

## TADSOL-integratie in Unigraphics II

**Citation for published version (APA):**

Abeelen, van, H. J. M. (1994). *TADSOL-integratie in Unigraphics II*. (TU Eindhoven. Fac. Werktuigbouwkunde, Vakgroep WPA : rapporten). Technische Universiteit Eindhoven.

**Document status and date:**

Gepubliceerd: 01/01/1994

**Document Version:**

Uitgevers PDF, ook bekend als Version of Record

**Please check the document version of this publication:**

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

**General rights**

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

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

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

**Take down policy**

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

Technische Universiteit Eindhoven

Faculteit Werktuigbouwkunde

Vakgroep WPA

## **TADSOL-integratie in Unigraphics II**

Afstudeerverslag H.J.M. van Abeelen

Rapportnummer: WPA-220023

november 1994

Afstudeerhoogleraar : Prof.ir. J.M. van Bragt

Begeleider : Ir. A.T.J.M. Smals

Adviseur : Dr.ir. A.J. Klein Breteler

## **Aantekening bij het afstudeerverslag**

### **Rapportnummer WPA-220023**

In overleg met Dr.ir. Klein Breteler is besloten om in plaats van tadsol\_1.exe en tadsol\_2.exe, de originele TADSOL-programma's te blijven gebruiken. Hierdoor blijft het voor hem mogelijk om bij eventuele problemen hulp te bieden en kunnen nieuwe versies van TADSOL probleemloos ingepast worden.

Het werken met UGTADSOL blijft precies hetzelfde. Voor het werken met TADSOL zie [2].

#### **Wijziging bijlage 2.**

De compile-commando's voor TADSOL worden nu:

```
f77 tadsol_c.f g0X.o GMS.o FILES.o STRINGS.o HPGL.o HEADL.o MS.o MR.o PIONXSDS.o -lxf11 -lx11
```

```
f77 tadsol.f g0X.o GMS.o FILES.o STRINGS.o HPGL.o MS.o MR.o sms.o smr.o PIONXSDS.o -lxf11 -lx11
```

```
f77 tadsol_s.f g0X.o GMS.o FILES.o STRINGS.o HPGL.o HEADS.o MS.o MR.o PIONXSDS.o -lxf11 -lx11
```

#### **Wijziging bijlage 5.**

In de hoofddirectory moeten tadsol\_c.exe, tadsol.exe en tadsol\_s.exe aanwezig zijn in plaats van tadsol\_1.exe en tadsol\_2.exe.

#### **Opmerking.**

Het originele tadsol\_c en tadsol\_s werken niet met unitnummer ipu=7. Dit is veranderd in het willekeurig gekozen unitnummer ipu=77.

## **SAMENVATTING**

Voor het realiseren van een bepaalde beweging in machines kunnen stangenmechanismen gebruikt worden. Vanwege het moeilijke ontwerpproces van stangenmechanismen kiest de ontwerper echter vaak voor een andere oplossing, bijvoorbeeld een nokmechanisme. Om het gebruik van stangenmechanismen te bevorderen wordt in dit onderzoek geprobeerd het ontwerpproces van stangenmechanismen te vereenvoudigen door middel van computerondersteuning. Eerder onderzoek heeft aangetoond dat het integreren van TADSOL in Unigraphics II daartoe mogelijkheden biedt. TADSOL is een op de TU Delft ontwikkeld pakket dat geschikt is voor de synthese van stangenmechanismen. Unigraphics II is het CAD/CAM/CAE-pakket dat binnen de vakgroep WPA van de TU Eindhoven gebruikt wordt. Vervolgonderzoek heeft als resultaat een aantal programma's (UGTADSOL genaamd) opgeleverd dat de communicatie tussen TADSOL en Unigraphics II verzorgt.

In dit onderzoek zijn deze bestaande programma's aangepast aan de laatste versie (10) van Unigraphics II. Daarna is de voor een zinvol gebruik van TADSOL noodzakelijke grafische uitvoer van overdrachtsfuncties verwezenlijkt. Ook is een methode aangegeven waarmee door TADSOL berekende mechanismen rechtstreeks in een machinetekening getekend kunnen worden en voor vier basistypen is deze uitgevoerd. Daarnaast is de gebruikersvriendelijkheid van UGTADSOL in zijn geheel een stuk verbeterd. Tenslotte is aangegeven wat de functie van UGTADSOL in het totale ontwerpproces van stangenmechanismen zou kunnen zijn.

Het belangrijkste resultaat van dit onderzoek is dat het ontwerpproces nu dusdanig ondersteund wordt dat het mogelijk is stangenmechanismen te ontwerpen zonder specialistische kinematische kennis.

## **SUMMARY**

Linkage mechanisms can be used to generate a given motion in a machine. Because of the complexity of the design process the engineer rarely applies a linkage in his design. To encourage the use of linkages the goal of this research is to simplify the design process of linkages by means of computer support. Former research has proven that the integration of TADSOL in Unigraphics II opens up possibilities in this direction. TADSOL is a computer program developed on the University of Technology in Delft that supports the synthesis of linkages. Unigraphics II is the CAD/CAM/CAE-system used by the section Specific Means of Production, faculty Mechanical Engineering of the University of Technology in Eindhoven. Follow-up research resulted in a number of computer programs (called UGTADSOL) which together make possible the communication between TADSOL and Unigraphics II.

In this research the existing computer programs have been adapted to the latest release (10) of Unigraphics II. After that the graphical output of TADSOL has been realised. For a proper use of TADSOL it is necessary that transfer functions can be shown on the screen. Then a method has been created by which a sketch of a solution calculated by TADSOL can be put directly in the drawing of a machine. This has been realised for four basic types of mechanisms. Next to that the overall user-friendliness of UGTADSOL has been improved. And finally the future function of UGTADSOL in the complete design process has been described.

The main result of this research is a computer supported design process that makes it possible to design linkage mechanisms without specific knowledge of kinetics.

## **VOORWOORD**

Dit rapport is het verslag van mijn afstudeerwerk op de TU Eindhoven in de periode van december 1993 t/m november 1994.

CAD-systemen en de mogelijkheden daarvan hebben altijd mijn belangstelling gehad. Ik vond het dan ook een mooie opdracht om hier gedurende mijn afstudeerperiode mee bezig te kunnen zijn. En ondanks dat het tijdens het onderzoek niet altijd heeft meegezeten, is het toch een interessante en leerzame tijd geweest.

Graag wil ik hier iedereen bedanken die mij heeft geholpen tijdens het afstuderen. Speciale dank aan mijn begeleiders Prof.ir. J.M. van Bragt, Ir. A.T.J.M. Smals en Dr.ir. A.J. Klein Breteler. Tevens wil ik de heren Soers en van Rooy bedanken voor hun hulp in de CAD-ruimte.

**Greer's third law:**

**A computer does what you tell it to do, not what you want it to do.**

## INHOUDSOPGAVE

SAMENVATTING	2
SUMMARY	3
VOORWOORD	4
1. INLEIDING	7
1.1. Algemene inleiding	7
1.2. De opdracht	7
2. ORIËNTATIE	9
2.1. Unigraphics II	9
2.2. TADSOL	9
2.2.1. Werking TADSOL	10
2.2.2. Opbouw TADSOL	11
2.3. UGTADSOL	12
3. PROBLEEMANALYSE	13
3.1. Deelprojecten	13
3.1.1. Herschrijven bestaande programma's	13
3.1.2. Grafische uitvoer op het scherm	13
3.1.3. Montagestandschets	13
3.1.4. Totale ontwerpproces	14
3.1.5. Gebruikersvriendelijkheid	14
4. HERSCHRIJVEN BESTAANDE PROGRAMMA'S	15
4.1. GRIP	15
4.2. UFUNC	15
4.3. Afweging	15
5. GRAFISCHE UITVOER OP HET SCHERM	17
5.1. Vervangen TADSOL	17
5.2. Aanroepen TADSOL	17
5.3. TADSOL los van Unigraphics	17
5.4. Grafische drivers	18
6. MONTAGESTANDSCHETS	19
6.1. Oriëntatie	19
6.2. Oplossingsgegevens	20
6.3. Van oplossingsgegevens tot montagestandschets	22
6.4. Automatische montagestandschets	24
6.4.1. Transformatieproces	24
6.4.2. Programma's	26
6.4.3. Alternatief	28

---

7. TOTALE ONTWERPPROCES	29
7.1. UGTADSOL	30
8. GEBRUIKERSVRIENDELIJKHEID	34
9. VOORBEELD GEBRUIK UGTADSOL	35
9.1. Bewegingsprobleem	35
9.2. TADSOL	35
9.3. UGTADSOL	37
9.4. Verschillende montagestanden	40
10. CONCLUSIES EN AANBEVELINGEN	42
10.1. Conclusies	42
10.2. Aanbevelingen	43
LITERATUURLIJST	44
BIJLAGEN	45
Bijlage 1. Opdrachtomschrijving	46
Bijlage 2. Grafische drivers	47
Bijlage 3. Tabel mechanismetypen	49
Bijlage 4. Programmeren montagestand	50
Bijlage 5. Installatie TADSOL	54
Bijlage 6. Voorbeeld TADSOL_IN.DAT	55
Bijlage 7. Voorbeeld TADSOL_OUT.LIS	56
Bijlage 8. Voorbeeld MECHAN_IN.DAT	60
Bijlage 9. Voorbeeld overdrachtsfuncties	61
Bijlage 10. Listing UGTADSOL.GRS	62
Bijlage 11. Listing V-ISO-3D.GRS	63
Bijlage 12. Listing INTERFACE.GRS	64
Bijlage 13. Listing MONTAGESTAND.GRS	70
Bijlage 14. Listing S006.GRS	72
Bijlage 15. Listing T008.GRS	74
Bijlage 16. Listing R003.GRS	75
Bijlage 17. Listing T006.GRS	77
Bijlage 18. Listing STANG.GRS	79
Bijlage 19. Listing TANDWIEL.GRS	80
Bijlage 20. Listing SCHUIF.GRS	81
Bijlage 21. Listing HEUGEL.GRS	82
Bijlage 22. Listing VASTTW.GRS	83
Bijlage 23. Listing VASTSCHAR.GRS	84



# 1. INLEIDING

## 1.1. Algemene inleiding

In machines moet vaak een bepaalde beweging gerealiseerd worden, afgeleid van een eenparig roterende as. Hiervoor worden mechanismen gebruikt. Er zijn mechanismen die aangedreven worden door hydraulische of pneumatische cilinders, maar de belangrijkste zijn de stangen- en nokmechanismen. In de praktijk blijkt dat ontwerpers vaak kiezen voor nokmechanismen in hun machine. De reden hiervoor is dat deze veel eenvoudiger te ontwerpen zijn dan stangenmechanismen. Maar nokmechanismen hebben ten opzichte van stangenmechanismen ook nadelen:

- de smering van het nok-volgercontact is moeilijk,
- vaak een grotere inbouwruimte nodig voor grote slaglengtes,
- de over te brengen kracht is kleiner door hoge contactspanning tussen nok en volger,
- bij een krachtgesloten verbinding tussen nok en volger is de maximale snelheid lager.

Daartegenover staat dat je met een nokmechanisme een rustperiode in de uitgaande beweging kunt realiseren.

In veel gevallen zou een stangenmechanisme misschien beter voldoen dan een nokmechanisme maar krijgt het vanwege het moeilijke ontwerpproces geen eerlijke kans. Het is dus zeker de moeite waard het ontwerpen van stangenmechanismen te vereenvoudigen. Het is van belang te onderzoeken of de computer daarbij behulpzaam kan zijn.

Voor de synthese van stangenmechanismen is op de Technische Universiteit Delft het rekenprogramma TADSOL ontwikkeld. Omdat bij het ontwerpen van machines de constructeur steeds vaker gebruik maakt van een CAD-systeem is het interessant om dit rekenprogramma te integreren in het CAD-systeem. Binnen onze vakgroep wordt het CAD/CAM/CAE-pakket Unigraphics II gebruikt, vandaar de titel van deze opdracht: TADSOL-integratie in Unigraphics II.

## 1.2. De opdracht

Het eerste werk in het kader van deze opdracht is geleverd door J.M.N. Leenders, waarna O.F.M. Niens de huidige koppeling, UGTADSOL genaamd, heeft gerealiseerd.

Mijn opdracht luidt nu:

Verbeter UGTADSOL zodat :

- het gebruikersvriendelijk is voor de ontwerper,
- alle functies van TADSOL beschikbaar zijn,
- alle catalogusmechanismen van TADSOL aanwezig zijn,
- de mechanismen parametrisch ontworpen kunnen worden.

De volledige opdrachtoomschrijving is te vinden in bijlage 1.

## 2. ORIËNTATIE

Om UGTADSOL te kunnen verbeteren is een gedegen kennis van zowel Unigraphics II als TADSOL nodig. Daarnaast moet er goed gekeken worden naar de mogelijkheden en beperkingen van het huidige UGTADSOL.

### 2.1. Unigraphics II

De vakgroep WPA heeft in haar CAD/CAM-ruimte de beschikking over een aantal Hewlett Packard werkstations. Daarop draait onder UNIX het pakket UniPRODUCTS geleverd door EDS. Eén van de onderdelen van UniPRODUCTS is het CAD/CAM/CAE-pakket Unigraphics II (UGII).

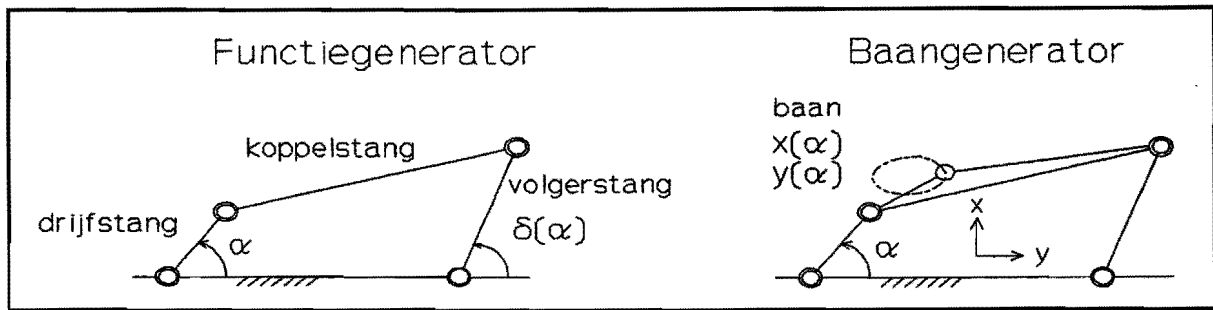
Binnen het CAD (Computer Aided Design) gedeelte is het mogelijk op eenvoudige wijze produktgeometrie te genereren in de vorm van solids (volumedelen).

Binnen het CAE (Computer Aided Engineering) gedeelte is het mogelijk allerlei berekeningen en analyses uit te voeren aan bestaande of nog te creëren geometrieën. Voor deze opdracht is de module MECHANISMS interessant. Binnen MECHANISMS is het mogelijk om bestaande geometrieën tot schakels te definiëren van een stangenmechanisme. Vervolgens kunnen de schakels verbonden worden met verschillende elementparen die de verbinding tussen de schakels vormen. Voorbeelden hiervan zijn draaipunten, rechtgeleidingen en tandwielparen. Het bewegingsverloop van het aldus gecreëerde mechanisme kan daarna geanalyseerd en gesimuleerd worden.

Binnen UGII is het mogelijk om met behulp van GRIP (GRaphical Interactive Programming) of UFUNC (User FUNCtions) programma's te maken die het CAD gedeelte van UGII bedienen. Bijna alle commando's die de ontwerper normaal gebruikt, kunnen geprogrammeerd worden. Ook kunnen mededelingen voor de gebruiker op het scherm getoond worden en zelf samengestelde keuzemenu's en invoermogelijkheden aan hem voorgelegd worden zodat hij interactief met de programma's kan werken. Daarnaast is het mogelijk om externe programma's in Fortran of C aan te roepen alsook externe files in te lezen.

### 2.2. TADSOL

Aan de Technische Universiteit Delft is door de CADOM-groep onder leiding van prof. Rankers een hele serie rekenpakketten ontwikkeld die het ontwerpproces van stangen- en nokmechanismen ondersteunen. Het pakket TADSOL (Type And Dimension Synthesis Of Linkages) is geschikt voor de synthese van een bepaalde groep stangenmechanismen, namelijk functiegeneratoren.



Figuur 1. Functie- en baangenerator.

Een functiegenerator realiseert een bepaalde overdrachtsfunctie tussen de in- en uitgaande schakel. Een ander type mechanismen zijn de baangeneratoren waarbij de baan van een punt op de koppelstang gerealiseerd moet worden (zie figuur 1). Voor beide wordt uitgegaan van een continu roterende aandrijfjas. Binnen de functiegeneratoren is nog een verdere indeling te maken op grond van het bewegingskenmerk. De uitgaande beweging

Bewegingskenmerk	Voorbeeld	Doelfunctie
Roterend (R)		
Slingerend (S)		
Translerend (T)		

Figuur 2. Bewegingskenmerk.

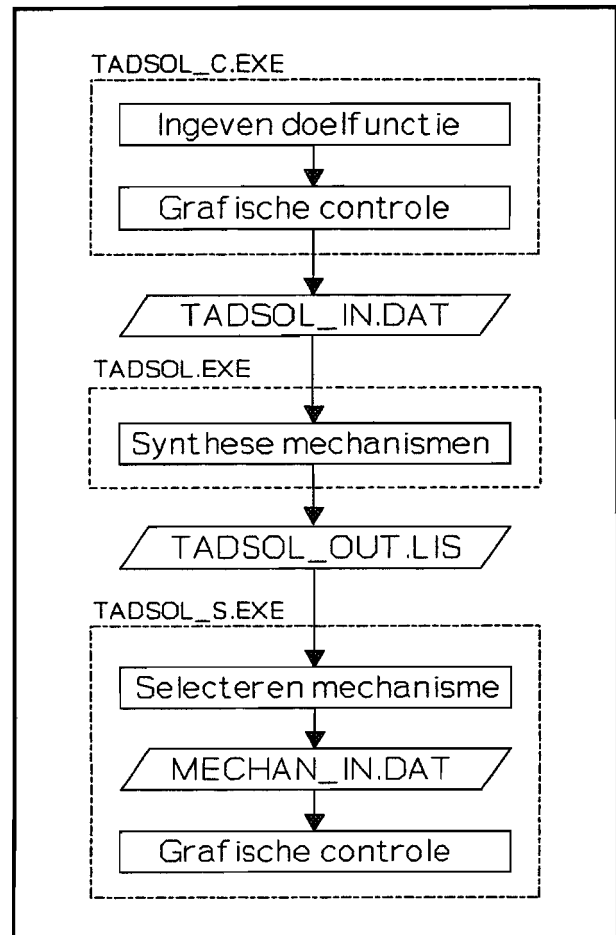
$\delta$  of  $S$  kan een roterend, slingerend of translierend zijn (zie figuur 2).

### 2.2.1. Werking TADSOL

TADSOL levert type, afmetingen en montageparameters van stangenmechanismen die een ingegeven doelfunctie bij benadering kunnen genereren. Deze mechanismen worden geselecteerd uit een catalogus met ongeveer 70 verschillende typen. Dit selecteren gebeurt in verschillende stappen. Ten eerste wordt de doelfunctie ontwikkeld in een Fourier-reeks. Daarna worden de Fouriercoëfficiënten van zo'n reeks vergeleken met die van de overdrachtsfuncties van de catalogusmechanismen. Op grond van het aanwezig zijn van bepaalde coëfficiënten wordt dan het type mechanisme gekozen (typesynthese) en met behulp van de waarden van die coëfficiënten worden de afmetingen van het mechanisme bepaald (maatsynthese). De montageparameters geven aan hoe de catalogusschets moet worden aangepast om tot de juiste montagestandschets te komen. Dit wordt nog uitgebreid besproken in hoofdstuk 6.

### 2.2.2. Opbouw TADSOL

TADSOL is geschreven in Fortran en bestaat uit 3 deelprogramma's (zie figuur 3). In het eerste deelprogramma TADSOL\_C.EXE (de extensie EXE staat voor executable) wordt het probleem gedefinieerd. Daartoe moeten het bewegingskenmerk, de toelaatbare onnauwkeurigheid en de doelfunctie ingegeven worden. De doelfunctie wordt ingegeven door middel van een aantal doelfunctiepunten. TADSOL genereert daarna een continue, periodieke spline-functie door deze punten. Deze doelfunctie kan ook in grafiekvorm op het scherm getoond worden. Het is van belang dat de ontwerper zich hier goed realiseert wat hij doet. Door in een bepaald traject meer punten op te geven zal de doelfunctie daar door het gevonden mechanisme beter benaderd worden. Ook kunnen punten, die voor het bewegingsverloop niet echt van belang zijn, een aantal mogelijke oplossingen uitsluiten. Daarnaast is het duidelijk dat naar mate de toelaatbare onnauwkeurigheid kleiner wordt gekozen het aantal oplossingen dat voldoet kleiner is. De invoergegevens worden weggeschreven in de datafile TADSOL\_IN.DAT.



Figuur 3. Stroomschema TADSOL

Het tweede deelprogramma TADSOL.EXE gebruikt TADSOL\_IN.DAT als invoer en schrijft zijn oplossingen vervolgens weg in de datafile TADSOL\_OUT.LIS. Hierin zijn ook de oplossingen die niet aan de nauwkeurigheidseis voldoen opgenomen.

Met het derde deelprogramma TADSOL\_S.EXE is het vervolgens mogelijk een selectie uit de oplossingen te maken waarvan de nulde, eerste en tweede orde overdrachtsfuncties bekeken kunnen worden. De datafile met geselecteerde oplossingen wordt MECHAN\_IN.DAT genoemd.

Uitgebreidere informatie over de werking en het gebruik van TADSOL is te vinden in [5] en [6]. Een voorbeeld van het gebruik van TADSOL is uitgewerkt in hoofdstuk 9. Voorbeelden van de genoemde datafiles zijn te vinden in bijlage 6 t/m 8.

### 2.3. UGTADSOL

Het huidige UGTADSOL bestaat uit een aantal GRIP-programma's dat het gebruik van TADSOL binnen UGII tot op zekere hoogte mogelijk maakt. Het hoofdprogramma UGTADSOL.GRS (de extensie GRS staat voor Grip Source File) zorgt voor het aanroepen van de verschillende deelprogramma's. Het deelprogramma V-ISO-3D.GRS zorgt voor het oproepen van de machinetekening en het weergeven hiervan in het V-ISO-3D-view. Daarna worden in het deelprogramma INTERFACE.GRS de interface-afspraken tussen machine en het te ontwerpen mechanisme (zoals de plaats van de krukas, plaats van de volgeras, referentielijnen en -richtingen) vastgelegd en in een aparte view op het scherm zichtbaar gemaakt. Vervolgens moet in het deelprogramma TADSOL.GRS de doelfunctie ingegeven worden. Daarna wordt de datafile TADSOL\_IN.DAT aangemaakt, TADSOL.EXE aangeroepen en de oplossingsgegevens uit TADSOL\_OUT.LIS ingelezen. Als laatste is het met het deelprogramma MONTAGESTAND.GRS mogelijk om van één specifiek mechanismetype, namelijk T008, een montagestandsheets in de machinetekening te laten maken.

Dit betekent dus dat van het oorspronkelijke TADSOL alleen het rekengedeelte gebruikt wordt. Het invoergedeelte is vervangen door een GRIP-programma met uitzondering van het gedeelte dat de gewenste doelfunctie op het scherm zichtbaar maakt. Het uitvoergedeelte, dat de overdrachtsfuncties van de gevonden oplossingen zichtbaar maakt, wordt in zijn geheel niet meer gebruikt.

### 3. PROBLEEMANALYSE

#### 3.1. Deelprojecten

Uit de opdrachtomschrijving en de voorafgaande oriëntatie zijn de volgende deelprojecten af te leiden.

##### 3.1.1. Herschrijven bestaande programma's

In januari 1993 is versie 10 van UGII beschikbaar gekomen. Hierin is onder andere een aantal GRIP-commando's gewijzigd of verdwenen. De bestaande, voor UGII versie 9 geschreven, UGTADSOL-programma's draaien daardoor niet op versie 10. Daarnaast heeft de leverancier van UGII laten weten dat de taal GRIP vanaf versie 11 niet meer ondersteund zal worden omdat ze met UFUNC verder willen gaan. Omdat de bestaande programma's dus toch aangepast moeten worden aan versie 10, lijkt het verstandig om dit meteen in UFUNC te doen. Het eerste deelproject is dus:

Herschrijf de bestaande UGTADSOL-programma's zodat ze op UGII versie 10 draaien, bij voorkeur met behulp van UFUNC.

##### 3.1.2. Grafische uitvoer op het scherm

Zoals gezegd is het met het bestaande UGTADSOL niet mogelijk om doelfuncties en overdrachtsfuncties op het scherm zichtbaar te maken. Om op een goede manier met TADSOL te kunnen werken is dit echter noodzakelijk. Je moet kunnen zien of uit doelfunctiepunten die je opgeeft wel de juiste doelfunctie berekend wordt. Ook moet je, om tot een juiste keuze uit de gegeven oplossingen te komen, de overdrachtsfuncties (nulde, eerste en tweede orde) van die oplossingen kunnen bekijken en vergelijken. Het tweede deelproject is dus:

Zorg ervoor dat de gewenste doelfunctiekromme en de bij de oplossingen behorende overdrachtsfuncties op het scherm getoond kunnen worden.

