.Cls
scales = 180 / XYMAX
  Do While Not EOF(1)
    Line Input #1, XBuff
    Line Input #1, YBuff
    XBuffv = val(XBuff)
    YBuffv = val(YBuff)
    frmDocument.Line -((200 - (XBuffv - Xft) * scales), 200 -
(YBuffv - Yft) * scales), QBColor(Col)
    Col = 2
  Loop
Close #1
Open "Templ" For Append As #1
Error_Handler:
Exit Sub
End Sub
```

BYLAAG D

Skematiese diagram van die koppel-elektronika



LITERATUURLYS

- [1] **Aerospace engineering and engineering mathamatics-stl files;**
2000, <http://www.ae.utexas.edu/~reyes/titanium/stl/stlmarch17.htm>

- [2] **Attwell, P. Preview: World trends: MTR**, April 1995: p2, 25

- [3] **Beyond logic**, 2000, <http://www.senet.com.au/~cpeacock>

- [4] **Born, G. The File Formats Handbook**, 1ste uitg. VK, International
Thompson Publishing Inc. 1995

- [5] **CCS. PCB, PCM, and PCW PIC C Compiler: Reference
Manual**,1999

- [6] **Cornell, G. Visual Basic from ground up**, 1ste uitg. VSA,
Osborne/McGraw-Hill.
1997

- [7] **Crummett, W.P. Western, A.B. University Physics-Models and
applications**, 1ste uitg. VSA, Wm.C.Brown Publishers, 1992

- [8] **Delft Spline Systems. DeskProto Reference manual**, Delft, 1995

- [9] **Dvorak, P. Gest ready for software objects: Machine Design**,
January 1996: p77-81
- [10] **EMCO Technics. Software Description Unimat PC**, Mei 1991
- [11] **Fraden, J. Handbook of modern Sensors: Physics, designs and applications**, 2de uitg. VSA, AIP Press, 1997
- [12] **Gibbs, D. Crandell, T.M. An Introduction to CNC Machining and Programming**,
1ste uitg. New York, Industrial Press Inc., 1991
- [13] **Hans Wedemeyer's CNC mill**, 2000, <http://hans-w.com/cnc/htm>
- [14] **Harrington, J. Spenik, M. Brumbaugh, H. Diamond, C. Visual Basic Interactive Course**, 1ste uitg. VSA, Waite Group, 1997
- [15] **Kohl, R. Basics of Design Engineering: Machine Design**, January
1996: p 69-72
- [16] **Khol, R. Hologram technology goes to work: Machine Design**,
January 1995: p32-35

- [17] **Max NC.** Operating manual, 1996
- [18] **Max NC-Home page**, 2000, <http://www.maxnc.com>
- [19] **Microchip.** PIC16F87X Data Sheet, VSA, 1999
- [20] **Murray, D. vanRyper, W.** Encyclopedia of Graphics File Formats,
2de uitg. VSA, O'Reilly & Associates Inc., 1996
- [21] **Neutral file formats**_, 1996,
http://cadd.cern.ch.cad_geant_int/thesis/node31.html
- [22] **Pappas, C.H. Murray, W.H.** Visual Basic Programming with the
Windows API, 1ste uitg. VSA, Prentice-Hall, 1998
- [23] **Polywka, J. Gabrel, S.** Programming of Computer numerically
Controlled Machines, 1ste uitg. New York, Industrial Press
Inc., 1992
- [24] **Programming the parallel port in Visual Basic**, 1999,
<http://www.aaroncake.net/electronics/vblpt.htm>

- [25] **Renishaw.** Cyclone Reference Manual, 1996
- [26] **RS.** 1999-2000 Catalogue, March 1999
- [27] **Schildt, H.** Borland C++: The Complete Reference, 1ste uitg. VSA,
McGraw-Hill, 1997
- [28] **SGS.** Motion Control Application Manual, 1ste uitg. Italie, SGS
Groep, 1987
- [29] **Tele-manufacturing facility project-stl format description**, 2000,
<http://www.sdsc.edu/tmf/Stl-specs/stl.html>
- [30] **The CALM project-stl file format**, 1998,
<http://www.jtap.ac.uk/projects/jtap-314/report/stl.htm>
- [31] **Thyer, G.E.** Computer Numerical Control of Machine Tools, 2de
uitg. Groot Britanje, Newnes, 1991
- [32] **Window, A.L.** Strain Gauge Technology, 2de uitg. London, Elsevier
Science Publishers LTD, 1992