##### 3.1.3. Montagestandsheets

TADSOL op zich is een nuttig hulpmiddel bij het ontwerpen van stangenmechanismen. Een constructeur met weinig of geen kinematische kennis kan hiermee vrij eenvoudig tot een oplossing voor zijn bewegingsproblemen komen. Een groot nadeel hierbij is dat de oplossingsgegevens van TADSOL moeilijk te interpreteren zijn. Deze gegevens bestaan uit een lijst met getallen die in een moeizaam proces omgezet kunnen worden in een montagestandsheets. Om het gebruik van TADSOL te bevorderen

is het van belang dit proces automatisch te laten verlopen. De mogelijkheden van een CAD-systeem lijken daarbij van pas te komen.

Hier begint de integratie van TADSOL in UGII ook pas resultaat op te leveren. Het bestaande UGTADSOL zorgt ervoor dat is vastgelegd waar het mechanisme in de machine moet komen en zoals gezegd is het mogelijk om de montagestandschets van één mechanismetype te laten maken, maar bij nadere bestudering blijkt dat deze schets helemaal niet klopt. Het derde deelproject luidt dan ook:

Realiseer een methode waarmee UGTADSOL voor alle catalogusmechanismen de TADSOL-uitvoer omzet in de juiste montagestandschets in de machinetekening.

#### 3.1.4. Totale ontwerpproces

Met een montagestandschets zijn we natuurlijk nog niet aan het einde van het ontwerpproces. Er moet nog heel wat gebeuren voor er een volledig ontwerp van een werkend mechanisme is. Het belangrijkste daarbij is dat de schets moet worden omgezet in een 3-dimensionale geometrie.

Het vierde deelproject:

Onderzoek de mogelijkheden om het verdere verloop van het ontwerpproces te ondersteunen met behulp van UGTADSOL.

#### 3.1.5. Gebruikersvriendelijkheid

Het is belangrijk om een gebruikersvriendelijk programma te maken. De gebruiker moet zijn volle aandacht bij het ontwerpproces kunnen houden en niet afgeleid worden door allerlei onduidelijkheden, onmogelijkheden en onlogische stappen in het programma. Je moet wel opletten hier niet te veel tijd aan te besteden; het kan altijd mooier en beter, maar het draagt in feite niets bij aan de werking van het programma. Het laatste deelproject is dus:

Probeer de gebruikersvriendelijkheid van UGTADSOL te verbeteren zonder dat dit ten koste gaat van de overige deelprojecten.

### 3.2. Uitvoering deelprojecten

Hoewel de aangegeven deelprojecten niet volledig onafhankelijk van elkaar kunnen worden uitgevoerd (de gebruikersvriendelijkheid moet bij alle deelprojecten in de gaten gehouden worden; bij het herschrijven van de bestaande programma's is het handig om al naar de mogelijkheden van de grafische uitvoer te kijken), zullen ze hierna elk in een apart hoofdstuk behandeld worden.



## 4. HERSCHRIJVEN BESTAANDE PROGRAMMA'S

Om te beginnen is goed gekeken naar de werking en de opbouw van de bestaande programma's en geprobeerd deze vast te leggen in de vorm van stroomschema's. Omdat het geheel nogal ongestructureerd is geprogrammeerd en omdat een beschrijving van de programma's ontbreekt, is dit nog een flinke klus. Vervolgens is goed gekeken naar de mogelijkheden van de beschikbare programmeertalen GRIP en UFUNC.

### 4.1. GRIP.

GRIP is speciaal voor Unigraphics ontwikkeld. Het bestaat uit een aantal statements nodig voor de opbouw en het verloop van een programma zoals IF-, DO- en JUMP-statements. Daarnaast is er een groot aantal commando's dat de werking van het programma verzorgt door het aansturen van Unigraphics. Bij de overgang van UGII versie 9 naar versie 10 is het accent verschoven van wire-frames naar solids. Grip is echter niet meegegaan met deze ontwikkeling en is nog grotendeels gebaseerd op versie 9. GRIP is dus wel te gebruiken op versie 10 maar de nieuwe mogelijkheden van versie 10 zijn niet beschikbaar. De leverancier heeft in eerste instantie meegedeeld dat de ontwikkeling van GRIP helemaal wordt stopgezet, maar het laatste (inofficiële) nieuws is dat het toch weer verder ontwikkeld zal worden zodat het ook na versie 10 gebruikt kan worden.

### 4.2. UFUNC

UFUNC is geen complete programmeertaal. Het is een commandoset die voor de aansturing van Unigraphics-functies zorgt. Bij het gebruik van UFUNC programmeer je met een bestaande taal, Fortran of C. De UFUNC-commando's kunnen gewoon in het programma verwerkt worden waarna een speciaal link-programma voor de koppeling met Unigraphics zorgt. Het voordeel hiervan is dat de statements van Fortran of C voor de opbouw en het verloop van het programma veel krachtiger en uitgebreider zijn dan die van GRIP. Ook de UFUNC-commando's voor het aansturen van Unigraphics bieden meer mogelijkheden dan die van GRIP.

### 4.3. Afweging

In eerste instantie is gekozen om de bestaande programma's te herschrijven met behulp van UFUNC. UFUNC is krachtiger en wordt ook bij komende versies van UGII steeds aangepast zodat bestaande programma's bruikbaar blijven. Na een tijd met UFUNC te zijn bezig geweest, is hier toch van afgestapt. Het programmeren met UFUNC bleek nog veel problemen te geven en daardoor te veel tijd te kosten. De oorzaak hiervan is dat UFUNC in een overgangsfase zit. Voorheen was alles gericht op het gebruik

in combinatie met Fortran, terwijl het zich nu meer richt op C. Als gevolg daarvan zijn er verschillende typen commando's; alleen geschikt voor C, geschikt voor zowel C als Fortran, alleen geschikt voor Fortran, maar via een speciale macro ook in C-programma's te gebruiken. Het gebruik van deze typen is duidelijk verschillend. De documentatie hierover is echter niet toereikend en omdat ook EDS, de leverancier van Unigraphics, niet genoeg ondersteuning kon bieden bij het gebruik van UFUNC is besloten toch weer verder te gaan met GRIP. De consequentie hiervan is dat er minder gestructureerd geprogrammeerd kan worden (omdat GRIP minder krachtige commando's heeft, moet vaak met behulp van vreemde constructies geprogrammeerd worden), maar dat de bestaande programma's in ieder geval te gebruiken zijn op UGII versie 9.

Versie 9 is nog wel beschikbaar, maar wordt inmiddels bijna niet meer gebruikt. Het is dus wenselijk dat UGTADSOL ook op versie 10 draait. Omdat er bij de overgang van UGII versie 9 naar 10 een aantal GRIP-commando's is verdwenen of gewijzigd, was dit in eerste instantie niet het geval. Om het bestaande UGTADSOL te gebruiken in UGII versie 10 zijn daarom de nodige aanpassingen aangebracht.

## 5. GRAFISCHE UITVOER OP HET SCHERM

Om de grafische uitvoer op het scherm te realiseren zijn verschillende mogelijkheden bekeken.

### 5.1. Vervangen TADSOL

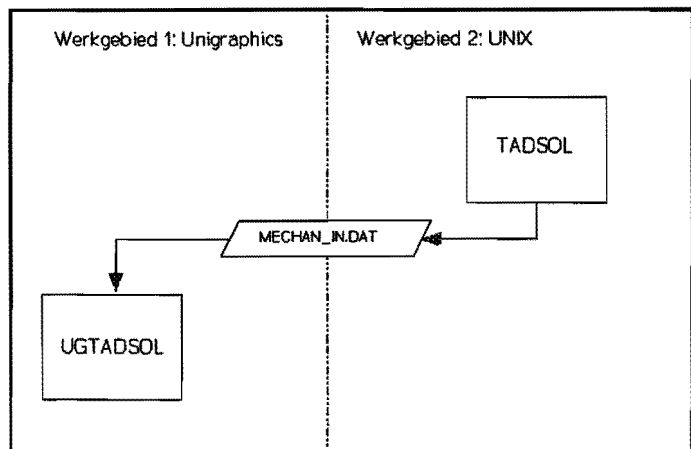
Het bestaande UGTADSOL is voor een groot gedeelte een vervanging van TADSOL. Het ingeven van de doelfunctie, het aanmaken van de invoerfile en het uitlezen van de uitvoerfile wordt door GRIP-programma's overgenomen. Een mogelijkheid is nu om het ontbrekende gedeelte (de grafische uitvoer) ook in GRIP te programmeren.

### 5.2. Aanroepen TADSOL

Van de andere kant gezien is er een bestaand programma (TADSOL\_S) dat de grafische uitvoer kan verzorgen. Het lijkt overbodig dit opnieuw te programmeren. Probeer dus TADSOL\_S te gebruiken door het vanuit UGTADSOL aan te roepen net zoals TADSOL.EXE wordt aangeroepen voor het rekenwerk. Een gedeelte van TADSOL\_S (het selecteren van oplossingen en het aanmaken van de selectiefile MECHAN\_IN.DAT) zal wel door GRIP-programma's overgenomen moeten worden. Ook zal TADSOL\_S zelf aangepast moeten worden om er op deze manier mee te werken. Een groter probleem hierbij is dat het niet mogelijk is om het Unigraphics-scherf te gebruiken voor uitvoer van externe programma's.

### 5.3. TADSOL los van Unigraphics

Het is onder UNIX mogelijk om meerdere werkgebieden te openen, zodat je verschillende programma's simultaan kunt draaien. Je kunt dus voor de uitvoer van TADSOL een tweede werkgebied openen. Je kunt ook het complete TADSOL-pakket los van Unigraphics gebruiken. In het ene werkgebied gebruik je TADSOL om daarna in het andere werkgebied de oplossingsgegevens daarvan verder te verwerken met behulp van UGTADSOL (zie figuur 4). Dit houdt in dat een gedeelte van de



*Figuur 4. TADSOL los van Unigraphics.*

bestaande UGTADSOL-programma's (het deelprogramma TADSOL.GRS) niet meer gebruikt wordt.

De voordelen hiervan zijn:

- TADSOL is op zich een goed werkend en gebruikersvriendelijk programma is. Het heeft meer correctiemogelijkheden en werkt veel vlotter dan de vervangende UGTADSOL-programma's.
- Het is niet meer nodig om voor de grafische uitvoer nog vervangende GRIP-programma's te schrijven.
- Er is een duidelijke programmastructuur met maar één communicatiepad tussen TADSOL en Unigraphics, de selectiefile MECHAN\_IN.DAT.

Vandaar dat is gekozen om deze ontkoppeling uit te voeren. UGTADSOL is aangepast zodat het MECHAN\_IN.DAT kan uitlezen en ook TADSOL zelf is enigszins aangepast. Nu zou je kunnen denken deze ontkoppeling de integratie van TADSOL in Unigraphics teniet doet. Maar dit deel van de integratie leverde helemaal geen voordelen op, de voordelen van het CAD-systeem werden hier nog niet gebruikt. Het maakte het alleen maar (op een omslachtige manier) mogelijk om in de Unigraphicsomgeving de TADSOL-invoer in te geven.

#### 5.4. Grafische drivers

Bij het gebruik van TADSOL onder UNIX is er echter nog een probleem. TADSOL is afkomstig van de TU in DELFT. Om de bruikbaarheid van de daar ontwikkelde software te vergroten, is het Graphical Mini System (GMS) ontworpen. Er wordt geprogrammeerd in een hogere taal, in dit geval Fortran. Zaken die dan nog problemen kunnen geven in verband met de uitwisselbaarheid, hebben betrekking op het operating system (file- en directory structuur) of de mogelijkheden van de hardware (b.v. grafische uitvoer). Om dit te ondervangen wordt zo onafhankelijk mogelijk geprogrammeerd. Voor hardware en operating system afhankelijke functies worden nauwkeurig omschreven subroutines gebruikt. Deze set subroutines wordt voor het draaien van de software aan het hoofdprogramma gekoppeld.

In eerste instantie waren alle subroutines om TADSOL op de Hewlett Packard/UNIX-combinatie waar Unigraphics op draait aanwezig behalve de driver-routines voor het scherm (het oorspronkelijke UGTADSOL maakt alleen gebruik van het rekengedeelte van TADSOL dat deze subroutines niet nodig heeft). Nu bleek dat onze werkstations gebruik maken van het X-windows systeem en dat daar in Delft routines voor ontwikkeld zijn. Deze routines maken gebruik van de programmeertaal van X-windows (Xlib) waarin alle functies voor het aansturen van het scherm geprogrammeerd kunnen worden. Maar omdat Xlib vanaf de laatste versie (X11) is gebaseerd op het gebruik in combinatie met C, zijn speciale files nodig om te zorgen dat de in Fortran geschreven routines Xlib kunnen gebruiken. Deze zijn opgevraagd bij Hewlett Packard en geïnstalleerd op het systeem. Daarna is het nog een hele puzzel geweest hoe alles aan elkaar gekoppeld moest worden en heeft het door allerlei kleine foutjes nog veel tijd gekost om het geheel werkend te krijgen maar uiteindelijk is dit toch gelukt.

Meer gedetailleerde informatie over GMS en het compileren van TADSOL is te vinden in bijlage 2.

## 6. MONTAGESTANDSCHETS

### 6.1 Oriëntatie

Bij de oriëntatie op dit deelproject wordt door middel van een aantal steekvragen het project nader omschreven:

#### Doelstelling

Het vereenvoudigen van het ontwerpproces van stangenmechanismen door middel van computerondersteuning.

#### Probleem

De uitvoergegevens van TADSOL zijn moeilijk te interpreteren.

#### Oorzaken probleem

- De uitvoergegevens bestaan uit een lijst met getallen die met de hand moeten worden omgezet in een schets.
- Er zijn 71 verschillende typen mechanismen.

#### Resultaat

Het aanbieden van verschillende oplossingen voor een bewegingsprobleem in de vorm van een montagestandsheets met bijbehorende overdrachtsfunctie. De montagestand van een mechanisme is de stand waarin het aangesloten wordt op de machine. De kruk moet in een bepaalde stand op de krukas gemonteerd worden en de uitgaande schakel op het aan te drijven machinedeel.

#### Systeemgrenzen

- Er wordt alleen gekeken naar functiegenererende stangenmechanismen.
- Wat betreft de synthese van oplossingen wordt niet gekeken naar andere pakketten dan TADSOL.
- UGTADSOL hoeft niet bruikbaar te zijn in combinatie met andere pakketten dan UGII.
- De toekomstige gebruiker van UGTADSOL is bekend met het gebruik van UGII.

#### Criteria

##### Eisen

- Het dubbel ingeven van informatie moet voorkomen worden. Informatie die in de CAD-tekening aanwezig is hoeft niet meer ingevoerd te worden.
- Van de gebruiker van UGTADSOL wordt een minimum aan kinematische kennis vereist.
- Het geheel moet gebruikersvriendelijk zijn.

- De programma's moet modulair van opbouw zijn zodat het eenvoudig aan te passen en uit te breiden is.
- De montagestandschets moet bestaan uit een duidelijk plaatje met de benaming van de verschillende onderdelen.
- Bij de montagestandschets moet een lijstje met afmetingen en montagehoeken van de verschillende onderdelen getoond worden.
- Het spreekt voor zich dat in alle gevallen de juiste stand van het mechanisme gegeven moet worden.
- UGTADSOL hoeft nog niet van alle typen mechanismen de montagestand te kunnen produceren. De methode waarmee dit gebeurt moet wel geschikt zijn voor alle typen.

#### Wensen

- Het is natuurlijk een wens dat UGTADSOL al wel de montagestand van alle typen mechanismen kan produceren.
- De montagestandschets moet geheel automatisch tot stand komen zonder tussenkomst van de ontwerper.
- De montagestandschets moet de mechanisme-definitie voor MECHANISMS bevatten zodat meteen een kinematische analyse uitgevoerd kan worden.
- Om de verschillende oplossingen makkelijk te vergelijken zouden de montagestandschetsen en de overdrachtsfuncties van een aantal oplossingen in één overzicht op het scherm te zien moeten zijn.

#### 6.2. Oplossingsgegevens

De TADSOL-uitvoer in de vorm van TADSOL\_OUT.LIS of MECHAN\_IN.DAT bestaat uit:

- het codenummer van het mechanisme (b.v. S006),
- de waarden van de kinematische parameters (PAR1, PAR2, enz.)
- de waarden van de montageparameters  $\tau$  en  $U_0$ , TAU en U0)
- het spiegelgetal L,
- de waarde van de grootste afwijking ERR.

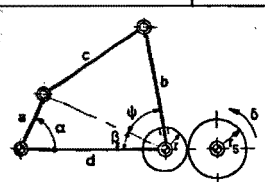
#### Codenummer

Dit codenummer bestaat uit een driecijferig volgnummer voorafgegaan door het bewegingskenmerk (R, S of T) en het verwijst naar een mechanismetype uit de catalogus. In deze catalogus zijn ongeveer 70 verschillende typen mechanismen opgenomen. Bij ieder type wordt gegeven (zie figuur 5):

- een schets met de topologie van het betreffende mechanisme,
- een lijst met parameters, hun formules en eventueel grenswaarden,
- de overdrachtsfuncties van de nulde, de eerste en de tweede orde,
- informatie over de Fouriercoëfficiënten.

MECHAN-UH-1 80

Delft University of Technology CADOM	Catalogue MECAT	Mechanism S006	Page 1(2)
---	--------------------	-------------------	--------------



Parameters:

- (1)  $0 < \lambda < \frac{\pi}{4} < 1$
- (2)  $i = 2 \frac{r}{r_2}$
- (3)  $\alpha = \frac{\pi}{4} > 0$
- (4)  $\lambda = \frac{\pi}{2} > 0$

Name:

Transfer FUNCTIONS:

$$\dot{\delta}(t) = -i \dot{\psi}_2(t) + \dot{\psi}(t)$$

$$\dot{\delta}^2(t) = -i \dot{\psi}_2^2(t) + \dot{\psi}^2(t)$$

$$\dot{\delta}^3(t) = -i \dot{\psi}_2^3(t) + \dot{\psi}^3(t)$$

tan  $\delta_2 = \frac{\lambda \sin \alpha}{1 - \lambda \cos \alpha}$

$$\dot{\psi}_2(t) = \rho^{-2} \lambda (\cos \alpha - 1)$$

$$\dot{\psi}_2(t) = \rho^{-2} \lambda (1^2 - 1) \sin \alpha$$

$$\cos \psi = \frac{\rho^2 r_2 \cos \alpha - \rho^2}{2 \rho r_2}; \quad \dot{\psi}(t) = \frac{-\lambda \sin \alpha}{\sin \psi} \left[ \frac{1}{2 \rho r_2} + \frac{\rho^2 r_2 \cos \alpha}{2 \rho r_2^2} \right]$$

$$\dot{\psi}^2(t) = \left[ \frac{\lambda \sin \alpha \cos \alpha - \rho^2}{\sin^2 \psi} + \frac{\lambda \cos \alpha}{\sin \psi} \right] \left[ \frac{1}{2 \rho r_2} + \frac{\rho^2 r_2 \cos \alpha}{2 \rho r_2^2} \right]^2 + \frac{\lambda^2 \sin^2 \alpha}{\sin^2 \psi} \left[ \frac{1}{2 \rho r_2^2} + \frac{\lambda (\rho^2 r_2 \cos \alpha)}{2 \rho r_2^3} \right]$$

$$\rho^2 = 1 + \lambda^2 - 2 \lambda \cos \alpha$$

MECHAN-UH-1 92

Delft University of Technology CADOM	Catalogue MECAT	Mechanism S006	Page 2(2)
---	--------------------	-------------------	--------------

Fourier coefficients, reference position:

$$B_k = -\frac{1}{2} \lambda^k \quad k = 2, 2, 3, \dots$$

Fourier configuration, reference position:

$C_0$	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$	$C_6$
2	1	1	2	2	2	2

shift criterion

$\beta_0$	$\beta_1$	$\beta_2$	$\beta_3$	$\beta_4$	$\beta_5$	$\beta_6$
3	0	3	-1	3	-2	3

Calculation of parameters:

- (1)  $\lambda = \frac{2 \rho r_2}{\rho^2}$
- (2)  $\lambda = -\frac{\rho^2}{2 \rho r_2}$
- (3)  $\psi$  and  $(\lambda) \times$  from special positions

References:

ANKERS, H.-  
Massbestimmung der Kurbelschwinge unter Verwendung der harmonischen Analyse der vorgegebenen periodischen Bewegung.  
Techn.Mitt.Neus der Techn. 55.Jahrgang(1962) Heft 7, S. 309-316.

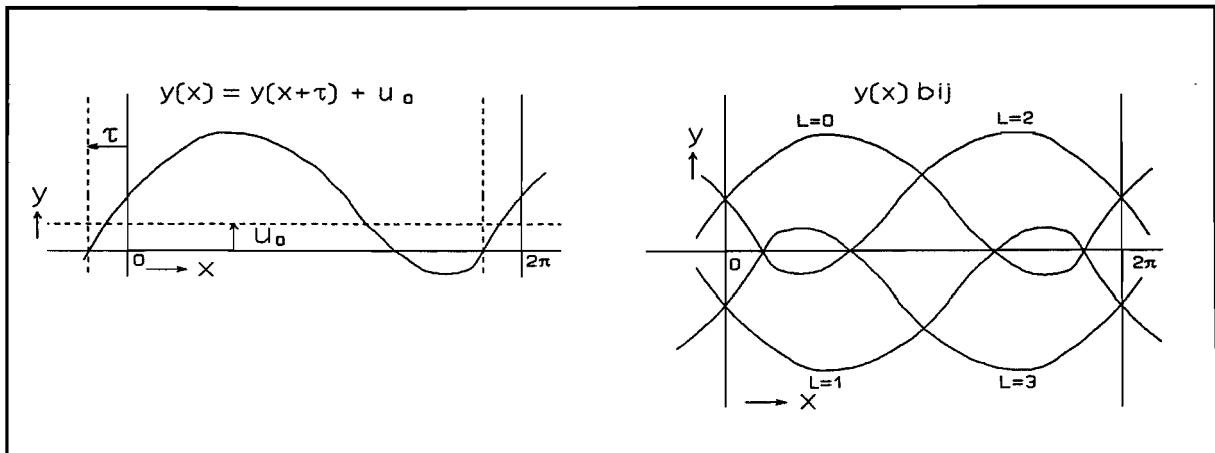
Figuur 5. Catalogusmechanisme S006.

Kinematische parameters

De kinematische parameters bepalen de kinematische afmetingen van het mechanisme. Voor mechanismen met een translaterende uitgang zijn dit de absolute waarden van de afmetingen. Bij een roterende of slingerende uitgaande beweging worden alleen de verhoudingen tussen de afmetingen vastgelegd door de kinematische parameters. De absolute waarden worden bepaald op grond van constructieve eisen.

Montageparameters en Spiegelgetal

Bij het zoeken naar de beste benadering van de doelfunctie zoekt TADSOL ook naar benaderingen van een langs het coördinatensysteem verschoven doelfunctie. De grootte van de verschuiving langs de abscis wordt aangegeven met  $\tau$ , de verschuiving langs de ordinaat met  $U_0$ . Dit schuiven gebeurt niet alleen voor de oorspronkelijke functie, maar ook voor de gespiegelde functies. Er zijn drie spiegel mogelijkheden: ten opzichte van de abscis (spiegelgetal  $L=1$ ), ten opzichte van de ordinaat ( $L=2$ ) of ten opzichte van beide ( $L=3$ ). Bij de niet gespiegelde functie is  $L=0$  (zie figuur 6). Omdat deze grootheden bepalen op welke wijze het mechanisme in de machine moet worden gemonteerd, worden dit de montageparameters genoemd.



Figuur 6. Vershoven en gespiegelde functies.

### Grootste afwijking

Bij iedere oplossing wordt ook een ERR-waarde gegeven. Deze is gelijk aan de maximale absolute verschil tussen de geleverde overdrachtsfunctie en de gewenste doelfunctie. Als dit verschil gedeeld door de maximale slag van de volger groter is dan de van te voren opgegeven toelaatbare onnauwkeurigheid, verschijnt er een min-teken voor de ERR-waarde. Dit kan dienen als een eerste selectie criterium voor de gegenereerde oplossingen.

### 6.3. Van oplossingsgegevens tot montage schets

De montage schets in de machine komt met behulp van gegevens uit drie informatiebronnen tot stand:

- de TADSOL-uitvoer,
- de mechanismecatalogus,
- de machinetekening.

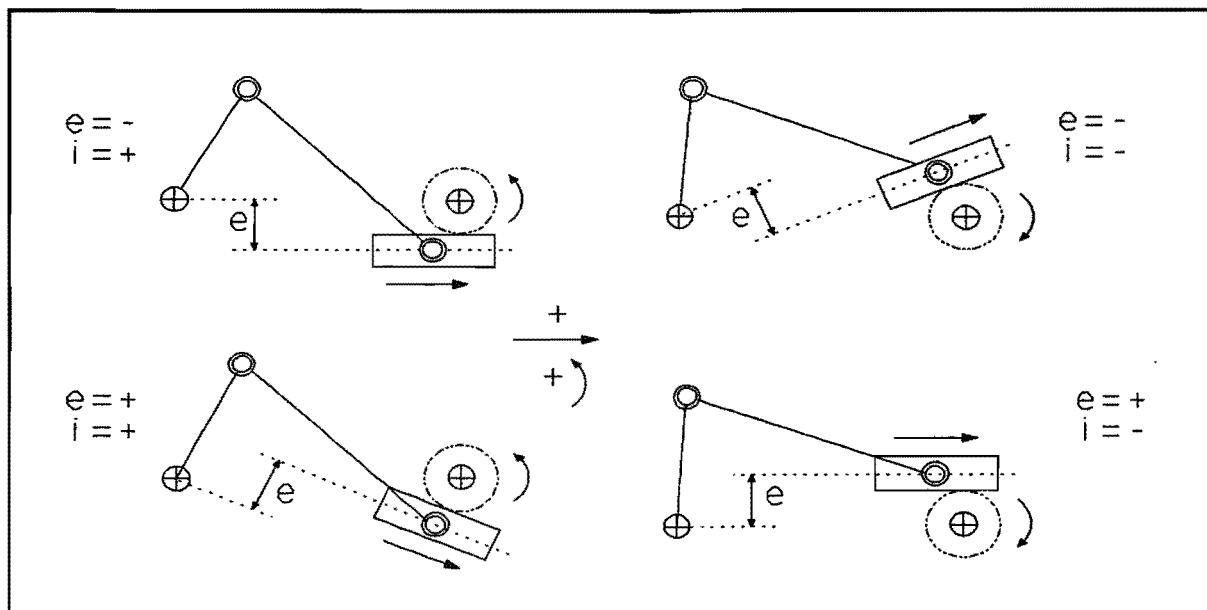
Zoals gezegd bepalen de kinematische parameters de kinematische afmetingen van het mechanisme. Daarmee is het mogelijk een catalogus-schets op schaal te maken.



### Bijstellen van de catalogusschets

#### a. Corrigeren van een tandwieloverbrenging en/of excentriciteit.

Dit is nodig als het teken van de parameters van deze overbrenging of excentriciteit in de TADSOL-uitvoer niet overeenkomt met het teken in de catalogusschets.



Figuur 7. Verschillende mogelijkheden excentriciteit ( $e$ ) en overbrengverhouding ( $i$ ) S008.

#### b. Toevoegen van een referentielijn.

In het geval dat de uitgaande schakel een tandwiel is, ontbreekt in de catalogusschets de referentielijn voor die uitgaande schakel. Er moet dan zowel een referentielijn als een referentiestraal op het tandwiel zelf gekozen worden.

### Transformeren van de bijgestelde catalogusschets

#### c. Transformeren van het coördinatenstelsel uit de catalogus.

Omdat we niet op zoek zijn naar een oplossing voor een verschoven of gespiegelde doelfunctie moet het coördinatenstelsel van de catalogusschets zodanig verschoven en gespiegeld worden dat de overdrachtsfunctie in het nieuwe coördinatenstelsel overeenkomt met de oorspronkelijke doelfunctie. Het verschuiven van het coördinatenstelsel komt overeen met het verplaatsen van de referentielijnen van de catalogusschets, het spiegelen met het omkeren van de referentierichtingen:

- Met  $r$  wordt een nieuwe referentielijn voor de krukhoek bepaald door  $\alpha = -r$  uit te zetten in de schets. De nieuwe krukhoek  $\alpha' = \alpha + r$ ,
- Bij  $L=1$  of  $L=3$  moet de positieve bewegingsrichting van de uitgaande schakel, zoals in de catalogus aangegeven, omgekeerd worden,

- Bij  $L=2$  of  $L=3$  moet de positieve bewegingsrichting van de nieuwe krukhoek  $\alpha'$  omgekeerd worden,
- Met  $U_0$  wordt een nieuwe referentielijn voor de rotatiehoek of de verplaatsing van de uitgaande schakel bepaald door  $\beta=-U_0$  of  $s=-U_0$  uit te zetten.

d. Transformeren van de schets in doelfunctiecoördinaten naar de montagestandschets in machinecoördinaten.

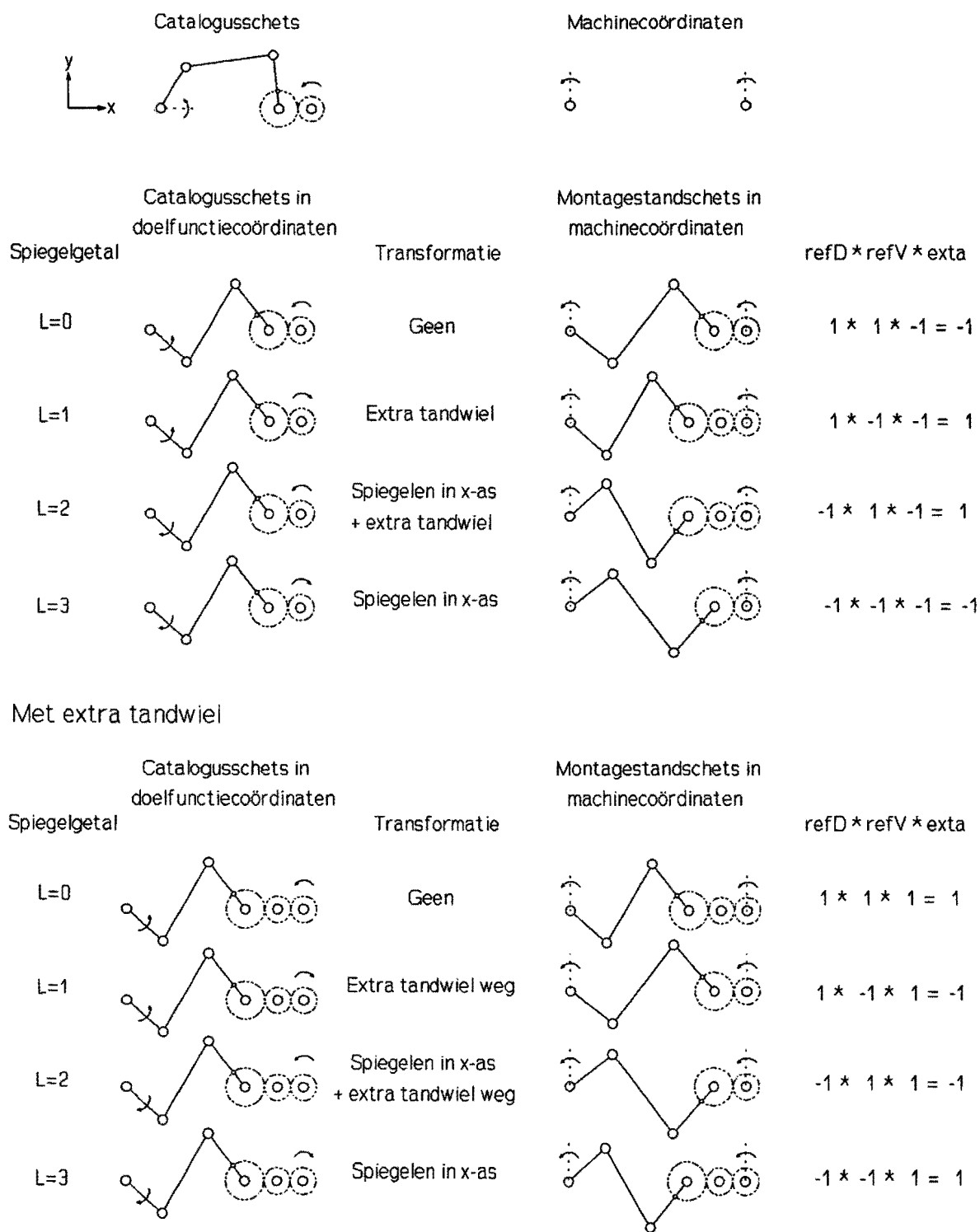
De machinecoördinaten volgen rechtstreeks uit de interface-afspraken. Als laatste moet de schets nu nog zo aangepast worden dat hij in die machinecoördinaten past. Het is niet mogelijk om door middel van een eenvoudig recept de noodzakelijke transformaties voor deze stap voor te schrijven. Er zal met enige inventiviteit gekeken moeten worden naar de mogelijke transformaties en de consequenties daarvan op de positie van het mechanisme en de overdrachtsfunctie. Transformatie van het mechanisme kan verkregen worden door:

- spiegelen om een lijn,
- draaien om een vast punt,
- transleren langs een lijn,
- schalen, dit is alleen bij roterende en slingerende mechanismen mogelijk zonder dat dit gevolgen heeft voor de overdrachtsfunctie.

## 6.4. Automatische montagestandschets

### 6.4.1. Transformatieproces

In figuur 8 is het transformatieproces voor catalogustype S006 samengevat. Het corrigeren van overbrengverhouding (a) wordt gedaan door een extra tandwiel toe te voegen (onderste vier gevallen). Het toevoegen van de referentielijn (b) is al gebeurd bij het ingeven van de interface-afspraken door het ingeven van een referentielijn voor de volgeras. De catalogusschets in doelfunctiecoördinaten ontstaat uit de catalogusschets door de kruk op de positie van de nieuwe referentielijn ( $\alpha=-\tau$ ) uit te zetten en de referentierichtingen aan te passen aan het spiegelgetal (c). Daarna moet de schets nog worden getransformeerd naar machinecoördinaten (d). Omdat dit in eerste instantie moeilijk te automatiseren was, zijn twee vereenvoudigingen aangebracht. Ten eerste is ervoor gezorgd dat het coördinatensysteem van waaruit de schets getekend wordt de juiste positie en oriëntatie heeft zodat de schets daarna niet meer getransleerd of geroteerd hoeft te worden. Dit is hier gedaan door aan het eind van INTERFA-CE.GRS het coördinatensysteem in het midden van de krukas te plaatsen met de z-as in richting van de krukas en x-as gericht naar de volgeras (of evenwijdig aan de volgerbaan bij translerende mechanismen).



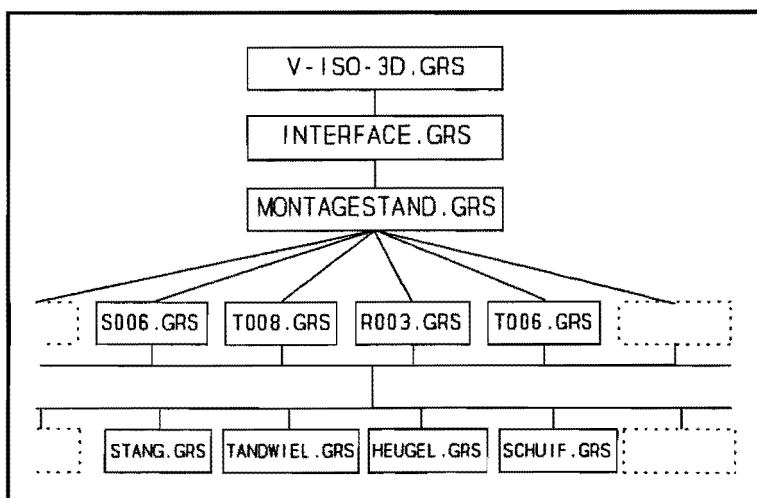
Figuur 8. Transformaties catalogustype S006.

Ten tweede is het aantal mogelijke transformaties met een factor vier verkleind door bij het ingeven van de interface-afspraken te eisen dat de referentierichtingen van krukas en volgeras beide linksom positief zijn. De gebruiker zal zijn doelfunctie hieraan moeten aanpassen. Het transformeren zelf gebeurt nu door toevoegen of weghalen van een tandwiel waardoor de draairichting van de uitgaande as omkeert of door het spiegelen van het geheel in de x-as waardoor de draairichting van beide assen omkeert. Met combinaties van deze twee operaties zijn alle schetsen te transformeren naar de machinecoördinaten. Het blijkt dat er acht mogelijkheden zijn afhankelijk van het spiegelgetal en het teken van de overbrengverhouding die twee aan twee dezelfde montagestand opleveren.

Het spiegelgetal wordt omgezet in waarden voor refD en refV en het teken van de overbrengverhouding in een waarde voor EXTA. Dit is niet echt noodzakelijk maar maakt het wat overzichtelijker. Het blijkt nu dat de montagestandschets drie tandwielen bevat als het produkt van deze drie referentiewaarden +1 is. Daarnaast blijkt dat het mechanisme gespiegeld moet worden als de waarde van refD -1 is. Als laatste moet de positie van de uitgaande as berekend worden aan de hand van de bij het mechanisme horende overdrachtsfunctie gecorrigeerd met  $U_0$ . Dit is in figuur 8 niet meer aangegeven. Op grond van deze gegevens kan de montagestandschets uitgetekend worden.

#### 6.4.2. Programma's

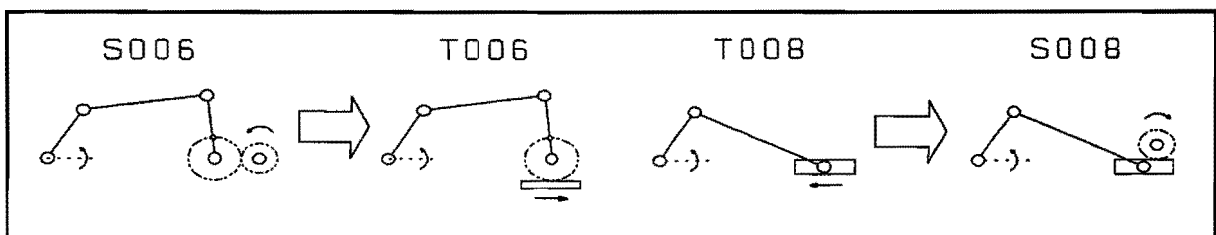
Om de omvang van de programma's te beperken is het van belang zoveel mogelijk stappen van het transformatieproces onder te brengen in een type-onafhankelijk gedeelte. Dit is op twee manieren gedaan. Ten eerste is een aantal gemeenschappelijke stappen opgenomen in het deelprogramma MONTAGE-  
STAND.GRS (het inlezen en tonen van de oplossingsgegevens, het instellen van kleur, layer en andere zaken, het aanroepen van het type-afhankelijke gedeelte en het



Figuur 9. Structuur UGTADSOL. S006.

eventueel wissen van de schets). Ten tweede is een aantal subroutines geschreven voor het tekenen van de verschillende onderdelen van de mechanismen (stang, tandwiel/rondsel, scharnier, schuif en heugel). De functie van het type-afhankelijke gedeelte is dan het berekenen van de posities van de onderdelen aan de hand van de oplossingsgegevens en het aanroepen van de subroutines voor het uittekenen van deze onderdelen.

Voor het type-afhankelijk gedeelte is geen algemeen recept te geven. Het is wel mogelijk de verschillende typen mechanismen in te delen in categorieën van gelijksoortige typen. Zo zijn een heel stel mechanismen gebaseerd op het kruk-slinger mechanisme (S006). Een andere groep mechanismen hebben het kruk-sleuf mechanisme (T008) als basis. De programma's die de montagestandschets verzorgen zijn binnen zo'n categorie vrij eenvoudig van het programma van het basistype af te leiden. Ook is het zo dat de S-typen en de T-typen in paren voorkomen. Door het toevoegen van een heugel veranderd een mechanisme met een tandwiel als slingerende uitgang in een mechanisme met een transliserende uitgang. Door het vervangen van de schuif van een transliserend mechanisme door een heugel met rondsel wordt het een mechanisme met slingerende uitgang (zie figuur 10).



Figuur 10. T- en S-mechanismen komen in paren voor.

De stap van het ene naar het andere type betekent voor al deze paren een overeenkomstige aanpassing aan de programma's. Zo is in bijlage 4 een overzicht van het transformatieproces van catalogustype T006 opgenomen. Dit proces is vrijwel gelijk is aan dat van S006. Het enige verschil is dat het teken van de overbrenging hier wordt gecorrigeerd door de heugel aan de andere zijde van het rondsel te plaatsen. Een tabel met een overzicht van de relaties tussen de verschillende typen is te vinden in bijlage 4. Op dit moment zijn de programma's voor de basistypen (S006 en T008) gerealiseerd evenals de daarvan afgeleide programma's voor de typen R003 en T006. Overzichten van de gebruikte variabelen en de opbouw van de programma's zijn eveneens opgenomen in bijlage 4.

Op deze manier kunnen ongeveer 25 mechanismen redelijk eenvoudig geprogrammeerd worden. Het blijkt dat bij bijna elk bewegingsprobleem wel één van deze typen een goede oplossing is. Daarnaast zijn de overige typen constructief vrij ingewikkeld waardoor ze toch al minder vaak toegepast worden. Dit wil niet zeggen niet gebruikt kunnen worden, TADSOL blijft gewoon oplossingen met deze typen geven, het is alleen niet mogelijk om er met behulp van UGTADSOL een montagestandschets van te maken.

In hoofdstuk 9.4. zijn voor het daar behandelde voorbeeld alle mogelijke montagestandschetsen uitgetekend, waarbij naast het spiegelgetal en het teken van de overbrengverhouding ook het teken van  $r$  wordt gevarieerd.

De automatische mechanisme-definitie voor MECHANISMS is met GRIP helaas niet mogelijk. GRIP

bevat geen commando's voor het aansturen van MECHANISMS. Een andere mogelijkheid zou kunnen zijn om voor ieder type een "standaard-definitie" op te slaan en deze later aan de schets te koppelen, maar dit bleek ook niet mogelijk te zijn omdat de definitie en schets als één geheel worden weggeschreven.

#### 6.4.3. Alternatief

Een eerste idee voor het maken van de montagestandschets was om te werken met geparameteriseerde schetsen van de verschillende catalogustypen. Een GRIP-programma zou er dan voor moeten zorgen dat de juiste schets opgeroepen wordt en dat de afmetingen daarvan worden aangepast aan de berekende parameters. UGII biedt de mogelijkheid om met behulp van constraints en expressies verbanden tussen dimensies vast te leggen om zo een geparameteriseerde schets te maken. Het is daarbij helaas niet mogelijk om kinematische verbanden in de schets aan te brengen. Dit betekent dat bij aanpassen van de afmetingen van de stangen de verbanden tussen die stangen verloren gaan. Het is daarom niet mogelijk deze methode te gebruiken.

## 7. TOTALE ONTWERPPROCES

Het totale ontwerpproces van stangenmechanismen is weergegeven in figuur 11 (naar [9]). Volgens de projectstrategie bestaat het uit een oriëntatie- een plan- en een uitvoerings-proces (naar [4]). Een korte toelichting bij de stappen in dit ontwerpproces:

### 1. Opstellen deeltaak

Uit de het eisenpakket voor een te ontwerpen machine is een aantal deeltaken af te leiden. Een zo'n deeltaak kan het genereren van een complexe beweging zijn. Hierin worden onder andere omschreven: toerental en draairichting van de ingaande as, bewegingseisen, beschikbare inbouwruimte en belastingen als gevolg van de uit te voeren taak.

### 2. Definiëren doelfunctie

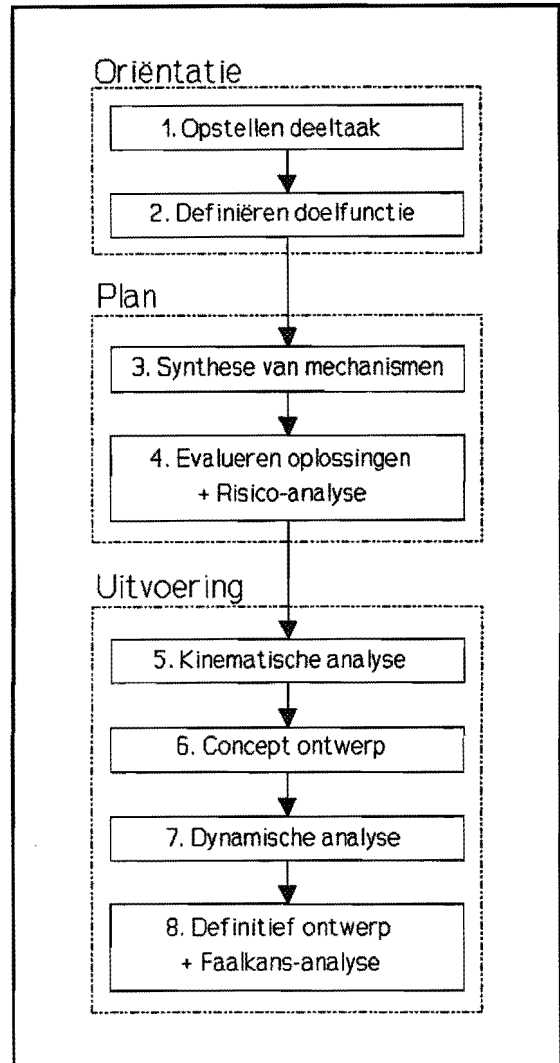
Uit de bewegingseisen wordt een doelfunctie met bijbehorende nauwkeurigheidseisen afgeleid. Voor de synthese van principe-oplossingen is dit de enige input.

### 3. Synthese van mechanismen

In deze fase worden alle stangenmechanismen die aan de doelfunctie kunnen voldoen gegenereerd. Het resultaat van deze synthese is het type en de afmetingen van mogelijke oplossingen.

### 4. Evalueren van oplossingen

Meestal zullen er meerdere kinematische schetsen als oplossing mogelijk zijn. Tijdens deze evaluatie kunnen oplossingen afvallen omdat ze bijvoorbeeld teveel inbouwruimte vragen of constructief te gecompliceerd zijn. Uiteindelijk zal er een keuze gemaakt moeten worden voor één van de oplossingen. Daarna dient er een risico-analyse te volgen waarbij gezocht moet worden naar oorzaken van buitenaf waardoor de oplossing niet zou kunnen voldoen.



Figuur 11. Ontwerpproces stangenmechanismen.

#### 5. Kinematische analyse

Er kan nu beter gekeken worden naar posities, snelheden en versnellingen van de verschillende schakels van de schetsoplossingen.

#### 6. Concept-ontwerp maken

De schetsoplossing die aan alle kinematische eisen voldoet moet nu omgezet worden in een 3-dimensionaal ontwerp. Hier kunnen nog problemen ontstaan door bijvoorbeeld het bewegen van twee stangen in hetzelfde vlak.

#### 7. Dynamische analyse

Nu de schakels massa en stijfheid hebben gekregen kan het conceptontwerp beoordeeld worden op optredende krachten.

#### 8. Definitief ontwerp maken

Het ontwerp wordt tot in detail uitgewerkt. Als laatste zal een faalkans-analyse uitgevoerd moeten worden. Daarbij moet nagegaan worden wat de nadelige gevolgen zijn van het falen van de verschillende functies waarbij de kans hierop wordt ingeschat. Het ontwerp wordt zonodig aangepast.

### 7.1. UGTADSOL

Het toekomstige UGTADSOL zou het gehele ontwerpproces moeten kunnen ondersteunen. Het zal daarbij een soort raamwerk zijn waarbinnen het ontwerpproces zich afspeelt, met Unigraphics als basis:

- a. Voor bepaalde delen van het proces kan gebruik gemaakt worden van bestaande software-pakketten. De functie van UGTADSOL daarbij is om deze pakketten te koppelen aan Unigraphics en ervoor te zorgen dat informatie die in het systeem aanwezig is niet bij ieder pakket opnieuw ingegeven moet worden.
- b. Voor andere delen is geen bestaande software voorhanden. UGTADSOL zal de hiervoor benodigde programmatuur moeten bevatten.
- c. Tenslotte zijn er delen die niet of met te veel beperkingen kunnen worden geautomatiseerd. Hier moet het werk overgelaten worden aan de constructeur.

#### Ad a.

Zoals bekend is kan voor de synthese van mechanismen (stap 3) TADSOL worden gebruikt. Dit pakket werkt buiten de Unigraphics-omgeving zodat UGTADSOL de communicatie moet verzorgen.



De kinematische analyse (stap 5) kan uitgevoerd worden met de Unigraphics-module MECHANISMS. Hiermee kan het bewegingsverloop van mechanismen geanalyseerd worden. Het levert een bewegingssimulatie of grafieken van verplaatsing, snelheid of versnelling van elk punt van het mechanisme. De taak van UGTADSOL is hier het aanbrengen van mechanismedefinities op de montagestandschets.

Voor de dynamische analyse (stap 7) zou het rekenpakket ADAMS gebruikt kunnen worden. Dit pakket draait binnen MECHANISMS. UGTADSOL zou hier een aantal aanvullende definities (massa's, stijfheden) moeten toevoegen.

#### Ad b.

Het definiëren van de doelfunctie (stap 2) zou je gedeeltelijk kunnen automatiseren door in de machinetekening de posities die de uitgaande schakel op bepaalde momenten moet innemen aan te geven en deze om te laten rekenen tot een aantal doelfunctiepunten.

Bij het gebruik van TADSOL voor de synthese (stap 3) wordt, zoals bekend, UGTADSOL gebruikt voor het omzetten van de oplossingsgegevens in een montagestandschets.

Het maken van het concept ontwerp (stap 6) zou je ook gedeeltelijk kunnen automatiseren. Een stangenmechanisme is opgebouwd uit een beperkt aantal basiselementen (met stangen, tandwielen en lagers zijn al vrij veel typen op te bouwen). Je zou voor het bepalen van de afmetingen van deze elementen een aantal algoritmen kunnen ontwerpen. Daarna kun je de resultaten hiervan gebruiken om de maten van parametrisch ingevoerde elementen aan te passen.

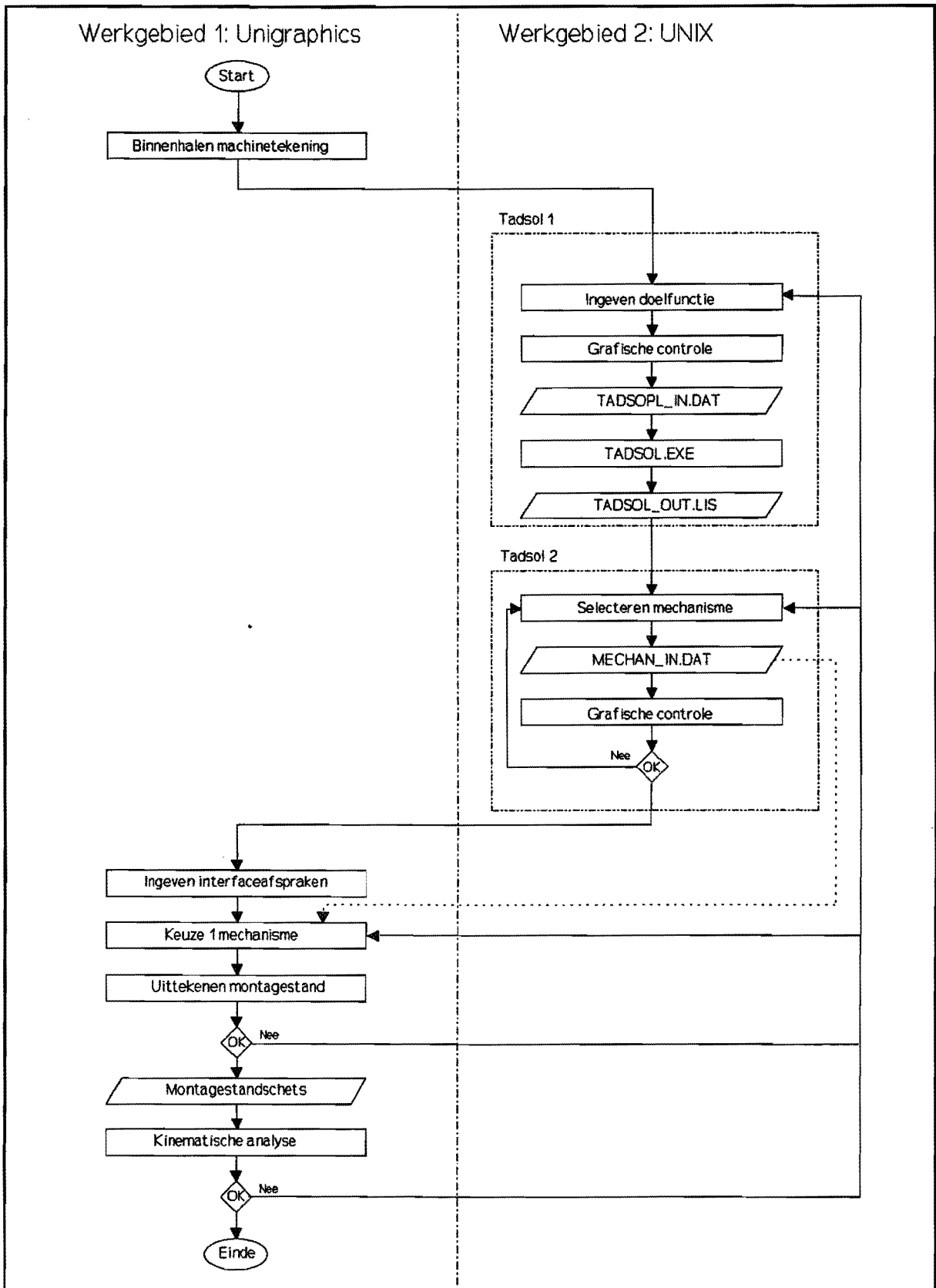
#### Ad c.

Het opstellen van de deeltaak (stap 1) zal altijd het werk van de constructeur blijven, al zal dit voor een deel met behulp van de machinetekening op het CAD-systeem gebeuren.

Een verantwoorde keuze uit de verschillende oplossingen zal je ook niet automatisch kunnen laten verlopen. Het vergelijken van de gerealiseerde overdrachtsfuncties van de principe-oplossingen met de doelfunctie (stap 3), het evalueren van die oplossingen met risico-analyse (stap 4) en het interpreteren van de kinematische en dynamische analyse (stap 5 en 7) zal de ontwerper zelf moeten doen.

Het maken van het definitieve ontwerp met faalkans-analyse (stap 8) blijft ook handwerk. Dit is veel te complex om te automatiseren. Het is natuurlijk wel logisch om hierbij het CAD-systeem te gebruiken.

Het huidige UGTADSOL ondersteunt de stappen 3 t/m 5 zoals is weergegeven in figuur 12. Stap 3 (de synthese van mechanismen) wordt uitgevoerd door TADSOL, waarna UGTADSOL de kinematische schets levert. De kinematische analyse kan uitgevoerd worden met de Unigraphics-module MECHANISMS, waarbij de mechanismedefinities nog met de hand moeten worden aangebracht. Het werken met UGTADSOL wordt met een voorbeeld verder toegelicht in hoofdstuk 9.



Figuur 12. Stroomschema huidige UGTADSOL.

## 8. GEBRUIKERSVRIENDELIJKHEID

Ten aanzien van de gebruikersvriendelijkheid zijn verschillende verbeteringen aangebracht:

- Zoals eerder gezegd, is de gebruikersvriendelijkheid van het originele TADSOL veel beter dan die van de vervangende UGTADSOL-programma's. Er zijn meer correctiemogelijkheden en het werkt veel vlotter.
- Deze gebruikersvriendelijkheid is nog verbeterd door het automatisch aanroepen van de verschillende deelprogramma's van TADSOL. Daarnaast is er een aantal voor deze toepassing overbodige keuzemogelijkheden binnen TADSOL verwijderd.

Van het oorspronkelijke UGTADSOL wordt alleen het gedeelte om de interface-afspraken in te geven nog gebruikt. Maar ook hierin is een aantal aanpassingen aangebracht:

- Bij iedere keuzemogelijkheid in een GRIP-programma worden de opties "BACK" en "CANCEL" getoond. Binnen het programma kan aangegeven worden wat te doen bij zo'n commando. Als je dit consequent doet, kun je de gebruikersvriendelijkheid aanzienlijk verhogen. Zo is het nu op ieder moment mogelijk om een stap terug te doen in het programma of om het ontwerpproces af te breken.
- Het invoeren van de krukas en de volgeras/volgerbaan is vereenvoudigd door hier een aantal logische default-mogelijkheden in te bouwen. Het is uiteraard mogelijk hiervan af te wijken.
- Het oorspronkelijke UGTADSOL werkte hinderlijk traag. Door een aantal maatregelen loopt het nu veel vlotter:
  - UGTADSOL gaf vrij veel mededelingen op het scherm met tekst en uitleg. Deze zijn nu samengevoegd of ingekort, zodat alleen nog de noodzakelijke informatie wordt gegeven.
  - Er werd voortdurend gewisseld tussen verschillende views. Omdat dit weinig extra informatie gaf gebeurt dit nu niet meer automatisch. De gebruiker kan natuurlijk altijd zelf een andere view kiezen.
  - Er moest een aantal gegevens ingegeven worden dat maar in enkele gevallen gebruikt werd. Nu hoeven die gegevens alleen maar in die gevallen ingegeven te worden.

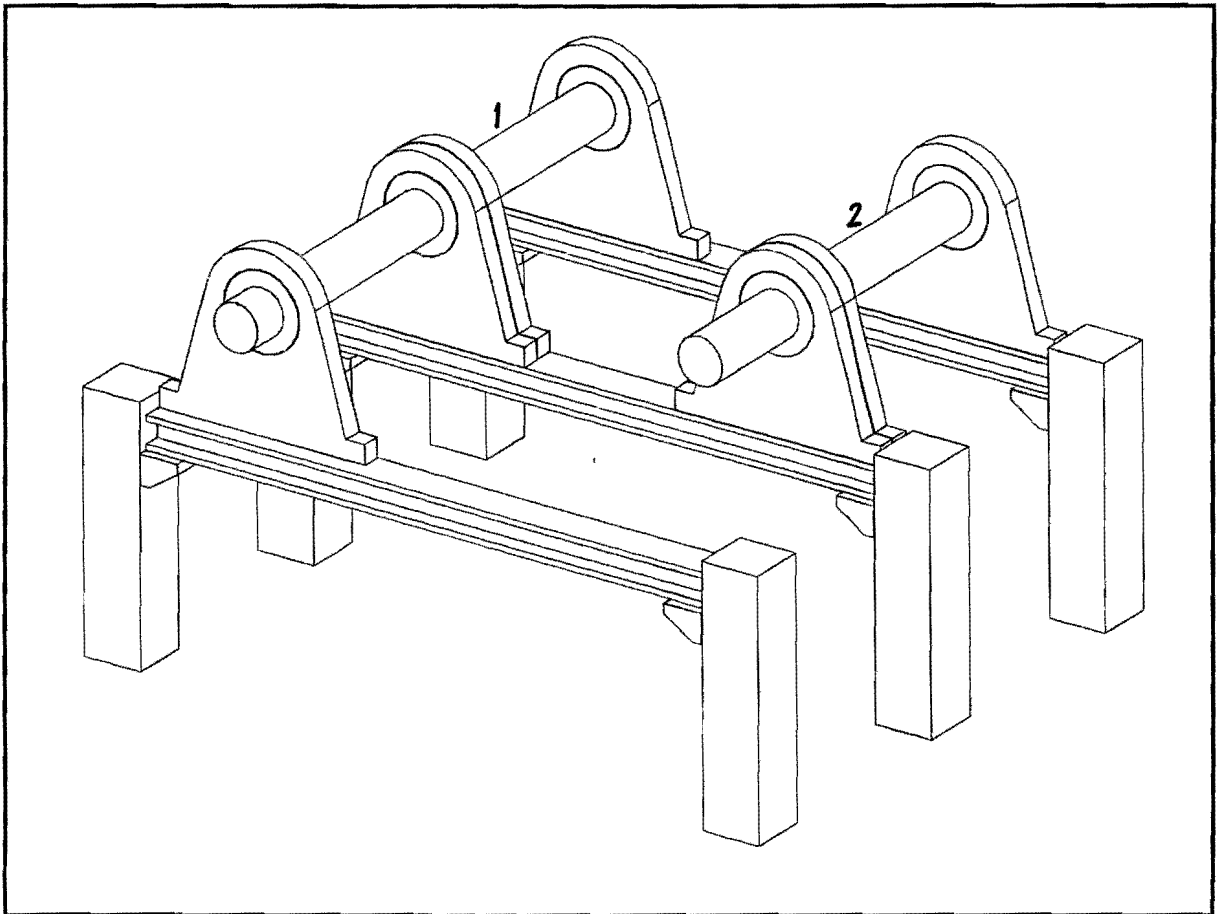
Al met al is de gebruikersvriendelijkheid van UGTADSOL een stuk toegenomen maar nog lang niet ideaal. In dat geval zou je TADSOL moeten kunnen draaien op een apart window in hetzelfde werkgebied als Unigraphics. Ook de presentatie van de oplossingen zou nog verbeterd kunnen worden. Nu moet je in eerste instantie nog kiezen op grond van een file met getallen (TADSOL\_OUT.LIS), die daarna omgezet worden in een overdrachtsfunctie en een montagestandsheets. Ideaal zou zijn als er meteen een overzicht gegeven zou worden van de oplossingen in de vorm van overdrachtsfuncties met bijbehorende sheets.

## 9. VOORBEELD GEBRUIK UGTADSOL

### 9.1. Bewegingsprobleem

In een denkbeeldige machine (figuur 13) moet voor as 2 een bepaald bewegingsverloop gerealiseerd worden, afgeleid van de eenparig roterende as 1. As 2 moet daarbij een slingerende beweging maken waaraan de volgende eisen worden gesteld:

- constante rotatiesnelheid over een hoek van  $40^\circ$ ,
- deze snelheid moet gelijk zijn aan die van as 1,
- maximale rotatie van  $90^\circ$ .



Figuur 13. Machinetekening.

### 9.2. TADSOL

Voor de synthese van oplossingen voor dit probleem moet nu TADSOL opgestart worden. Dit gebeurt door de batch-file "start" aan te klikken. Na het ingeven van het bewegingskenmerk (in dit geval

S(lingerend)) en de toelaatbare onnauwkeurigheid (de default-waarde van 5% is hier voldoende) moet de doelfunctie gedefinieerd worden. De bewegingseisen moeten daarvoor omgezet worden in doelfunctiepunten. De eisen in dit voorbeeld worden omgezet in de punten van tabel 1. Punt 3 t/m 9 beschrijven een rotatiehoek van 40° met constante snelheid. Punt 1 en 2 begrenzen de slag aan de ene kant en punt 8 en 9 aan de andere kant.

Punt	Hoek as 1 (°)	Hoek as 2 (°)
1	0	0
2	10	0
3	50	20
4	60	30
5	70	40
6	80	50
7	90	60
8	130	80
9	140	80

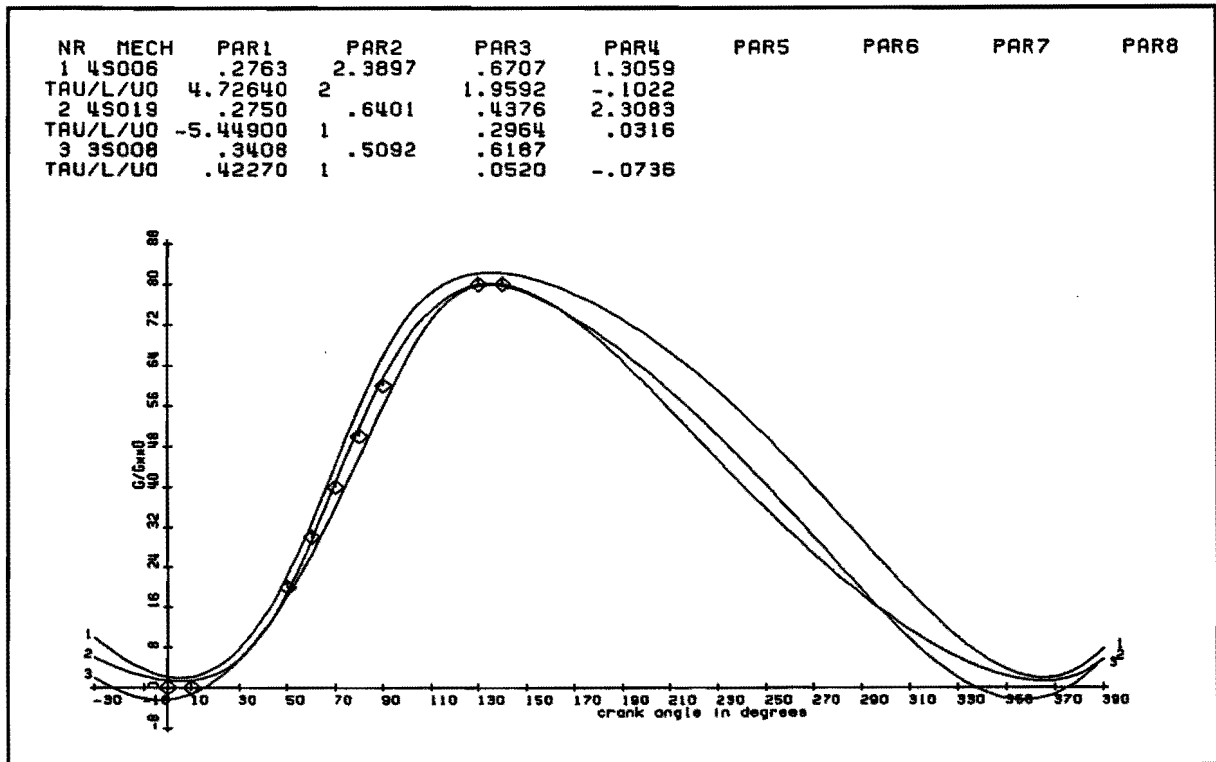
Daarna kan de functie die door deze punten is berekend in grafiekvorm op het scherm getoond worden en op grond daarvan kunnen de punten eventueel worden gewijzigd. Nu worden deze invoergegevens weggeschreven in de data-file TADSOL\_IN.DAT (zie bijlage 6) en start het rekengedeelte van TADSOL. De oplossingsgegevens worden weggeschreven in de data-file TADSOL\_OUT.LIS (zie bijlage 7) en er wordt gemeld hoeveel oplossingen er berekend zijn (in dit geval 22). Nu moet een selectie gemaakt worden uit de oplossingen. Dit gebeurt in een aantal stappen.

Tabel 1. Doelfunctiepunten.

Ten eerste kan TADSOL\_OUT.LIS bekeken worden en kan op grond van de ERR-waarde een eerste selectie gemaakt worden. Let hierbij op niet te veel oplossingen te schrappen. De ERR-waarde geeft slechts het maximale absolute verschil tussen geleverde overdrachtsfunctie en gewenste doelfunctie. Als maar een gedeelte van de doelfunctie van belang is voor het probleem, kan een oplossing met een negatieve ERR-waarde toch voldoen in dat gedeelte. Het is handig om bij deze eerste selectie ook de catalogus te raadplegen. Zo kunnen mechanismen die om constructieve redenen afvallen (te ingewikkeld, ongewenste onderdelen) meteen verwijderd worden.

De geselecteerde mechanismen worden verzameld in de data-file MECHAN\_IN.DAT (zie bijlage 8). MECHAN\_IN.DAT is een append-file, dat wil zeggen dat deze bij het maken van een nieuwe selectie niet overschreven wordt. Nieuw geselecteerde oplossingen worden eraan toegevoegd en al geselecteerde oplossingen blijven erin staan totdat ze verwijderd worden. Het voordeel hiervan is dat oplossingen van verschillende probleemdefinities met elkaar vergeleken kunnen worden.

Vervolgens kunnen de overdrachtsfuncties van de geselecteerde mechanismen op het scherm getoond worden en worden vergeleken met de doelfunctie zodat je een gegronde keuze uit de oplossingen kunt maken. In dit voorbeeld zijn nu de drie oplossingen (oplossingnummers 14, 16, 21 uit TADSOL\_OUT.LIS) gekozen die de doelfunctie het beste benaderen. De nulde orde overdrachtsfuncties daarvan zijn te zien in figuur 14, de eerste en tweede orde overdrachtsfuncties staan in bijlage 9. Je kunt hier overigens op ieder moment terug om een andere selectie uit de oplossingen te maken of helemaal terug naar het begin om de doelfunctie te wijzigen.



Figuur 14. Nulde orde overdrachtsfuncties met doelfunctiepunten.

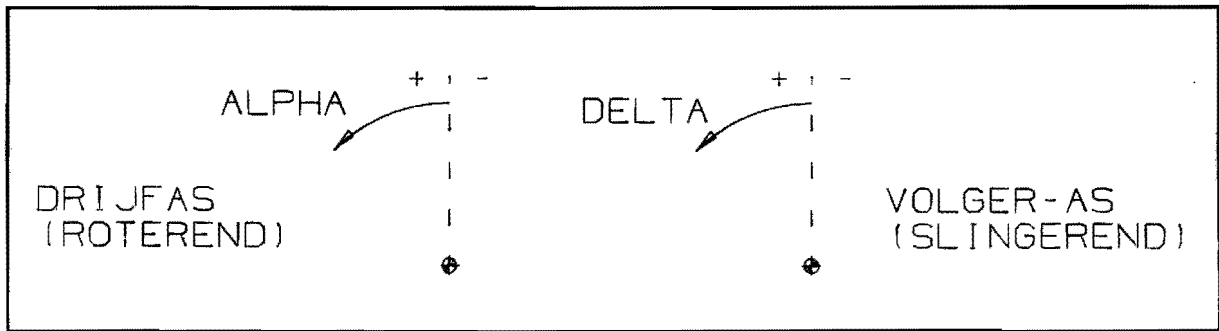
Je kunt bij het gebruik van TADSOL natuurlijk ook andere namen kiezen in plaats van TADSOL\_IN.DAT, TADSOL\_OUT.LIS en MECHAN\_IN.DAT, zodat je bestaande files niet steeds overschrijft en later nog eens kunt gebruiken.

Meer informatie over het gebruik van TADSOL is te vinden in [2].

### 9.3. UGTADSOL

Nu kunnen met behulp van UGTADSOL montagestandschetsen van de geselecteerde oplossingen gemaakt worden. Daartoe moet in een tweede werkgebied onder Unigraphics UGTADSOL.GRX gestart worden. Nadat de machinetekening is binnengehaald moeten de interface-afspraken ingegeven worden. Bij dit voorbeeld houdt dat in dat beide assen, de positie op de assen waar het mechanisme moet komen en de referentielijnen aangegeven moeten worden (Dit geldt ook voor mechanismen met een roterende uitgang, bij een translerende uitgang moet in plaats van de tweede as een volgerbaan gedefinieerd worden). Deze afspraken worden dan afgebeeld in het zogenaamde mechanismevlak (zie figuur 15).

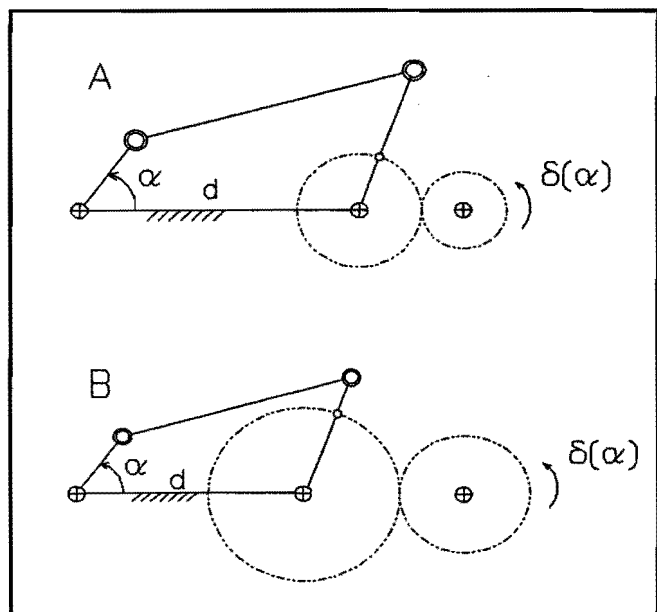
Daarna wordt MECHAN\_IN.DAT op het scherm getoond en kun je daaruit een oplossing kiezen waarvan vervolgens een montagestandschets in het mechanismevlak en in de machinetekening getekend wordt. Daarbij verschijnt ook een lijstje met de afmetingen en montageparameters. Ook hier kunnen nog oplossingen afvallen omdat ze bijvoorbeeld niet in de beschikbare ruimte passen.



Figuur 15. Interface-afspraken.

Veel mechanismetypen zijn niet volledig bepaald door de parameters. Er is nog een afmeting vrij te kiezen, zoals bij type S006 de lengte van stang  $d$ . In figuur 16 staan twee mechanismen met verschillende lengte  $d$ . Maar ze zijn getekend vanuit dezelfde parameters en geven dus ook dezelfde overdrachtsfunctie. UGTADSOL geeft ook de mogelijkheid het eindpunt van stang  $d$  vrij te kiezen, de overige afmetingen worden dan automatisch aangepast.

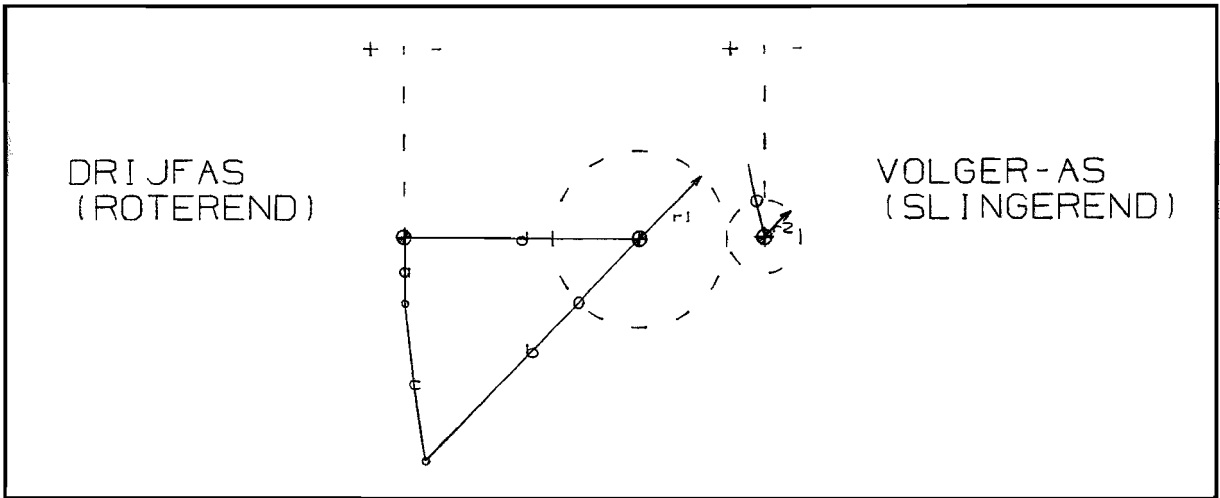
Vervolgens kun je de schets eventueel wissen en een andere oplossing uit MECHAN\_IN.DAT kiezen. Ook kun je terug naar TADSOL om een andere MECHAN\_IN.DAT aan te maken en hier een oplossing uit kiezen. In dit voorbeeld valt de keuze uiteindelijk op oplossing 1 (type S006). De gerealiseerde overdrachtsfunctie is wat minder dan die van oplossing 2, maar de constructie is een stuk eenvoudiger. In figuur 17 en 18 is de montagestandschets van deze oplossing te zien en in figuur 19 het lijstje met afmetingen.



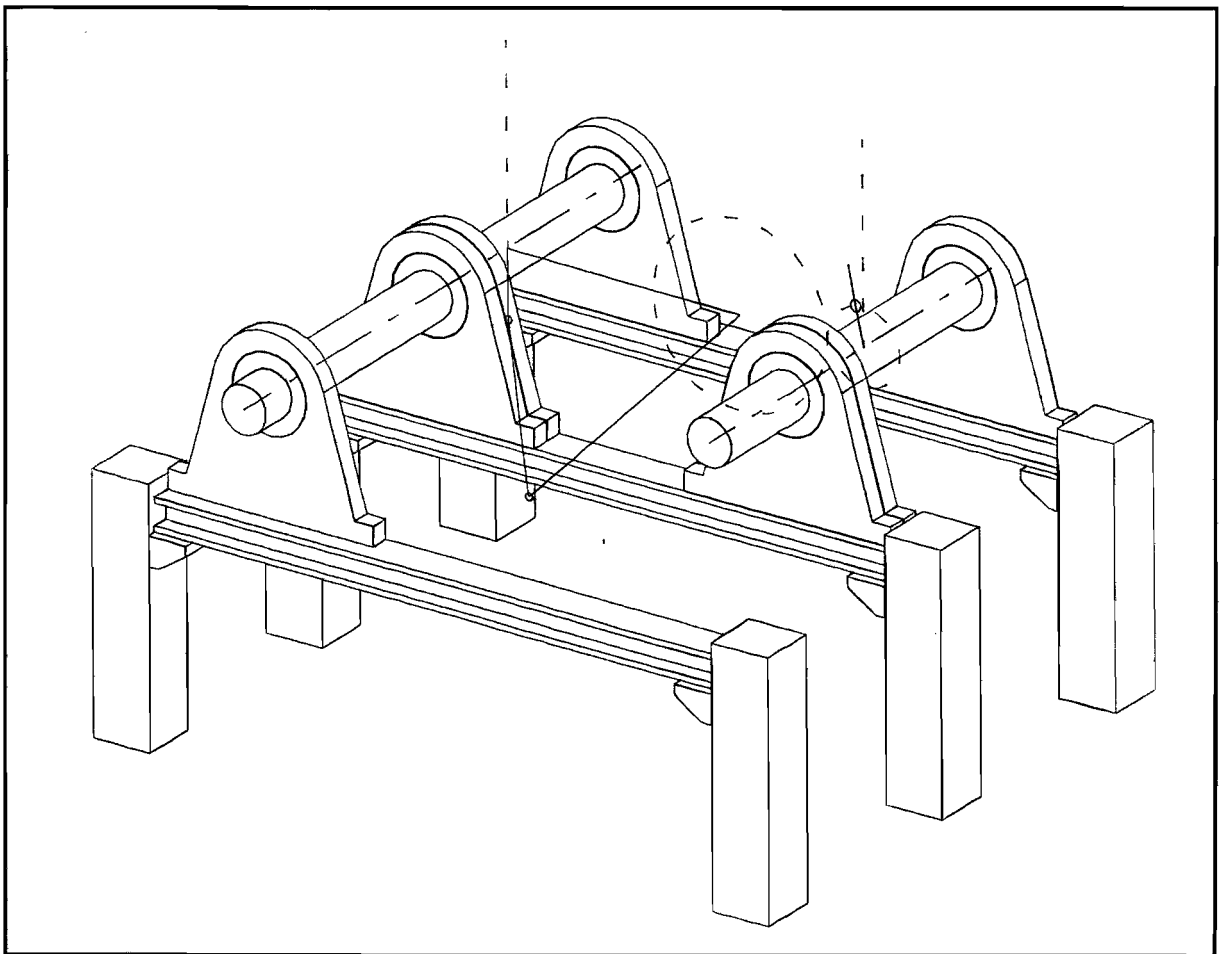
Figuur 16. Mechanismen met verschillende stanglengte  $d$ .

Hier houden de mogelijkheden van het huidige UGTADSOL op. Het is nog wel mogelijk om met behulp van MECHANISMS een kinematische analyse uit te voeren, maar dit wordt niet ondersteund door UGTADSOL.





Figuur 17. Montagestandsheets in mechanisめvlak.



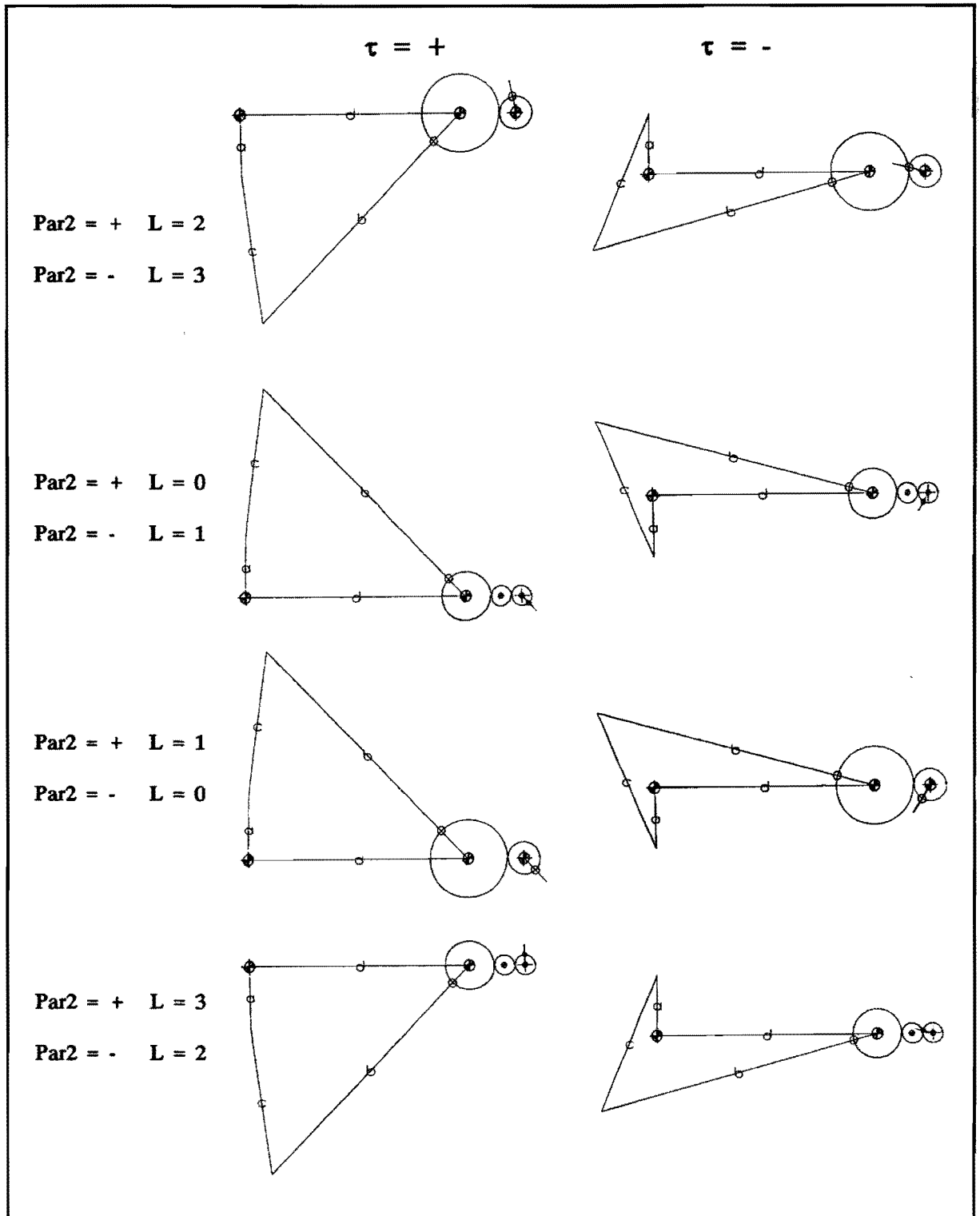
Figuur 18. Montagestandsheets in machinetekening.

OPLOSSING NR. 3			
MECHANISMETYPE S006			
STANGLENGTE	a = 34.37	STANGLENGTE	c = 83.44
STANGLENGTE	b = 162.47	STANGLENGTE	d = 124.41
MONTAGEHOEK KRUK	tm = 180.43	MONTAGEHOEK VOLGERAS	um = 2.11
TANDWIELSTRAAL	r1 = 46.25	TANDWIELSTRAAL	r2 = 19.35
TANDWIELVERHOUDING	r1/r2 = 2.39		

*Figuur 19. Lijstje met afmetingen en montageparameters.*

#### 9.4. Verschillende montagestanden

In figuur 20 zijn alle mogelijke montagestanden van catalogustype S006 getekend bij gelijke waarden van de kinematische parameters. Dat wil zeggen dat alleen het teken van  $\tau$  en de overbrengverhouding en het spiegelgetal gevarieerd worden. Dit levert in principe 16 mogelijkheden op, maar omdat ze twee aan twee gelijk zijn leidt dit tot 8 verschillende montagestanden. Hiervoor zijn de parameters van het voorgaande voorbeeld gebruikt. De montagestand linksboven is gelijk aan de oplossing uit het voorbeeld.



Figuur 20. Verschillende montagestanden S006.

## 10. CONCLUSIES EN AANBEVELINGEN

### 10.1. Conclusies

- Door het herschrijven van de oorspronkelijke UGTADSOL-programma's is het nu mogelijk deze onder Unigraphics versie 10 te gebruiken.
- De voortgang van het onderzoek wordt sterk geremd door het voortdurend aanpassen van de software aan nieuwe versies van Unigraphics en/of besturingssysteem.
- Het gebruik van User Functions biedt waarschijnlijk veel mogelijkheden maar is vanwege de gebrekkige documentatie nog niet aan te bevelen.
- De grafische uitvoer van TADSOL op het scherm is mogelijk gemaakt door de grafische drivers voor onze werkstations op de juiste manier te installeren.
- Het is niet mogelijk gebleken een algemene methode te geven waarmee UGTADSOL voor alle catalogustypen een montagestandschets kan maken. Het is wel mogelijk om een aantal basisprogramma's te schrijven waarvan programma's van ongeveer 25 typen vrij eenvoudig afgeleid kunnen worden.
- Het toekomstige UGTADSOL zou het gehele ontwerpproces moeten kunnen ondersteunen. Het zal daarbij voor bepaalde delen van het proces de koppeling tussen een aantal bestaande pakketten en Unigraphics moeten verzorgen en voor andere delen zelf de benodigde programmatuur moeten bevatten.
- De gebruikersvriendelijkheid van UGTADSOL is een stuk toegenomen maar nog altijd voor verbetering vatbaar.
- De oorspronkelijke UGTADSOL-programma's hebben de nodige aanpassingen ondergaan:
  - Het deelprogramma V-ISO-3D.GRS voor het binnenhalen van de machinetekening is onveranderd gebleven.
  - Het deelprogramma INTERFACE.GRS voor het ingeven van de interface-afspraken is ten aanzien van de gebruikersvriendelijkheid verbeterd.
  - Het deelprogramma TADSOL.GRS werd gebruikt om de invoergegevens voor het rekenge-deelte van TADSOL aan te maken. Omdat het gehele TADSOL-pakket nu los van UGII wordt gebruikt is dit niet meer nodig.
  - Het deelprogramma MONTAGESTAND.GRS maakte een (onjuiste) montagestandschets

van één catalogustype en is nu vervangen door een aantal programma's dat op dit moment montagestandschetsen van vier verschillende typen kan produceren.

## **10.2. Aanbevelingen**

- Bij verbeterde documentatie van UserFunctions is het zeker de moeite waard hier nog eens naar te kijken. Het biedt in tegenstelling tot GRIP bijvoorbeeld wel de mogelijkheid om MECHANISMS aan te sturen.
- UGTADSOL is op het moment nog niet praktisch bruikbaar omdat te weinig catalogustypen automatisch kunnen worden omgezet in een montagestandschets. Zorg er daarom voor dat dit mogelijk wordt voor de genoemde 25 typen waarbij dit vrij eenvoudig kan.
- Met het huidige UGTADSOL wordt maar een gedeelte van het totale ontwerpproces ondersteund. Het omzetten van de montagestandschets naar een 3-dimensionale geometrie (concept-ontwerp) zou je ook kunnen automatiseren. Bekijk wat hier de mogelijkheden zijn op het gebied van algoritmisch ontwerpen.

**LITERATUURLIJST**

- [1] Leenders, J.M.N., *Computerondersteund ontwerpen van vlakke stangenmechanismen*, Afstudeerverslag WPA-1123.  
Technische Universiteit Eindhoven, juli 1991.
- [2] Leenders, J.M.N., *Gebruikershandleiding TADSOL*, Bijlage afstudeerverslag WPA-1124.  
Technische Universiteit Eindhoven, mei 1991.
- [3] Niens, O.F.M., *TADSOL-integratie in Unigraphics II*, Afstudeerverslag WPA-1575.  
Technische Universiteit Eindhoven, augustus 1993.
- [4] Bragt, J.M. van, *Projectstrategie*, Collegedictaat 4697  
Technische Universiteit Eindhoven, oktober 1989.
- [5] Dijk, A. van, *CAD van stangenmechanismen, van probleemdefinitie tot montagestand (1)*.  
In: De constructeur, september 1984.
- [6] Dijk, A. van, *CAD van stangenmechanismen, van probleemdefinitie tot montagestand (2)*.  
In: De constructeur, oktober 1984.
- [7] Klein Breteler, A.J., *User Manual GMS 2.1*.  
Technische Universiteit Delft, november 1993.
- [8] Mikkers, P.J.C., *Energie en beweging, Deel: Grondbeginselen van de toegepaste kinematica*,  
Collegedictaat 4521.  
Technische Universiteit Eindhoven, 1988.
- [9] Klein Breteler, A.J., *Ontwerpen van mechanismen*, Collegedictaat w76.  
Technische Universiteit Delft, 1989.
- [10] Grinsven, J.H.A. van, *De MECHANISMS-module van Unigraphics*, Verslag onderzoekopdracht  
WPA-0909.  
Technische Universiteit Eindhoven, mei 1990.
- [11] *FORTTRAN/9000 Reference, Series 700 Computers*  
Hewlett Packard Company, 1991.

**BIJLAGEN**

## Bijlage 1. Opdrachtschrijving

### EINDSTUDIEOPDRACHT

TECHNISCHE UNIVERSITEIT EINDHOVEN  
Faculteit Werktuigbouwkunde  
Vakgroep WPA

december 1993

Student : H.J.M. van Abeelen  
Hoogleraar : Prof.ir. J.M. van Bragt  
Begeleider : Ir. A.T.J.M. Smals (TUE)  
Adviseur : Dr.ir. A.J. Klein Breteler (TU Delft)  
Start : 10 november 1993  
Titel : TADSOL-integratie in Unigraphics II

#### Toelichting

Bij het ontwerpen wordt tegenwoordig door de constructeur steeds vaker gebruik gemaakt van CAD/CAM-systemen. In vakgroep WPA wordt er naar gestreefd te ontwerpen met het UGII-systeem. Eén van de ontwikkelingen die plaats vindt is de vereenvoudiging van het ontwerpproces van mechanismen d.m.v. CAD-ondersteuning.

TADSOL is een op de TU Delft ontwikkeld pakket dat geschikt is voor de synthese van mechanismen waaronder stangenmechanismen. Bij een gebruiksvriendelijke integratie van TADSOL in het UGII-systeem kan het gebruik van stangenmechanismen worden bevorderd in de bouw van produktiemiddelen. De eerste aanzet hiervoor is gegeven door Leenders, waarna Niens de huidige koppeling, UGTADSOL genaamd, heeft gerealiseerd.

#### Projectresultaat

Een verbeterd, aangepast en uitgebreid UGTADSOL:

- Gebbruiksvriendelijk voor de ontwerper achter UGII.
- Alle functies van TADSOL beschikbaar, waaronder het tonen op scherm van doelfunctiekromme en overdrachtsfunctie van geschikte mechanismen.
- Alle catalogusmechanismen van TADSOL aanwezig.
- De mechanismen kunnen parametrisch worden ontworpen.  
Dit houdt in:-statische en dynamische analyse.  
-ontwerpen en dimensioneren.

#### Opdracht

Realiseer a en b.

Realiseer voor c een procedure waarmee de mechanismen op eenvoudige maar correcte wijze ingevoerd kunnen worden. Voer deze procedure uit voor tenminste één mechanisme.

Onderzoek d.

#### Verslag etc.

Ga te werk volgens de bij de secretaresse te verkrijgen instructies:

- "Afstuderen in de Produktietechnologie en Automatisering".
- "Wat moet waar en hoe in het verslag" door P. Westendorp.

Prof.ir. J.M. van Bragt

Ir. A.T.J.M. Smals

H.J.M. van Abeelen



## Bijlage 2. Grafische drivers

De subroutines van het Graphical Mini System zijn onderverdeeld in drie categorieën:

- STRINGS voor string-operaties,
- FILES voor file-operaties,
- GMS voor graphics.

Deze drie categorieën zijn hiërarchisch opgebouwd; STRINGS is het laagste en GMS het hoogste (subroutines uit STRINGS gebruiken nooit subroutines uit FILES en GMS, enzovoort).

Het GMS is zo opgebouwd dat je kunt programmeren onafhankelijk van de uitvoereenheid (plotter of scherm). Afhankelijk van welke eenheid geactiveerd is roepen de GMS-routines de bijbehorende driver-routines aan. GDRAW is bijvoorbeeld de GMS-routine om een lijn te tekenen; als het scherm is geactiveerd, zorgt GDRAW ervoor dat G0DRAW (de driver-routine voor het scherm) wordt aangeroepen. De overeenkomstige driver-routine voor de plotter heet G1DRAW. De gebruiker moet dus de beschikking hebben over STRINGS-, FILES-, en GMS-routines en bovendien over driver-routines voor de specifieke uitvoereenheid die hij wil gebruiken.

In eerste instantie waren alle subroutines om TADSOL op de Hewlett Packard/UNIX-combinatie waar Unigraphics op draait aanwezig behalve de driver-routines voor het scherm. Nu bleek dat onze werkstations gebruik maken van het X-windows systeem en dat daar in Delft routines voor ontwikkeld zijn. Deze routines maken gebruik van de programmeertaal van X-windows (Xlib) waarin alle functies voor het aansturen van het scherm geprogrammeerd kunnen worden. Maar omdat Xlib vanaf de laatste versie (X11) is gebaseerd op het gebruik in combinatie met C, zijn headerfiles (xfalias.h en xflib.h) om te zorgen dat de in Fortran geschreven routines Xlib kunnen gebruiken. Deze zijn opgevraagd bij Hewlett Packard en nu aanwezig op het systeem onder:

```
usr/include/X11R4/xf11/xfalias.h
usr/include/X11R4/xf11/xflib.h
```

Om tot een werkend programma (executable) te komen zal al deze software nog gecompileerd en gelinkt moeten worden. De driver-routines voor X-terminals zijn verzameld in de source-file g0X.F. Bij het compileren van deze routines moet aangegeven worden welke software daarbij nodig is:

- De gecompileerde GMS-, FILES-, en STRINGS-subroutines.
- De GMS-routines moeten ook driver-routines voor een plotter kunnen aanroepen. Vandaar dat hier de gecompileerde driver-routines voor een HPGL-plotter worden aangegeven.

- Omdat de G0X-routines gebruik maken van Xlib moeten ook de bibliotheken waarin deze commando's te vinden zijn aangegeven worden. Deze bibliotheken zijn te vinden onder:

```
usr/lib/X11R4/libxf11.a
```

```
usr/lib/X11R4/libx11.sl
```

Het aangeven van een bibliotheek gebeurt met het -l commando. Daarvoor moet in de VUEPROFILE (user personal environment variables) een pad aangegeven worden, in dit geval dus:

```
LPATH = usr/lib/X11R4
```

Het compile-commando luidt dan:

```
f77 -c g0X.F GMS.o FILES.o STRINGS.o HPGL.o -lxf11 -lx11
```

De object-files moeten hiervoor in de dezelfde directory als de source -file aanwezig zijn. De -c optie geeft aan dat de linker onderdrukt wordt.

Voor het compileren van TADSOL zelf zijn nog meer subroutines noodzakelijk:

- HEADD bevat de tekst die aan het begin van het programma op het scherm getoond wordt.
- PIONXSDS bevat een groot aantal uitgebreidere reken- en teken- en tekstfuncties.
- MS, MR, sms en smr zijn routines met gegevens over de catalogusmechanismen.

De compile-commando's zijn dan:

```
f77 tadsol_1.f g0X.o GMS.o FILES.o STRINGS.o HPGL.o HEADD.o MS.o MR.o sms.o smr.o  
PIONXSDS.o -lxf11 -lx11
```

```
f77 tadsol_2.f g0X.o GMS.o FILES.o STRINGS.o HPGL.o HEADD.o MS.o MR.o PIONXSDS.o -  
lxf11 -lx11
```

Ook hier moeten de object-files in dezelfde directory als de source-file aanwezig zijn.

UNIX maakt verschil tussen hoofdletters en kleine letters. Let hierop bij het benoemen en aanroepen van files.

Voor meer informatie over het Graphical Mini System en een beschrijving van de verschillende subroutines zie [7]. Informatie over het compileren van Fortran-programma's is te vinden in [11].

## Bijlage 3. Tabel mechanismetypen

	Roterend (R)	Slingerend (S)	Translerend (T)
1	*	(T001)	*
2	*	(T002)	*
3	S006	(T003)	*
4	*	(T004)	*
5	*	R003	S005
6	S006	S006	S006
7	R003	T007	T008
8	R003	T008	T008
9	S006	(T009)	*
10	T008	T010	T008
11	R003	(T011)	(T012)
12	X	(T012)	*
13	R006	(T013)	*
14	R006	T014	T010
15	*	X	X
16	*	(T016)	*
17	*	X	X
18	*	(T018)	*
19	X	S005	S019
20	*	X	X
30	X	(T030)	*
35	X	(T035)	*
37	R006	R014	S037
38	X	(T038)	*
39	X	(T039)	*
40	X	(T040)	*

Tabel 1. Indeling catalogustypen.

- X = Mechanismetype bestaat niet  
 \* = Mechanismetype niet eenvoudig van ander type af te leiden  
 S006 = Mechanismetype rechtstreeks af te leiden van S006  
 (T001) = Mechanismetype rechtstreeks af te leiden van T001, maar T001 is niet van een basistype af te leiden

## Bijlage 4. Programmeren montagestand

### 4.1. Gebruikte variabelen

(E) = ENTITY

(S) = STRING

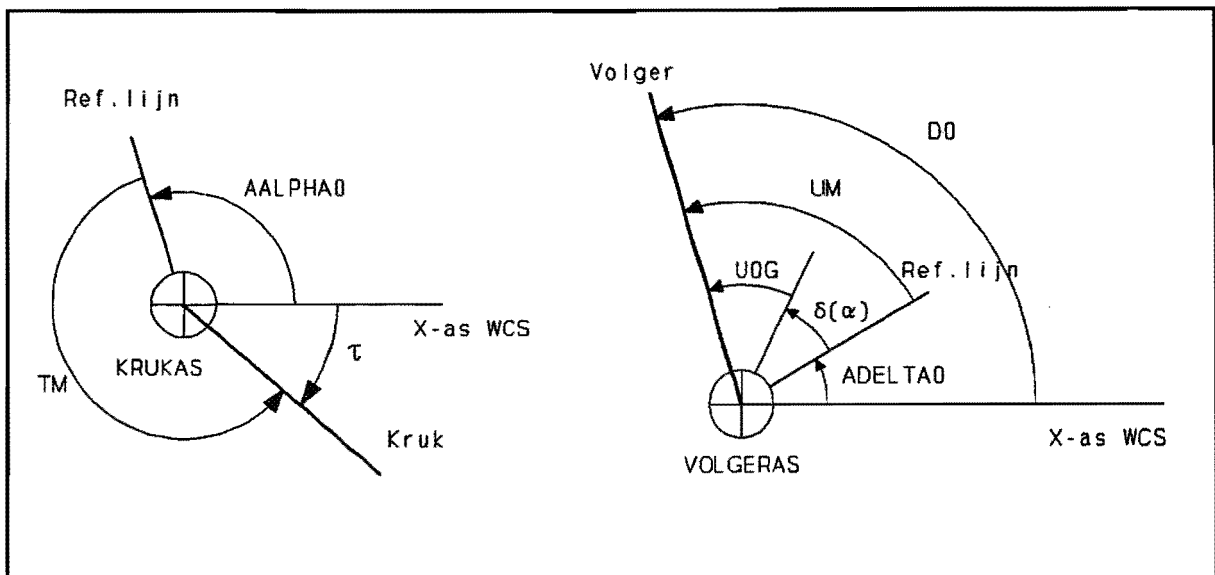
rest = NUMBER

#### Montagestand.grs

KEUS2	bewegingskenmerk (5=R, 6=S, 7=T)
LAYM	layer met mechanismevlak
ALPHA0 (E)	referentielijn krukas
DELTA0 (E)	referentieijk volgeras
AALPHA0	hoek referentielijn kruk in graden
ADELTA0	hoek referentielijn volgeras in graden
DMACH	afstand tussen krukas en volgeras
EMACH	excentriciteit bij T-mechanismen
X	volgnummer gekozen oplossing
M	aantal geselecteerde oplossingen
PAR(Y,X)	array met parameters geselecteerde oplossingen (Y=parameternummer)
TYPE(X)	array met typenummers geselecteerde oplossingen
TAU(X)	array met $\tau$ -waarden geselecteerde oplossingen
LS(X)	array met spiegelgetallen geselecteerde oplossingen
U0(X)	array met $U_0$ -waarden geselecteerde oplossingen
KENM(X) (S)	array met bewegingskenmerk geselecteerde oplossingen
DATNAAM (S)	pad waar selectiefile te vinden is

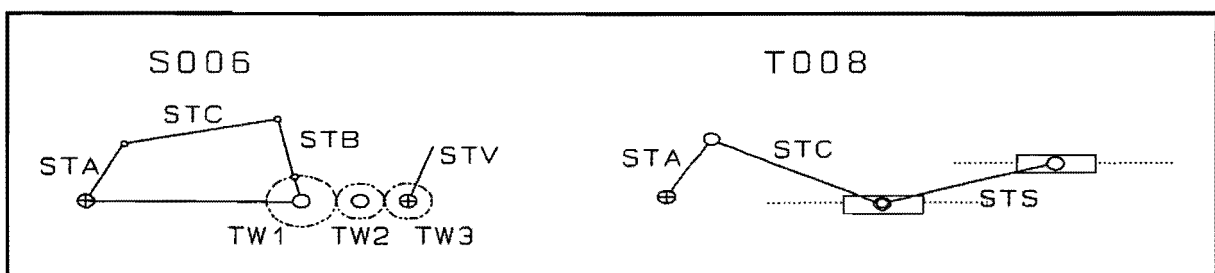
#### Type-afhankelijk deel

TAUG	$\tau$ -waarde gekozen oplossing in graden
U0G	$U_0$ -waarde gekozen oplossing in graden
TM	montagehoek kruk in graden
UM	montagehoek volger in graden
D0	tekenhoek volger in graden
AGAMMA	scheefstandhoek mechanisme in graden
REFD	draairichting kruk (1=rechtsom, -1=linksom)
REFV	draairichting volger (1=rechtsom, -1=linksom)
EXTA	extra tandwiel (1=wel, -1=niet)



Figuur 2. Montagehoeken.

HAZ	heugel andere zijde (1=wel, -1=niet)
X1STA	x-positie startpunt stang A
Y1STA	y-positie startpunt stang A
X2STA	x-positie eindpunt stang A
Y2STA	y-positie eindpunt stang A
LA	lengte stang A
XTW1	x-positie tandwiel 1
YTW1	y-positie tandwiel 1
R1	straal tandwiel 1



Figuur 3. Benaming entiteiten.

ST(N) (E)	array met stangentiteiten (N=volgnummer)
TW(N) (E)	array met tandwielentiteiten (N=volgnummer)
SCH(N) (E)	array met schuif-/heugelentiteiten (N=volgnummer)
VS(N) (E)	array met vastscharnierentiteiten (N=volgnummer)
VT(N) (E)	array met vasttandwielentiteiten (N=volgnummer)

#### 4.2. Opbouw type-afhankelijk deel

Het type-afhankelijk deel van S006 bevat de volgende stappen (zie listing bijlage 14):

1. Bepalen van de referentiewaarden refD, refV, EXTA
2. Bepalen montagehoek volger m.b.v overdrachtsfunctie en  $U_0$
3. Bepalen aantal tandwielen m.b.v de referentiewaarden
4. Bepalen grootte en positie tandwielen
5. Bepalen lengte stangen m.b.v kinematische parameters
6. Bepalen montagehoek kruk (stang a) m.b.v  $\tau$ ,  $\alpha$  en refD
7. Bepalen positie stang b en c door middel van snijcirkels
8. Bepalen positie stang d en volgerstang v
9. Eventueel wissen schets

Het type-afhankelijke deel van T006 ontstaat uit S006 door de volgende aanpassingen (zie listing bijlage 17):

2. De de positie van de volger (in dit geval dus een heugel) wordt hier bepaald door de overdrachtsfunctie,  $U_0$  én de excentriciteit (EMACH)
3. Hier is maar één tandwiel nodig
4. De positie van het tandwiel wordt hier bepaald m.b.v de referentiewaarden en de excentriciteit
8. Volgerstang v is hier niet nodig

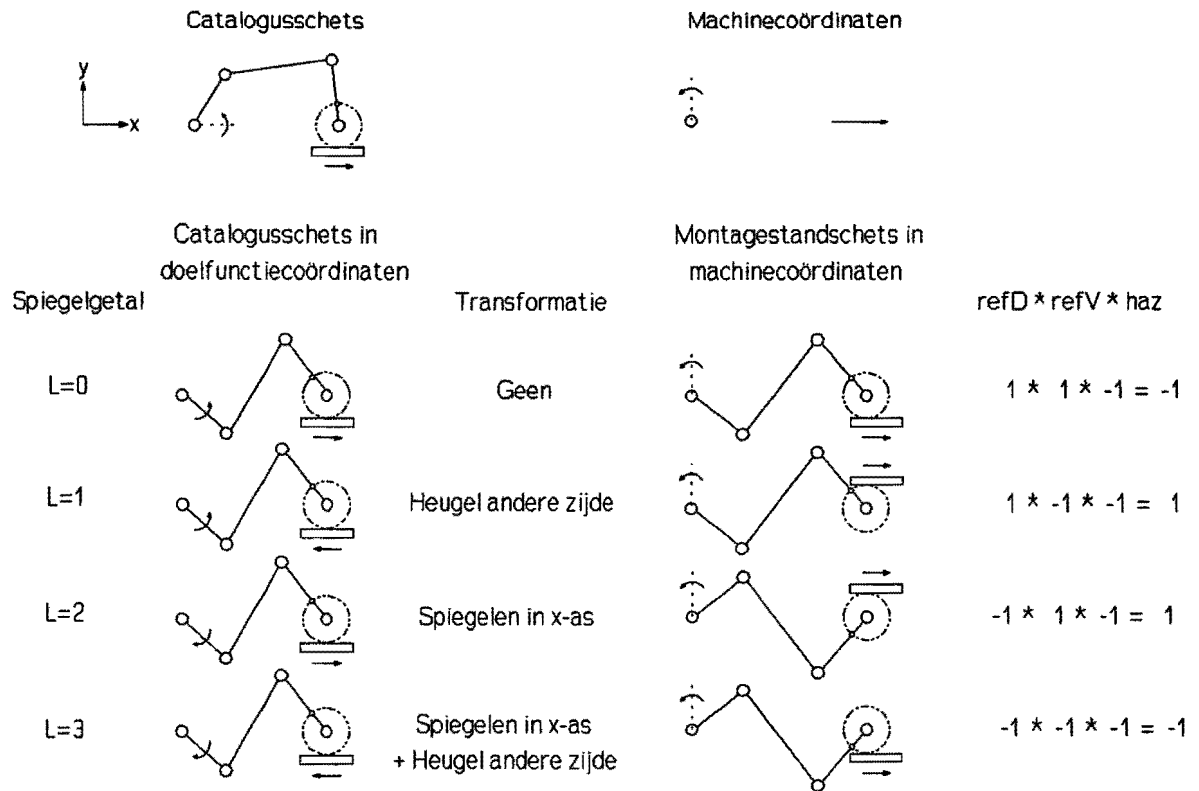
Het type-afhankelijke deel van R003 ontstaat uit S006 door de volgende aanpassingen (zie listing bijlage 16):

2. De overdrachtsfunctie moet worden vervangen door die van R003
5. Stang c wordt hier vervangen door een schuif
7. De positie van de schuif en stang c wordt hier bepaald door middel van één snijcirkel

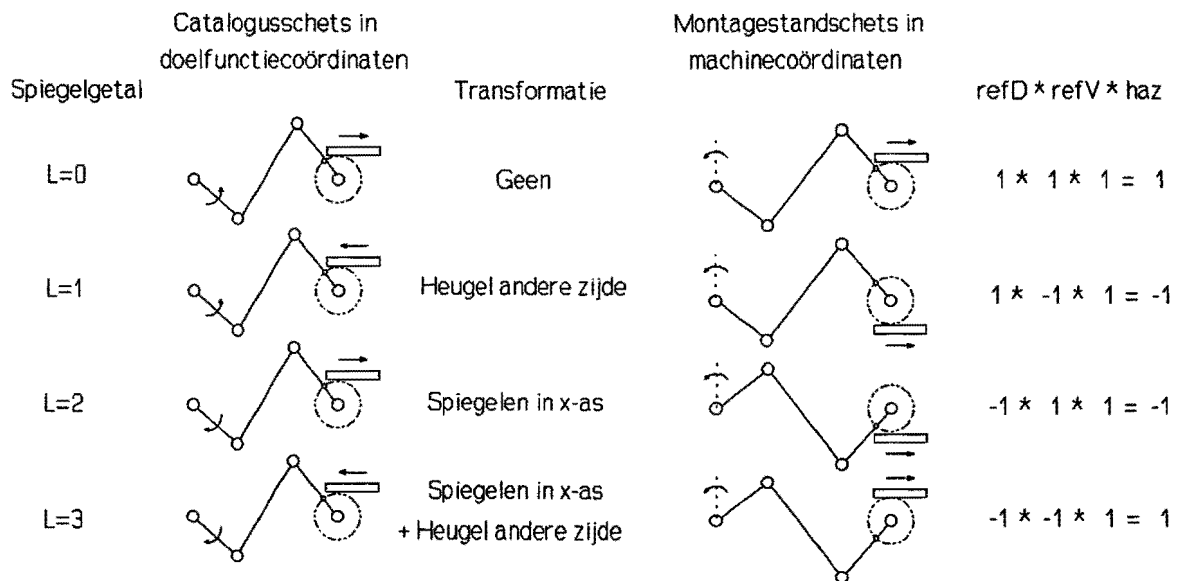
Het type-afhankelijk deel van T008 bevat de volgende stappen (zie listing bijlage 15):

1. Bepalen referentiewaarden refX, refY
2. Bepalen lengte stangen m.b.v kinematische parameters
3. Bepalen montagehoek kruk (stang a) m.b.v  $\tau$  en referentiewaarden
4. Bepalen positie stang c en volger m.b.v excentriciteit
5. Bepalen positie stang s m.b.v. overdrachtsfunctie en  $U_0$
6. Eventueel wissen schets

4.3. Transformaties catalogustype T006



Met corrigeren tandwieloverbrenging



Figuur 4. Transformaties catalogustype T006.

## Bijlage 5. Installatie TADSOL

Om TADSOL te gebruiken moeten alle programmadelen in de hoofddirectory (in mijn geval /users/std/abeelen/) aanwezig zijn :

- **tadsol\_1.exe en tadsol\_2.exe.**

Zoals eerder gezegd zijn de oorspronkelijke TADSOL-programma's enigszins aangepast. Daarbij zijn TADSOL\_C.EXE en TADSOL\_E.EXE samengevoegd tot TADSOL\_1.EXE en is TADSOL\_S.EXE veranderd in TADSOL\_2.EXE.

- **start.exe**

Dit is een batch-file waarmee TADSOL\_1.EXE en TADSOL\_2.EXE worden aangeroepen.

- **SMECAT.DAT en RMECAT.DAT**

Dit zijn data-files met informatie over de catalogusmechanismen.

- **gmsfont.cfg**

Deze file bevat de beschrijving van het font dat gebruikt wordt bij de grafische uitvoer van TADSOL.

Tijdens het gebruik van UGTADSOL worden de aangemaakte data-files (tadsol\_in.dat, tadsol\_out.dat en mechan\_in.dat ook in deze directory weggeschreven. Om te zorgen dat in montagestand.grs de selectiefile mechan\_in.dat weer uitgelezen kan worden moet de pad-aanduiding hier aangepast worden aan deze directory (in mijn geval DATNAAM='/users/std/abeelen/mechan\_in.dat).

Ook hier is het van belang te letten op het verschil tussen hoofdletters en kleine letters bij het benoemen en aanroepen van files.



**Bijlage 6. Voorbeeld TADSOL\_IN.DAT**

SMECAT.DAT

-9DOELFUNCTIE

1	0.00000	.00000	DOELFUNCTIE
2	10.00000	.00000	DOELFUNCTIE
3	50.00000	20.00000	DOELFUNCTIE
4	60.00000	30.00000	DOELFUNCTIE
5	70.00000	40.00000	DOELFUNCTIE
6	80.00000	50.00000	DOELFUNCTIE
7	90.00000	60.00000	DOELFUNCTIE
8	130.00000	80.00000	DOELFUNCTIE
9	140.00000	80.00000	DOELFUNCTIE

KENMERK 2 EENHX GRADEN EENHY GRADEN

EPS .05000 A0 .00000 POL 0

## Bijlage 7. Voorbeeld TADSOL\_OUT.LIS

MECHANISM CATALOGUE INITIATED  
 1 = MECHANISM NUMBER  
 2 = MECHANISM TYPE: R=1, S OR T=2  
 3 = NUMBER OF HARMONICS  
 4 = MODE OR SUBGROUP  
 5 = AMPLITUDE CONFIGURATION  
 6 = SHIFT KRITERION  
 7 = COS-SIN-CONFIGURATION  
 8 = SEQUENCE NUMBER OR ADDRESS

1	234	5	6	7	8	
1	21	1	2100000	1	2001000000000000	1
2	22	1	2110000	0	3022330000000000	2
2	22	2	2101000	0	3022003300000000	3
2	22	3	2100100	0	3022000033000000	4
2	22	4	2100010	0	3022000000330000	5
2	22	5	2100001	0	3022000000003300	6
4	23	1	2111000	0	3022333300000000	7
4	23	2	2110100	0	3022330033000000	8
4	23	3	2110010	0	3022330000330000	9
4	23	4	2110001	0	3022330000003300	10
4	23	5	2101100	0	3022003333000000	11
3	23	6	2102020	2	20-100010003000	12
4	23	6	2101010	0	30220033003300	13
9	23	6	2101020	1	20010001000200	14
13	23	6	2101020	1	3001000-1000200	15
38	23	6	3102020	0	30220033003300	16
4	23	7	2101001	0	30220033000033	17
4	23	8	2100110	0	30220000333300	18
4	23	9	2100101	0	30220000330033	19
4	23	10	2100011	0	30220000003333	20
7	24	5	2110202	2	30101000-200020	21
30	25	2	2122202	2	30103333330033	22
30	25	4	2120222	2	30103300333333	23
12	26	1	2112222	1	30010102020202	24
5	26	1	2112222	6	30-2-1-2-1-2-2-2-2-2	25
14	26	1	2112222	4	301-11002-200-220	26
16	26	1	2111222	1	300-11001-200-220	27
18	26	1	2111222	1	30010-1010-2020-2	28
19	26	1	2112222	6	302-13-13-23-23-2	29
8	26	1	2112222	4	301-11002-200-220	30
11	26	1	2112222	1	20010102020202	31
35	26	1	2112222	1	200-1010-2020-202	32
37	26	1	2112222	6	302-13-13-23-23-2	33
6	26	1	2112222	6	303-13-13-23-23-2	34
41	26	1	2122222	0	30223333333333	35

INPUT LOG -9

1	.00000	.00000DOELFUNCTIE
2	10.00000	.00000DOELFUNCTIE
3	50.00000	20.00000DOELFUNCTIE
4	60.00000	30.00000DOELFUNCTIE
5	70.00000	40.00000DOELFUNCTIE
6	80.00000	50.00000DOELFUNCTIE
7	90.00000	60.00000DOELFUNCTIE
8	130.00000	80.00000DOELFUNCTIE
9	140.00000	80.00000DOELFUNCTIE

KENMERK 2 EENHX GRADEN EENHY GRADEN  
 1FUNCTION VALUES OF CONVERTED FUNCT.

INDEX	ARGUMENT	FUNCTION VALUE
1	.00000	.00000
2	.17453	.00000
3	.87266	.34907
4	1.04720	.52360
5	1.22173	.69813
6	1.39626	.87266
7	1.57080	1.04720
8	2.26893	1.39626
9	2.44346	1.39626

EPS= .05000 ANUL= .71147 IPOL= 0  
 NUMBER OF SIGNIFICANT HARMONICS = 2  
 FOR ARG= .08727 EXTREME VALUE= -.00069  
 FOR ARG= 2.35620 EXTREME VALUE= 1.39695

## 1SOLUTIONS AFTER DIMENSION SYNTHESIS AND EVALUATION

NO	MECH	PAR1	PAR2	PAR3	PAR4	PAR5	PAR6	PAR7	PAR8
1	S002	.6932	.1717	5.9602	.8454	-1.00			
		TAU= .0000		L= 3	U0= .7172	ERR= .0402			
2	S004	.6932	.1717	.0506	5.9602	.8454	2.0287	-1.00	-1.00
		TAU= .0000		L= 3	U0= .7221	ERR= .0254			
3	S004	.6932	.1717	.0110	5.9602	.8454	3.3230	-1.00	-2.00
		TAU= .0000		L= 3	U0= .7172	ERR= .0327			
4	S004	.6932	.1717	.0024	5.9602	.8454	.8701	-1.00	-3.00
		TAU= .0000		L= 3	U0= .7181	ERR= .0426			
5	S004	.6932	.1717	.0039	5.9602	.8454	2.4892	-1.00	-4.00
		TAU= .0000		L= 3	U0= .7176	ERR= .0430			
6	S012	.3326	2.0248						
		TAU= 1.2478		L= 0	U0= .6970	ERR= .0380			

NO	MECH	PAR1	PAR2	PAR3	PAR4	PAR5	PAR6	PAR7	PAR8
7	S012	.3326	2.0248						
		TAU= -1.2478	L=	U0= 3	ERR= .6970	.0380			
8	S005	.2763	2.3897	.2980					
		TAU= -4.7264		L= 1	U0= -.0830	ERR= -.1033			
9	S065	.2750	2.3083	.3425					
		TAU= 5.4490		L= 2	U0= 1.6013	ERR= -.1506			
10	S014	.7703	.5092	1.0911					
		TAU= .4227		L= 1	U0= .0686	ERR= .0549			
11	S014	.8100	.4703	1.2008					
		TAU= -1.9935		L= 2	U0= 1.3351	ERR= -.0753			
12	S018	.5163	.3540	1.3885					
		TAU= -1.8938		L= 1	U0= .6946	ERR= .0459			
13	S018	.5163	.3540	1.3885					
		TAU= 1.8938		L= 2	U0= .6946	ERR= .0459			
14	S019	.2750	.6401	.4376	2.3083				
		TAU= -5.4490		L= 1	U0= .2964	ERR= .0316			
15	S019	.2763	.6522	.5060	2.3897				
		TAU= 4.7264		L= 2	U0= 1.1475	ERR= -.1129			

NO	MECH	PAR1	PAR2	PAR3	PAR4	PAR5	PAR6	PAR7	PAR8
16	S008	.3408	.5092	.6187					
		TAU= .4227		L= 1	U0= .0520	ERR= -.0736			
17	S008	.2973	.4703	.6796					
		TAU= -1.9935		L= 2	U0= 1.3607	ERR= -.0996			
18	S011	.3478	2.0000						
		TAU= 1.2478		L= 0	U0= .6955	ERR= .0372			
19	S035	.3326	.6734						
		TAU= -1.8938		L= 0	U0= .6970	ERR= .0380			
20	S006	.2750	2.3083	.8099	1.4391				
		TAU= -5.4490		L= 1	U0= -.5679	ERR= .0318			
21	S006	.2763	2.3897	.6707	1.3059				
		TAU= 4.7264		L= 2	U0= 1.9592	ERR= -.1022			
22	S041	.4821	.0648	.6988	1.1679				
		TAU= .0000		L= 3	U0= .7180	ERR= .0337			

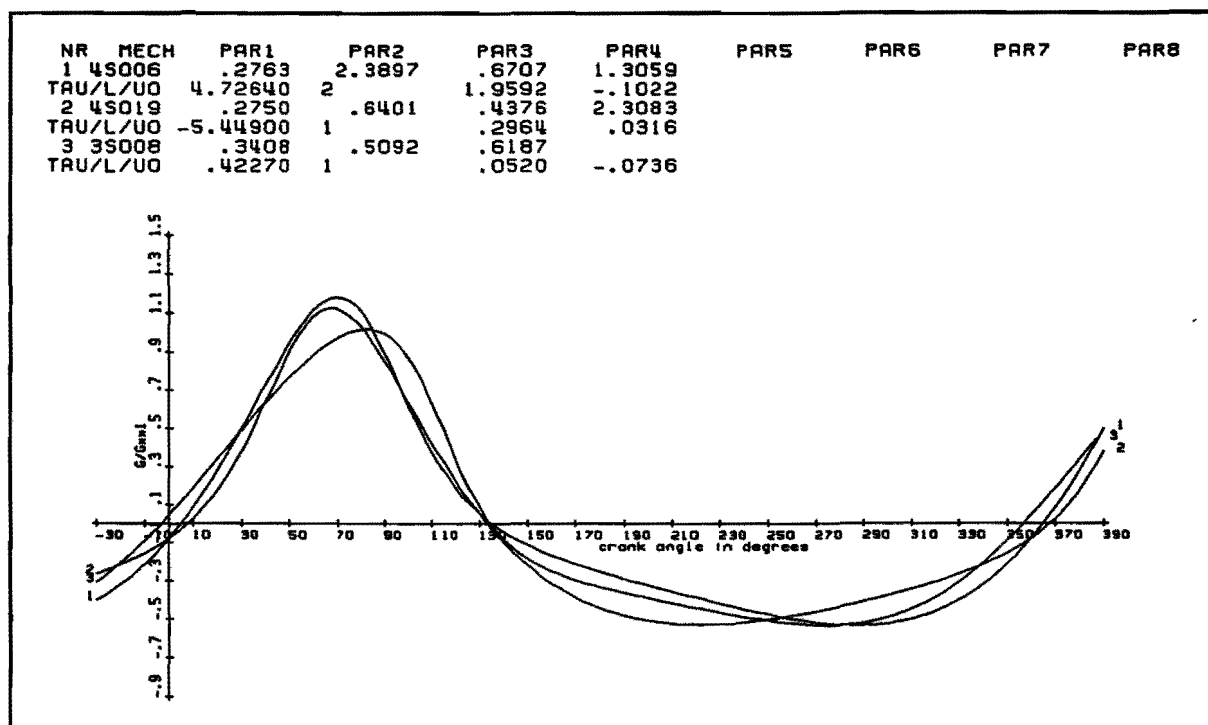
FINISH

**Bijlage 8. Voorbeeld MECHAN\_IN.DAT**

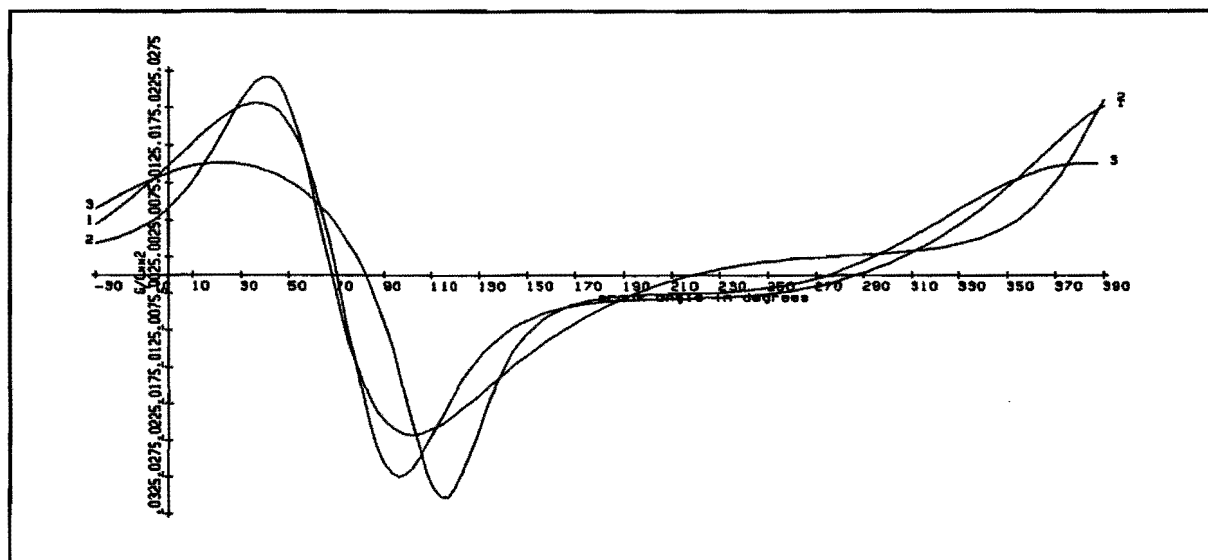
```
4 2 EENHX GRADEN  EENHM GRADEN
1  4S006  .2763    2.3897  .6707  1.3059
      4.72640  2      1.9592  -.1022
2  4S019  .2750    .6401  .4376  2.3083
      -5.44900  1      .2964  .0316
3  3S008  .3408    .5092  .6187
      .42270  1      .0520  -.0736
```

-1

## Bijlage 9. Voorbeeld overdrachtfuncties



Figuur 5. Eerste orde overdrachtfuncties.



Figuur 6. Tweede orde overdrachtfuncties.

## Bijlage 10. Listing UGTADSOL.GRS

```

$$-----
$$
$$ NAAM : UGTADSOL (HOOFDPROGRAMMA)
$$
$$ ZORGT VOOR DE COMMUNICATIE TUSSEN DE
$$ VERSCHILLENDE DEELPROGRAMMA'S DIE
$$ NODIG ZIJN BIJ HET ONTWERPEN
$$ VAN VLAkke FUNCTIEGENERERENDE
$$ STANGENMECHANISMEN
$$
$$ PROGRAMMEUR : O.F.M. NIENS
$$ AANGEPAST : H.J.M. VAN ABEELLEN
$$
$$ TU EINDHOVEN, VAKGROEP WPA
$$
$$ DATUM : 07-06-93
$$
$$ DECLARATIES=====
$$

ENTITY/MSYS(20),CSYS(10),ALPHA0,DELTA0,CLO,G
RP(100)

NUMBER/KEUS2,EMACH,DMACH,PAR1(100),$
PAR2(100),PAR3(100),PAR4(100),PAR5(100),$
PAR6(100),PAR7(100),PAR8(100),TAU(100),$
LS(100),UO(100),OPL,CODE(100),DK,LAYM

$$
MESSG/'DIT PROGRAMMA IS BEDOELD VOOR
DE VERWERKING',$
'VAN DE OPLOSSINGGEGEVENS VAN TADSOL.'
MESSG/'HET ZET DEZE GEGEVENS OM IN EEN
MONTAGESTANDSCHETS',$
'IN EEN MACHINETEKENING.'
MESSG/'VOOR HET AANMAKEN OF WIJZIGEN
VAN DE OPLOSSINGS-',$
'FILE MOET TADSOL GESTART WORDEN OP EEN
TWEDE SCHERM.'
MESSG/'KEER NA AFLOOP TERUG NAAR DIT
SCHERM.'

CALL/'V-ISO-3D'

CALL/'INTERFACE',KEUS2,EMACH,DMACH,$
MSYS,ALPHA0,DELTA0,CLO,DK,LAYM

CALL/'MONTAGESTAND',KEUS2,LAYM,ALPHA0,DE
LTA0,DMACH,EMACH
$$
HALT:
MESSG/'EINDE ONTWERPPROCES.',$
'VERGEET NIET TADSOL AF TE SLUITEN !'
HALT

```



## Bijlage 11. Listing V-ISO-3D.GRS

```

$$$$=====
$$
$$ NAAM : V-ISO-3D (DEELPROGRAMMA)
$$
$$ ZORGT VOOR HET CREEREN VAN EEN
$$ LAYOUT MET EEN V-ISO-3D VIEW
$$ VAN EEN BESTAANDE MACHINETEKENING
$$
$$ PROGRAMMEUR : H.F. BONEKAMP
$$
$$ TU EINDHOVEN, VAKGROEP WPA
$$
$$ DATUM : 01-04-93
$$
$$ DECLARATIES=====
$$
PROC/
  ENTITY/PT1,PT2,PT3,CSYS1,CSYS2,CSYS3,CSYS4
  STRING/S1(20),S2(20)
  NUMBER/AP,MAT1(12),MAT2(12),MAT3(12)
$$
$$ INITIALISATIES=====
$$
  S1 = &PNAME

$$
  L10:
  AP = &ACTPRT
  IF/AP = 1,JUMP/L20:
  CHOOSE/'IS DIT DE GEWENSTE
MACHINETEKENING ?',$
  'JA','NEE',DEFLT,1,RSP
  JUMP/L10:,L999:,,,L30:,,RSP
  FTERM/PART
$$
  L20:
  TEXT/'GEEF NAAM GEWENSTE
MACHINETEKENING',S1,RSP
  JUMP/L20:,L999:,,,RSP
  FETCH/PART,S1
  JUMP/L40:
$$
  L30:
  S1 = &PNAME
$$
  L40:

$ LAYR/'V-TUEAM',IFERR,L90:
  &WORKVW = 'V-ISO-3D'
  LAYER/WORK,100
  CSYS4 = &WCS

LAYC/'MACHPART','V-ISO-3D',WORK,AUTO,IFERR,L
996:
  RPAINT
  JUMP/L999:
$$
  L90:
  MESSG/'GEEN 3-D BODY PART! (EC)'
  &WCSDRW = &YES
  &ENTCLR = &WHITE
  &SYSCLR = &YELLOW
  &WBORD = &YES

```

```

PT1=POINT/0,0,0
PT2=POINT/1,0
PT3=POINT/0,1
CSYS1=CSYS/PT1,PT2,PT3
MAT1=MATRIX/XYROT,16
CSYS2=TRANSF/MAT1,CSYS1
N=DVIEW/CSYS2
VIEWN/'HELP1'
MAT2=MATRIX/ZXROT,31
CSYS3=TRANSF/MAT2,CSYS2
L=DVIEW/CSYS3
VIEWN/'HELP2'
MAT3=MATRIX/YZROT,61
CSYS4=TRANSF/MAT3,CSYS3
K=DVIEW/CSYS4
VIEWN/'V-ISO-3D'
DELETE/PT1,PT2,PT3
VIEWD/'HELP1'
VIEWD/'HELP2'

LAYC/'MACHPART','V-ISO-3D',WORK,AUTO,IFERR,L
10:
  JUMP/L999:
$$
  L996:
  LAYR/'MACHPART',AUTO
$$
  L999:
  &WCS = CSYS4
$$
RETURN

```

## Bijlage 12. Listing INTERFACE.GRS

```

$$=====
$$
$$ NAAM : INTERFACE AFSPRAKEN
$$ (DEELPROGRAMMA)
$$
$$ ZORGT VOOR DE DEFINITIE VAN
$$ HET MECHANISMEVLAK MET DE
$$ BUBEHORENDE TEKENAFSPRAKEN
$$ IN EEN BESTAANDE MACHINETEKENING
$$
$$ PROGRAMMEUR : O.F.M. NIENS
$$ AANGEPAST : H.J.M. VAN ABEELLEN
$$
$$ DATUM : 16-05-93
$$
$$ TU EINDHOVEN, VAKGROEP WPA
$$
$$ DECLARATIES=====
$$
PROC/KEUS2,EMACH,DMACH,MSYS,ALPHA0,DELTA
0,CLO,DK,LAYM

ENTITY/KRUKAS,VOLGAS,VOLGER,CLK,CLV,CLO,K
SYS(3),$

MSYS(20),K(2),CSYS1,CSYS2,CSYS3,MVLAK,GRP(100),
ALPHA0

ENTITY/PUNT1,PUNT2(3),PUNT3,PUNT4,PUNT5,PUN
T6,CK(6),$

SPIEK(18),SPIEA(9),SPIE(2),SPIEAS(6),VBP(100),DELT
A0,$

DELTA0,ALPHA0,DIM(100),DIMP,TSYS(10),VVLAK,$
XASL(2),X0,XASLL,K0,V0,KV(2),DELTVO,CLS(6)
NUMBER/PT1(3),PT2(3),PT3(3),PT4(3),PT5(3),$

PT6(3),LAYM,LAY(100),ANG(4),MAT1(12),MAT2(12),$

MDIST(7),XAS1(3),XAS2(3),X1(3),KDIST(7),VDIST(7),$

MAT3(12),MAT4(12),MAT5(12),PT7(3),MAT6(12),DK,DV
,S,$

CKP1(3),MAT7(12),MAT8(12),MAT9(12),HOEK(3),MAT1
0(12)

NUMBER/MAT11(12),MAT12(12),MAT13(12),DIM1(3),D
IM2(3),$
DIM3(3),DIM4(3),PT8(3),PT9(3),PT10(3),MAT14(12),$

MAT15(12),PT20(3),MAT16(12),MAT17(12),MT(12),MAT1
8(12),$
MAT19(12),MAT20(12),PDIST(7),PT11(3),MAT21(12)
STRING/STR(10,20),LAYSTR(20,3)
$$
$$
INITIALISATIES=====
$$
LAYM=150
&LDEV=1
DK=25

DV=25
$$
L15:
MESSG/'VOOR HET MAKEN VAN EEN
MONTAGESTANDSCHETS',$
'IS HET NODIG INTERFACEAFSPRAKEN TUSSEN
MACHINE >>'
MESSG/'EN MECHANISME TE MAKEN.'
MESSG/'DEZE WORDEN VASTGELEGD OP HET
ZOGENAAMDE',$
'MECHANISMELAYER.'
$$
L20:
PARAM/'WELKE LAYER WORDT HET
MECHANISMELAYER ?',$
'MECHANISMELAYER.',INT,LAYM,RSP
JUMP/L20,HALT:,,,RSP
LAYER/WORK,LAYM
$$
L25:
MASK/5
MESSG/'KIES VOOR DE BEPALING V/D
HARTLIJN V/D',$
'DRIJFAS >>'
MESSG/'TWEË PUNTEN AAN WEERSZIJDE V/D
AS',$
'GEBRUIK ARC CENTER ALS METHODE.'
L25A:
RPAINT
GPOS/'PUNT 1 OP DRIJFAS.',PT1(1),PT1(2),$
PT1(3),RSP
JUMP/L25A,HALT:,,,RSP
L25B:
GPOS/'PUNT 2 OP DRIJFAS.',PT2(1),PT2(2),$
PT2(3),RSP
JUMP/L25A,HALT:,,,RSP
IFTHEN/PT1(1)=-PT2(1) AND PT1(2)=-PT2(2) $
AND PT1(3)=-PT2(3)
MESSG/'PUNTEN ZIJN GELIJK EN BEPALEN
DUS',$
'GEEN LIJN.'
JUMP/L25A:
ENDIF
CLK=LINE/PT1(1),PT1(2),PT1(3),PT2(1),$
PT2(2),PT2(3)
&COLOR(CLK)=&SYSCLR
&FONT(CLK)=&CLINE
RPAINT
CHOOSE/'LIGT HARTLIJN DRIJFAS GOED ?',$
'JA','NEE',DEFLT,1,RSP
JUMP/HALT:,,,L25C:,,,RSP
DELETE/CLK
RPAINT
JUMP/L25A:
L25C:
RPAINT
$$
L30:
CHOOSE/'BEWEGINGSKENMERK VAN DE
VOLGER ?',$
'ROTTEREND (R)','SLINGEREND (S)',$
'TRANSLEREND (T)',KEUS2
JUMP/L30,HALT:,,,L130:,,,KEUS2

```

```

$$
L95:
  MASK/5
  MESSG/'KIES VOOR DE BEPALING V/D
  HARTLIJN V/D', $
  'VOLGAS TWEE PUNTEN AAN WEERSZIJDE V/D
  AS.'
  MESSG/'GEBRUIK ARC CENTER ALS METHODE.'
  L95A:
  RPAINT
  GPOS/'PUNT 1 OP VOLGER-AS',PT3(1),PT3(2),$
  PT3(3),RSP
  JUMP/L95A:,HALT:,,,RSP
  PUNT6=POINT/PT3(1),PT3(2),PT3(3)
  L95B:
  GPOS/'PUNT 2 OP VOLGER-AS',PT4(1),PT4(2),$
  PT4(3),RSP
  JUMP/L95A:,HALT:,,,RSP
  IFTHEN/PT3(1) == PT4(1) AND PT3(2) == PT4(2) $
  AND PT3(3) == PT4(3)
  MESSG/'PUNTEN ZIJN GELIJK EN BEPALEN ', $
  'DUS GEEN LIJN.'
  JUMP/L95A:
  ENDIF
  CLV = LINE/PT3(1),PT3(2),PT3(3),PT4(1),PT4(2),$
  PT4(3)
  &COLOR(CLV) = &SYSCLR
  &FONT(CLV) = &CLINE
  RPAINT
  CHOOSE/'LIGT HARTLIJN VOLGER-AS GOED '?,$
  'JA','NEE',DEFLT,1,RSP
  JUMP/,HALT:,,,L95C:,,RSP
  DELETE/CLV
  RPAINT
  JUMP/L95A:
  L95C:
  RPAINT
$$
L100:
  KSYS(1) = &WCS
  PUNT3 = POINT/(PT1(1) + PT2(1))/2,$
  (PT1(2) + PT2(2))/2,(PT1(3) + PT2(3))/2
  PUNT4 = POINT/PT1(1),PT1(2),PT1(3)
  PUNT5 = POINT/(PT1(1) + PT2(1))/2,$
  ((PT1(2) + PT2(2))/2 + 1),((PT1(3) + PT2(3))/2 + 1)
  CSYS3 = CSYS/PUNT3,PUNT4,PUNT5
  &WCS = CSYS3
  DELETE/PUNT4,PUNT5
  IF/KEUS2 == 7,JUMP/L140:
  ANG(1) = ANGLF(CLK)
  ANG(2) = ANGLF(CLV)
  IF/ANG(1) < > ANG(2) AND ANG(1) < > ANG(2) + 180
  AND$
  ANG(1) < > ANG(2) - 180,JUMP/L997:
  MAT15 = MATRIX/YZROT,90
  TSYS(3) = TRANSF/MAT15,CSYS3
  &WCS = TSYS(3)
  ANG(1) = ANGLF(CLK)
  ANG(2) = ANGLF(CLV)
  IF/ANG(1) < > ANG(2) AND ANG(1) < > ANG(2) + 180
  AND$
  ANG(1) < > ANG(2) - 180,JUMP/L997:
  JUMP/L200:
$$
L130:
  MASK/2
  MESSG/'KIES VOOR BEPALING V/D
  VOLGERBAAN', $
  'TWEE PUNTEN OP DIE BAAN.'
  MESSG/'GEBRUIK CONTROL POINT OF END
  POINT.'
  L130A:
  RPAINT
  GPOS/'PUNT 1',PT5(1),PT5(2),PT5(3),RSP
  JUMP/L130A:,HALT:,,,RSP
  L130B:
  GPOS/'PUNT 2',PT6(1),PT6(2),PT6(3),RSP
  JUMP/L130A:,HALT:,,,RSP
  IFTHEN/PT5(1) == PT6(1) AND PT5(2) == PT6(2)
  AND$
  PT5(3) == PT6(3)
  MESSG/'PUNTEN ZIJN GELIJK EN BEPALEN', $
  'DUS GEEN LIJN.'
  JUMP/L130A:
  ENDIF
  CLO = LINE/PT5(1),PT5(2),PT5(3),PT6(1),PT6(2),PT6(3)
  &COLOR(CLO) = &SYSCLR
  RPAINT
  CHOOSE/'IS DE TRANSLATIE-RICHTING JUIST
  '?,$
  'JA','NEE',DEFLT,1,RSP
  JUMP/L130C:,HALT:,,,L130D:,L130C:,,RSP
  L130C:
  DELETE/CLO
  RPAINT
  JUMP/L130A:
  L130D:
  JUMP/L100:
$$
L140:
  ANG(1) = ANGLF(CLK)
  ANG(2) = ANGLF(CLO)
  IF/ANG(1) == ANG(2) + 90 OR
  ANG(1) == ANG(2) - 90,JUMP/L199:
  MAT15 = MATRIX/YZROT,90
  TSYS(3) = TRANSF/MAT15,CSYS3
  &WCS = TSYS(3)
  ANG(1) = ANGLF(CLK)
  ANG(2) = ANGLF(CLO)
  IF/ANG(1) == ANG(2) + 90 OR
  ANG(1) == ANG(2) - 90,JUMP/L200:
  JUMP/L996:
$$
L199:
  MAT15 = MATRIX/YZROT,90
  TSYS(3) = TRANSF/MAT15,CSYS3
$$
L200:
  &WCS = TSYS(3)
  MT = MATRIX/ZXROT,90
  CSYS1 = TRANSF/MT,TSYS(3)
  &WCS = CSYS1
$$
L201:
  IF/KEUS2 == 7,JUMP/L201AA:
  XAS1(1) = 0
  XAS1(2) = 0
  XAS1(3) = 0
  L201A:
  OBTAIN/PUNT6,XAS2(1),XAS2(2),XAS2(3)
  XAS2(3) = 0

```

```

JUMP/L201AB:
L201AA:
OBTAIN/CLO,XAS1(1),XAS1(2),XAS1(3),XAS2(1),$
XAS2(2),XAS2(3)
L201AB:
XASL(1)=LINE/XAS1(1),XAS1(2),XAS1(3),XAS2(1),$
XAS2(2),XAS2(3)
RPAINT
$$
L202:

MAT3=MATRIX/TRANSL,-XAS1(1),-XAS1(2),-XAS1(3)
XASLL=TRANSF/MAT3,XASL(1)
X1=&ORIGIN(CSYS1)
X0=POINT/X1(1),X1(2),X1(3)
CSYS2=CSYS/XASLL,CLK,ORIGIN,X0
&WCS=CSYS2
DELETE/XASLL,XASL(1),X0
MAT4=MATRIX/YZROT,90
CSYS3=TRANSF/MAT4,CSYS2
&WCS=CSYS3
RPAINT
IF/KEUS2==7,JUMP/L230A:
$$
L210:
MDIST=RELDST/MIN,CLK,CLV
IF/MDIST(4)==0 AND MDIST(5)==0,JUMP/L211:
JUMP/L230:
$$
L211:
MESSG/'DRIJFAS EN VOLGER-AS LIGGEN IN
ELKAARS', $
'VERLENGDE !'
D=0
JUMP/L230B:
$$
L230:
MDIST=RELDST/MIN,CLK,CLV
IF/MDIST(4)>MDIST(1),JUMP/L230B:
MAT4=MATRIX/XYROT,180
MSYS(2)=TRANSF/MAT4,CSYS3
JUMP/L230C:
$$
L230A:
MDIST=RELDST/MIN,CLK,CLO
IF/MDIST(4)>MDIST(1),JUMP/L230B:
MAT4=MATRIX/ZXROT,180
MSYS(2)=TRANSF/MAT4,CSYS3
JUMP/L230C:
$$
L230B:
MAT4=MATRIX/XYROT,0
MSYS(2)=TRANSF/MAT4,CSYS3
$$
L230BB:
IFTHEN/KEUS2<>7
MAT5=MATRIX/XYROT,0
JUMP/L230BC:
ENDIF
&WCS=MSYS(2)

OBTAIN/CLO,PTS(1),PTS(2),PTS(3),PT6(1),PT6(2),PT6(3
)
MAT5=MATRIX/TRANSL,0,0,PTS(3)
L230BC:
MSYS(3)=TRANSF/MAT5,MSYS(2)
$$
L230C:
&WCS=MSYS(3)
MVLAK=PLANE/XYPLAN,0
&COLOR(MVLAK)=&SYSCLR
RPAINT
$$
L240:
IF/KEUS2==7,JUMP/L240A:
MESSG/'HET DRIEHOEKJE GEEFT HET VLAK
AAN WAARIN', $
'DE KRUK ROTTEERT (DEFAULT: MIDDEN
KRUKAS). '
JUMP/L240B:
L240A:
MESSG/'HET DRIEHOEKJE GEEFT HET VLAK
AAN WAARIN', $
'DE KRUK ROTTEERT (DEFAULT: VOLGERBAAN). '
L240B:
RPAINT
CHOOSE/'WIJZIGEN Z-WAARDE V/H VLAK ?', $
'JA','NEE',DEFLT,2,RSP
JUMP/,HALT:,,,L261:,RSP
PARAM/'GEEF Z-VERPLAATSING.', $
'Z-WAARDE',MECHZ,RSP
DELETE/MVLAK
RPAINT
MAT6=MATRIX/TRANSL,0,0,MECHZ
MSYS(4)=TRANSF/MAT6,MSYS(3)
MSYS(3)=MSYS(4)
RPAINT
JUMP/L230C:
$$
L261:
MSYS(4)=&WCS
CHOOSE/'GEEF REF. LIJN DRIJFAS
(KRUKHOEK=0).', $
'POSITIEVE Y-AS','POSITIEVE X-AS','NEGATIEVE
Y-AS', $
'NEGATIEVE X-AS','ANDERE LIJN',DEFLT,1,RSP
JUMP/L261:,HALT:,,,L261A:,L261B:,L261C:,L261D:,RSP
ALPHA0=LINE/0,0,0,0,4*DK,0
HOEK(1)=90
JUMP/L262:
$$
L261A:
ALPHA0=LINE/0,0,0,4*DK,0,0
HOEK(1)=0
JUMP/L262:
$$
L261B:
ALPHA0=LINE/0,0,0,0,-4*DK,0
HOEK(1)=-90
JUMP/L262:
$$
L261C:
ALPHA0=LINE/0,0,0,-4*DK,0,0
HOEK(1)=180
JUMP/L262:
$$
L261D:
PARAM/'HOEK TUSSEN REF. LIJN EN POS.
X-AS.', $
'HOEK',HOEK(1),RSP

```

```

JUMP/L261D:;HALT:;,RSP
MAT10=MATRIX/XYROT,HOEK(1)
MSYS(5)=TRANSF/MAT10,MSYS(4)
&WCS=MSYS(5)
ALPHA0=LINE/0,0,0,4*DK,0,0
&WCS=MSYS(4)
DELETE/MSYS(5)
$$
L262:
&FONT(ALPHA0)= &DASHED
&COLOR(ALPHA0)= &WHITE
RPAINT
CHOOSE/'IS DE LIGGING V/D REF. LIJN NU
GOED ?', $
'JA','NEE',DEFLT,1,RSP
JUMP/,HALT:;,L263:;,RSP
JUMP/L261:
$$
L263:
MESSG/'HET VLAK WAARIN EN DE REF. LIJN
T.O.V.:', $
'WAARVAN DE KRUK ROTEERT ZIJN NU
BEPAALD.'
IF/KEUS2 = = 7,JUMP/L2700:
$$
L270:
PT1 = &SPOINT(CLV)
PT2 = &EPOINT(CLV)
JUMP/L2701:
L2700:
PT1 = &SPOINT(CLO)
PT2 = &EPOINT(CLO)
L2701:
MAT16=MATRIX/TRANSL,(PT1(1)+PT2(1))/2,(PT1(2)
+PT2(2))/2,0
MSYS(3)=TRANSF/MAT16,MSYS(4)
L2702:
&WCS=MSYS(3)
VVLAK=PLANE/XYPLAN,0
&COLOR(VVLAK)= &SYSCLR
RPAINT
IF/KEUS2 < > 7,JUMP/L270A:
OBTAIN/CLO,PT5(1),PT5(2),PT5(3),PT6(1),PT6(2),PT6(3
)
DELETE/CLO
CLO=LINE/PT5(1),PT5(2),0,PT6(1),PT6(2),0
&COLOR(CLO)= &SYSCLR
RPAINT
JUMP/L270AA:
$$
L270A:
MESSG/'HET DRIEHOEKJE GEEFT HET VLAK
AAN WAARIN', $
'DE VOLGER ROTEERT (DEFAULT: Z-WAARDE
KRUKAS).'
JUMP/L270B:
L270AA:
MESSG/'HET DRIEHOEKJE GEEFT HET VLAK
AAN WAARIN', $
'DE VOLGER TRANSLEERT (DEFAULT:
Z-WAARDE KRUKAS).'
L270B:
RPAINT
CHOOSE/'WIJZIGEN Z-WAARDE V/H VLAK ?', $
'JA','NEE',DEFLT,2,RSP
JUMP/,HALT:;,L270C:;,RSP
PARAM/'GEEF Z-VERPLAATSING.', $
'Z-WAARDE',MECHZ,RSP
DELETE/VVLAK
RPAINT
MAT6=MATRIX/TRANSL,0,0,MECHZ
KSYS(2)=TRANSF/MAT6,MSYS(3)
DELETE/MSYS(3)
MSYS(3)=KSYS(2)
RPAINT
JUMP/L2702:
$$
L270C:
IF/KEUS2 = = 7,JUMP/L280:
$$
L271:
MSYS(5)= &WCS
PT1 = &ORIGIN(MSYS(4))
MAT7=MATRIX/TRANSL,-PT1(1),-PT1(2),-PT1(3)
DELTA0=TRANSF/MAT7,ALPHA0
CHOOSE/'T.O.V. WELKE LIJN IS DE VOLGHOEK 0
GRD ?', $
'DE WITTE LIJN','POSITIEVE Y-AS','POSITIEVE
X-AS', $
'NEGATIEVE Y-AS','NEGATIEVE X-AS', $
'ANDERE LIJN',DEFLT,1,RSP
JUMP/L271:;HALT:;,L271A:;,L271B:;,L271C:;,L271D:;, $
L271E:;,RSP
RPAINT
HOEK(3)=HOEK(1)
MAT11=MATRIX/XYROT,0
JUMP/L272:
$$
L271A:
DELETE/DELTA0
DELTA0=LINE/0,0,0,0,4*DV,0
HOEK(3)=90
JUMP/L272:
$$
L271B:
DELETE/DELTA0
DELTA0=LINE/0,0,0,4*DV,0,0
HOEK(3)=0
JUMP/L272:
$$
L271C:
DELETE/DELTA0
DELTA0=LINE/0,0,0,0,-4*DV,0
HOEK(3)=-90
JUMP/L272:
$$
L271D:
DELETE/DELTA0
DELTA0=LINE/0,0,0,-4*DV,0,0
HOEK(3)=180
JUMP/L272:
$$
L271E:
DELETE/DELTA0
PARAM/'HOEK TUSSEN REF. LIJN EN POS.
X-AS.', $
'HOEK',HOEK(3),RSP
JUMP/L271E:;HALT:;,RSP
MAT12=MATRIX/XYROT,HOEK(3)
MSYS(6)=TRANSF/MAT12,MSYS(5)

```

```

&WCS=MSYS(6)
DELTA0=LINE/0,0,0,4*DV,0,0
&WCS=MSYS(5)
DELETE/MSYS(6)
$$
L272:
&FONT(DELTA0)= &DASHED
&COLOR(DELTA0)= &WHITE
RPAINT
CHOOSE/'IS DE LIGGING V/D REF. LIJN NU
GOED?', $
'JA','NEE',DEFLT,1,RSP
JUMP/,HALT:,,,L273:,,,RSP
DELETE/DELTA0
JUMP/L271:
$$
L273:
&WCS=MSYS(4)
RPAINT
DELETE/VVLAK
RPAINT
JUMP/L300:
$$
L280:
&WCS=MSYS(4)
RPAINT
&ENTCLR= &WHITE
MDIST=RELDST/MIN,CLK,CLO
E=MDIST(5)
$$
L280A:
RPAINT
MESSG/'DE NU VOLGENDE WAARDE IS DE
HUIDIGE', $
'EXCENTRICITEIT V/D VOLGERBAAN.'
MESSG/'DEZE EXCENTRICITEIT IS DE AFSTAND
', $
'VANAF DE HARTLIJN V/D DRIJFAS > >'
MESSG/'TOT DE VOLGERBAAN IN DE
Y-RICHTING.', $
'DEZE KUNT U EVENTUEEL WIJZIGEN.'
L280B:
PARAM/'GEEF EXCENTRICITEIT VAN DE
VOLGERBAAN.', $
'EXCENTRICITEIT',E,RSP
JUMP/L280A:,,HALT:,,,RSP
&ENTCLR= &SYSCLR

OBTAIN/CLO,PTS(1),PTS(2),PTS(3),PT6(1),PT6(2),PT6(3)
)
DELETE/CLO
CLO=LINE/PTS(1),E,PT6(1),E
RPAINT
CHOOSE/'WIJZIGEN EXCENTRICITEIT?', $
'JA','NEE',DEFLT,2,RSP
JUMP/,HALT:,,,L280B:,,,RSP

MAT18=MATRIX/TRANSL,MDIST(4),MDIST(5),MDIS
T(6)
MSYS(5)=TRANSF/MAT18,MSYS(4)
JUMP/L300:
$$
L300:
&DPREL= &NO
&WCS=MSYS(4)
PT9= &ORIGIN(MSYS(4))

PT10= &ORIGIN(MSYS(5))
EMACH=E

DMACH=SQRT((PT10(1))*(PT10(1))+(PT10(2))*(PT10
(2)))
DELETE/MVLAK,VVLAK
O=DVIEW/MSYS(4)
VIEWN/'MECHVLAK'

LAYC/'MECHANISME',TOP,'MECHVLAK','V-ISO-3D', $
WORK,AUTO

VIEWWC/'MECHVLAK',VSBL,LAYM,LAYM+1,INVSBL
,REST
&WORKVW='MECHVLAK'
RPAINT
CALL/'VASTSCHAR',11,0,0
&CNMODE= &VIEW
&ENTCLR= &WHITE
&CSIZE=DK/2
VIEWE/AUTO
STR(5)='DRIJFAS'
STR(6)='(ROTREND)'
NOTE/-6*DK,DK,STR(5),STR(6)
$$
L300C:
VIEWE/AUTO

OBTAIN/ALPHA0,PT1(1),PT1(2),PT1(3),PT2(1),PT2(2),P
T2(3)
DIM(1)=POINT/PT1(1),PT1(2),PT1(3)
DIM(2)=POINT/(6/7)*PT2(1),(6/7)*PT2(2),PT2(3)
MAT13=MATRIX/XYROT,45
ALPHA0=TRANSF/MAT13,ALPHA0

OBTAIN/ALPHA0,PT3(1),PT3(2),PT3(3),PT4(1),PT4(2),P
T4(3)
&ASIZE=DK/2
&CSIZE=DK/2

DIM(3)=CIRCLE/CENTER,DIM(1),DIM(2),START,HO
EK(1),END, $
HOEK(1)+45
&FONT(DIM(3))= &DASHED
IF/PT4(1)>0,JUMP/L301:
DIM(4)=DIMBP/DIM(3),NOTE,PT4(1)-2*DK, $
PT4(2)+DK/3,'ALPHA',ARROW,(6/7)*PT4(1),(6/7)*PT4(
2), $
HOEK(1)+135
JUMP/L302:
L301:
DIM(4)=DIMBP/DIM(3),NOTE,PT4(1)+DK/2, $
PT4(2)+DK/3,'ALPHA',ARROW,(6/7)*PT4(1),(6/7)*PT4(
2), $
HOEK(1)+135
L302:
DELETE/ALPHA0
DIM(5)=POINT/PT2(1),PT2(2),PT2(3)

DIM(6)=CIRCLE/CENTER,DIM(1),DIM(5),START,HO
EK(1)+10, $
END,HOEK(1)-10
DIM1= &SPOINT(DIM(6))

```

```

DIM2 = &EPOINT(DIM(6))
DELETE/DIM(5..6),K(1)
STR(1) = '+'
STR(2) = '-'
NOTE/DIM1(1),DIM1(2),STR(1)
NOTE/DIM2(1),DIM2(2),STR(2)
DELETE/DIM(1),DIM(2)
RPAINT
VIEWE/AUTO
IF/KEUS2 = = 7,JUMP/L320:
$$
L310:
OBTAIN/DELTA0,PT10(1),PT10(2),PT10(3),PT11(1),$
PT11(2),PT11(3)
CALL/'VASTSCHAR',12,PT10(1),0
&ENTCLR = &WHITE
&CSIZE = DK/2
STR(7) = 'VOLGER-AS'
STR(8) = '(ROTTEREND)'
STR(9) = '(SLINGEREND)'
IF/KEUS2 = = 6,JUMP/L310D:
NOTE/PT10(1)+6*DVK,DK,STR(7),STR(8)
JUMP/L310G:
$$
L310D:
NOTE/PT10(1)+8*DVK,DK,STR(7),STR(9)
JUMP/L310G:
$$
L310G:
&WCS = MSYS(5)
&ASIZE = DK/2
&CSIZE = DK/2

OBTAIN/DELTA0,PTS(1),PTS(2),PTS(3),PT6(1),PT6(2),P
T6(3)
DIM(5) = POINT/PTS(1),PTS(2),PTS(3)
DIM(6) = POINT/(6/7)*(PT6(1)),(6/7)*(PT6(2)),PT6(3)
DELTA0 = TRANSF/MAT13,DELTA0

DIM(7) = CIRCLE/CENTER,DIM(5),DIM(6),START,HO
EK(3),$
END,HOEK(3)+45
&FONT(DIM(7)) = &DASHED

OBTAIN/DELTA0,PT7(1),PT7(2),PT7(3),PT8(1),PT8(2),P
T8(3)
IF/PT8(1) > 0,JUMP/L311:

DIM(8) = DIMBP/DIM(7),NOTE,PT8(1)-2*DK,PT8(2)+D
K/5,$
'DELTA',ARROW,(6/7)*PT8(1),(6/7)*PT8(2),HOEK(3)+
135
JUMP/L312:
L311:

DIM(8) = DIMBP/DIM(7),NOTE,PT8(1)+DK/2,PT8(2)+
DK/5,$
'DELTA',ARROW,(6/7)*PT8(1),(6/7)*PT8(2),HOEK(3)+
135
L312:
DELETE/DELTA0
DIM(9) = POINT/PT6(1),PT6(2),PT6(3)

DIM(10) = CIRCLE/CENTER,DIM(5),DIM(9),START,HO
EK(3)+10,$
END,HOEK(3)-10
DIM3 = &SPOINT(DIM(10))
DIM4 = &EPOINT(DIM(10))
DELETE/DIM(9..10)
NOTE/DIM3(1),DIM3(2),STR(1)
NOTE/DIM4(1),DIM4(2),STR(2)
DELETE/DIM(5),DIM(6)
VIEWE/AUTO
&WCS = MSYS(4)
RPAINT
JUMP/L999:
$$
L320:
&CNMODE = &MODEL
&ENTCLR = &SYSCLR

OBTAIN/CLO,PTS(1),PTS(2),PTS(3),PT6(1),PT6(2),PT6(3
)
CLS(2) = LINE/PTS(1),PTS(2),PTS(1)+5,PTS(2)+5
CLS(3) = LINE/PTS(1),PTS(2),PTS(1)+5,PTS(2)-5
CLS(4) = LINE/PT6(1),PT6(2),PT6(1)-5,PT6(2)+5
CLS(5) = LINE/PT6(1),PT6(2),PT6(1)-5,PT6(2)-5
&CNMODE = &VIEW
&ENTCLR = &WHITE
&CSIZE = DK/2
STR(10) = 'VOLGERBAAN'
NOTE/PT6(1)+90,PT6(2),STR(10)
NOTE/PTS(1)+15.5,STR(2)
NOTE/PT6(1)-15.5,STR(1)
VIEWE/AUTO
JUMP/L999:
$$
L996:
MESSG/'DRIJFAS EN VOLGERBAAN NIET
LOODRECHT',,$
'CONSTRUCTIE VLAK STANGENMECHANISME
NIET MOGELIJK !'
DELETE/PUNT3
L996A:
DELETE/CLK,CLO
&WCS = KSYS(1)
RPAINT
JUMP/L25:
$$
L997:
MESSG/'DRIJFAS EN VOLGERAS NIET
EVENWIJDIG',,$
'CONSTRUCTIE VLAK STANGENMECHANISME
NIET MOGELIJK !'
DELETE/PUNT3
L997A:
DELETE/CLK,CLV
&WCS = KSYS(1)
RPAINT
JUMP/L25:
$$
L999:
&CNMODE = &MODEL
$$
HALT:
RETURN

```

## Bijlage 13. Listing MONTAGESTAND.GRS

```

$$$$-----
$$
$$ NAAM : MONTAGESTAND.GRS
$$ (DEELPROGRAMMA)
$$
$$ DIT PROGRAMMA LEEST DE
$$ TADSOL-UITVOER EN ROEPT
$$ VOOR DE GEKOZEN OPLOSSING DE
$$ BETREFFENDE MONTAGESTANDSCHETS-
$$ SUBROUTINE AAN
$$
$$ PROGRAMMEUR : H.J.M. VAN ABEELLEN
$$
$$ TU EINDHOVEN, VAKGROEP WPA
$$
$$ DATUM : 07-06-94
$$
$$ DECLARATIES=====
$$
PROC/KEUS2,LAYM,ALPHA0,DELTA0,DMACH,EMAC
H
ENTITY/ALPHA0,DELTA0
NUMBER/PAR(8,10),TYPE(10),TAU(10),LS(10),U0(10),P
NI(3)
STRING/KENM(10,1),STR(2,100),DATNAAM(40)
$$
$$ ALGEMEEN
$$
DATNAAM = '/users/std/abeelen/mechan_in.dat'
LAYER/WORK,LAYM+1,REF,LAYM
&ENTCLR=&MAGENT
&CSIZE=DMACH/30
&LDEV=1
RPAINT
$$
$$ INLEZEN TADSOLUITVOER
$$
L100:
TEXT/'GEEF DATASETNAAM
MECHANISMEFILE',DATNAAM,RSP,DEFLT
JUMP/L100:,L999:,,,RSP
CREATE/TXT,1
FETCH/TXT,1,DATNAAM
PRINT/ '
PRINT/'MECHANISMEFILE '
PRINT/ '
FPRINT/1
PRINT/ '
M=1
L200:
READ/1,LINNO,(M*20),USING,$
'#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@,$
IFERR,ERR:STR(1)
IFTHEN/(SUBSTR(STR(1),1,2))= '-1'
M=M-1
JUMP/L300:
ENDIF
KENM(M)=SUBSTR(STR(1),5,1)
TYPE(M)=VALF(SUBSTR(STR(1),6,3))
NPAR=VALF(SUBSTR(STR(1),4,1))
P=1
DO/LOOPPAR:P,1,NPAR
PAR(P,M)=VALF(SUBSTR(STR(1),P*9,9))
LOOPPAR:
READ/1,LINNO,(M*20)+10,USING,$
'#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@,$
IFERR,ERR:STR(2)
TAU(M)=VALF(SUBSTR(STR(2),10,8))
LS(M)=VALF(SUBSTR(STR(2),20,1))
U0(M)=VALF(SUBSTR(STR(2),27,9))
$$ *****
$$PRINT/'kenm'+KENM(M)
$$PRINT/'type'+FSTR(TYPE(M))
$$DO/LOOP2:,L2,1,NPAR
$$PRINT/'par'+FSTR(L2)+FSTR(PAR(L2,M))
$$LOOP2:
$$PRINT/'TAU'+FSTR(TAU(M))
$$PRINT/'LS'+FSTR(LS(M))
$$PRINT/'U0'+FSTR(U0(M))
$$ *****
M=M+1
JUMP/L200:
L300:
$$
$$ KEUZE MECH
$$
PARAM/'KIES MECHANISME VOOR
MONTAGESTANDSCHETS',,$
'OPLOSSINGSNUMMER',INT,X,RSP
JUMP/L300:,L999:,,,RSP
IFTHEN/X==0 OR X>M
MESSG/'ONJUISTE KEUZE'
JUMP/L300:
ENDIF
IFTHEN/ASCII(KENM(X),1)-77<>KEUS2
MESSG/'BEWEGINGSKENMERK OPLOSSING
KOMT',,$
'NIET OVEREEN MET INTERFACEAFSPRAKEN'
JUMP/L300:
ENDIF
&DECPL=0
PRINT/ '

```



```
PRINT/'OPLOSSINGNR. '+FSTR(X)
&DECPL=2
```

```
$$
$$ AANROEPEN MECHANISMESUBROUTINE
$$ EN BEPALEN REFERENTIEHOEKEN
$$
```

```
AALPHA0= ANGLF(ALPHA0)
PNT = &EPOINT(ALPHA0)
IF/PNT(2)<0,AALPA0=AALPHA0+180
```

```
IF/KENM(X) = ='R',JUMP/L500:
IF/KENM(X) = ='S',JUMP/L600:
IF/KENM(X) = ='T',JUMP/L700:
```

```
L500:
ADELTA0= ANGLF(DELTA0)
PNT = &EPOINT(DELTA0)
IF/PNT(2)<0,ADELTA0=ADELTA0+180
```

```
IFTHEN/TYPE(X) = = 3
CALL/'R003',DMACH,ADELTA0,AALPHA0,X,PAR,LS,T
AU,U0
JUMP/L900:
ENDIF
```

```
L600:
ADELTA0= ANGLF(DELTA0)
PNT = &EPOINT(DELTA0)
IF/PNT(2)<0,ADELTA0=ADELTA0+180
```

```
IFTHEN/TYPE(X) = = 6
CALL/'S006',DMACH,ADELTA0,AALPHA0,X,PAR,LS,T
AU,U0
JUMP/L900:
ENDIF
```

```
L700:
IFTHEN/TYPE(X) = = 6
CALL/'T006',DMACH,EMACH,AALPHA0,X,PAR,LS,TA
U,U0
JUMP/L900:
ENDIF
```

```
IFTHEN/TYPE(X) = = 8
CALL/'T008',DMACH,EMACH,AALPHA0,X,PAR,LS,TA
U,U0
JUMP/L900:
ENDIF
```

```
L800:
MESSG/'DIT MECHANISME IS NOG NIET
OMZETBAAR', $
'IN EEN MONTAGESTANDSCHETS.'
```

```
L900:
VIEWE/AUTO
```

```
CHOOSE/'WILT U EEN ANDER MECHANISME
KIEZEN ?', $
'JA, ZELFDE MECH. FILE', $
'JA, ANDERE MECH. FILE', $
```

```
'NEE',DEFLT,1,RSP
JUMP/L900:,L999:,,,L950:,L999:,RSP
```

```
MASK/ALL
DELETE/ALL
PRINT/' '
PRINT/'_____',
JUMP/L300:
```

```
L950:
MASK/ALL
DELETE/ALL
PRINT/' '
PRINT/'_____',
JUMP/L100:
```

```
ERR:
MESSG/'FOUT.'
```

```
L999:
RETURN
```

## Bijlage 14. Listing S006.GRS

```

$$$$=====
$$
$$ NAAM : S006.GRS (SUBPROGRAMMA)
$$
$$ DIT PROGRAMMA ZET DE
$$ OPLOSSINGSGEGEVENS OM IN EEN
$$ MONTAGESTANDSCHETS IN DE
$$ MACHINETEKENING
$$
$$ PROGRAMMEUR : H.J.M. VAN ABELEN
$$
$$ TU EINDHOVEN, VAKGROEP WPA
$$
$$ DATUM : 08-06-94
$$
$$ DECLARATIES=====
$$

PROC/DMACH,ADELTA0,AALPHA0,X,PAR,LS,TAU,U
0

ENTITY/C1,C2,SP,ST(20),TW(10)

NUMBER/PAR(8,10),TAU(10),LS(10),U0(10)

$$
$$ BEPALEN REFERENTIEWAARDES
$$

PI=3.1415926
TAUG=TAU(X)*180/PI
U0G=U0(X)*180/PI

REFV=1
IF/LS(X)=1 OR LS(X)=3,REFV=-1
REFD=1
IF/LS(X)=2 OR LS(X)=3,REFD=-1

EXTA=1
IF/PAR(2,X)<0,EXTA=-1

TAUG=-1*REFD*TAUG

RKW=1+PAR(1,X)*PAR(1,X)-2*PAR(1,X)*COSF(TAUG)
)
PSI=ACOSF((RKW+PAR(4,X)*PAR(4,X)-PAR(3,X)*PAR(3,X))/
(2*PAR(4,X)*SQRTF(RKW)))
IF/REFD<0,PSI=-1*PSI
BETA=ATANF((PAR(1,X)*SINF(TAUG))/(1-PAR(1,X)*
COSF(TAUG)))
UM=ABSF(PAR(2,X))*(PSI+BETA)+U0G

L50:
IFTHEN/UM>=360
UM=UM-360
JUMP/L50:
ENDIF
L60:
IFTHEN/UM<0
UM=UM+360
JUMP/L60:
ENDIF

XTW1=(4/5)*DMACH
YTW1=0

L10:
PRINT/ '
PRINT/MECHANISMETYPE S006'
PRINT/ '

$$
$$ TANDWIELEN
$$

XL=SQRTF(XTW1*XTW1+YTW1*YTW1)
YL=SQRTF((DMACH-XTW1)*(DMACH-XTW1)+YTW
1*YTW1)

IFTHEN/REFD*REFV*EXTA==1
R2=YL/(3+ABSF(PAR(2,X)))
R1=R2*ABSF(PAR(2,X))
R3=R2

XTW3=XTW1+((R1+R2)/(R1+3*R2))*(DMACH-XTW1
)
YTW3=((2*R2)/(R1+3*R2))*YTW1

CALL/'TANDWIEL',3,XTW3,YTW3,R3,TW
CALL/'VASTSCHAR',2,XTW3,YTW3
ELSE
R2=YL/(1+ABSF(PAR(2,X)))
R1=R2*ABSF(PAR(2,X))
ENDIF

CALL/'TANDWIEL',1,XTW1,YTW1,R1,TW
CALL/'VASTSCHAR',1,XTW1,YTW1

XTW2=DMACH
YTW2=0

CALL/'TANDWIEL',2,XTW2,YTW2,R2,TW
PRINT/'TANDWIELVERHOUDING r1/r2 =
'+FSTR(R1/R2)

$$
$$ STANGEN
$$

LD=XL
LB=LD*PAR(4,X)
LA=LD*PAR(1,X)
LC=LD*PAR(3,X)
LV=2*R2

$$ BEPALEN STANG A

AGAMMA=ATANF(YTW1/XTW1)
TM=TAUG-AALPHA0+AGAMMA
IF/TM>=360,TM=TM-360
IF/TM<0,TM=TM+360

PRINT/'MONTAGEHOEK KRUK tm =
'+FSTR(TM)

X1STA=0
Y1STA=0

```

```

X2STA = COSF(TAUG + AGAMMA)*LA
Y2STA = SINP(TAUG + AGAMMA)*LA

CALL/'STANG',1,X1STA,Y1STA,X2STA,Y2STA,LA,ST

$$ BEPALEN STANG C

X1STC = X2STA
Y1STC = Y2STA

C1 = CIRCLE/X1STC,Y1STC,LC
C2 = CIRCLE/XTW1,YTW1,LB
IFTHEN/REFD = = 1
  SP = POINT/YLARGE,INTOF,C1,C2
ELSE
  SP = POINT/YSMALL,INTOF,C1,C2
ENDIF
OBTAIN/SP,XSP,YSP
DELETE/SP,C1,C2

X2STC = XSP
Y2STC = YSP

CALL/'STANG',3,X1STC,Y1STC,X2STC,Y2STC,LC,ST

$$ BEPALEN STANG B

X1STB = X2STC
Y1STB = Y2STC
X2STB = XTW1
Y2STB = YTW1

CALL/'STANG',2,X1STB,Y1STB,X2STB,Y2STB,LB,ST

$$ BEPALEN STANG D

X1STD = X2STB
Y1STD = Y2STB
X2STD = 0
Y2STD = 0

CALL/'STANG',4,X1STD,Y1STD,X2STD,Y2STD,LD,ST

$$ BEPALEN VOLGERSTANG V

D0 = UM + ADELTA0

X1STV = DMACH
Y1STV = 0
X2STV = COSF(D0)*LV + DMACH
Y2STV = SINP(D0)*LV

ST(5) = LINE/X1STV,Y1STV,X2STV,Y2STV

PRINT/'MONTAGEHOEK VOLGERAS          um =
'+FSTR(UM)

CALL/'VASTTW',1,ST(2),TW(1),Y1STB-YTW1
CALL/'VASTTW',2,ST(5),TW(2),Y2STV

$$
$$ EVENTUEEL WIJZIGEN POS.
$$ VOLGERSCHARNIER
$$

RPAINT

```

```

VIEWE/AUTO
L100:
CHOOSE/'WIJZIGEN POSITIE
VOLGERSCHARNIER?',$
'JA','NEE',DEFLT,2,RSP
JUMP/L100:,L999:,,,L999:,RSP

```

```

L110:
GPOS/'KIES NIEUWE PUNT',XTW1,YTW1,ZTW1,RSP
JUMP/L110:,L999:,,,RSP

```

```

PRINT/' '
PRINT/'_____',
MASK/ALL
DELETE/ALL
JUMP/L10:

```

```

L999:
RETURN

```

## Bijlage 15. Listing T008.GRS

```

$$$$=====
$$
$$ NAAM : T008.GRS (SUBPROGRAMMA)
$$
$$ DIT PROGRAMMA ZET DE
$$ OPLOSSINGSGEGEVENS OM IN EEN
$$ MONTAGESTANDSCHETS IN DE
$$ MACHINETEKENING
$$
$$ PROGRAMMEUR : H.J.M. VAN ABEELLEN
$$
$$ TU EINDHOVEN, VAKGROEP WPA
$$
$$ DATUM : 12-07-94
$$
$$ DECLARATIES=====
$$
PROC/DMACH,EMACH,AALPHA0,X,PAR,LS,TAU,U0
ENITTY/ST(20)
NUMBER/PAR(8,10),TAU(10),LS(10),U0(10)
$$
$$ BEPALEN REFERENTIEWAARDES
$$
PI=3.1415926
TAUG=-TAU(X)*180/PI
REFX=1
IF/LS(X)= =1 OR LS(X)= =3,REFX=-1
REFY=1
IF/LS(X)= =1 OR LS(X)= =2,REFY=-1
L10:
PRINT/ '
PRINT/'MECHANISMETYPE T008'
PRINT/ '
$$
$$ STANGEN
$$
LA=PAR(2,X)
LC=LA/PAR(1,X)
$$ BEPALEN STANG A
TM=90-REFX*90+REFX*REFY*TAUG-AALPHA0
IF/TM> =360,TM=TM-360
IF/TM<0,TM=TM+360
PRINT/'MONTAGEHOEK KRUK          tm =
'+FSTR(TM)
X1STA=0
Y1STA=0
X2STA=REFX*COSF(TAUG)*LA
Y2STA=REFY*SINF(TAUG)*LA
CALL/'STANG',1,X1STA,Y1STA,X2STA,Y2STA,LA,ST
$$$$=====
$$ BEPALEN STANG C
X1STC=X2STA
Y1STC=Y2STA
YC=ABSF(Y2STA)+LC*PAR(3,X)
XC=SQRTR(LC*LC-(YC*YC))
X2STC=X1STC+REFX*XC
Y2STC=Y1STC-REFY*YC
CALL/'STANG',3,X1STC,Y1STC,X2STC,Y2STC,LC,ST
CALL/'SCHUIF',1,X2STC,Y2STC,0
$$ BEPALEN STANG S
X1STS=X2STC
Y1STS=Y2STC
CF10=SQRTR(1-(PAR(3,X)*PAR(3,X)))
X2STS=X1STS+U0(X)-REFX*(LA+LC*CF10)
Y2STS=EMACH
LS1=ABSF(X1STS-X2STS)*ABSF(X1STS-X2STS)
LS2=ABSF(Y1STS-Y2STS)*ABSF(Y1STS-Y2STS)
LSS=SQRTR(LS1+LS2)
CALL/'STANG',19,X1STS,Y1STS,X2STS,Y2STS,LSS,ST
CALL/'SCHUIF',2,X2STS,Y2STS,0
PRINT/'Montagestand heugel      sm =
'+FSTR(SM)
PRINT/'Excentriciteit eindschui f      e =
'+FSTR(EMACH)
PRINT/'Excentriciteit tussenschui f      =
'+FSTR(Y2STC)
RPAINT
L999:
RETURN

```

## Bijlage 16. Listing R003.GRS

```

$$=====
$$
$$ NAAM : R003.GRS (SUBPROGRAMMA)
$$
$$ DIT PROGRAMMA ZET DE
$$ OPLOSSINGSGEGEVENS OM IN EEN
$$ MONTAGESTANDSCHETS IN DE
$$ MACHINETEKENING
$$
$$ PROGRAMMEUR : H.J.M. VAN ABEELLEN
$$
$$ TU EINDHOVEN, VAKGROEP WPA
$$
$$ DATUM : 20-06-94
$$
$$ DECLARATIES=====
$$
PROC/DMACH,ADELTA0,AALPHA0,X,PAR,LS,TAU,U
0

ENTITY/C1,C2,SP,ST(20),TW(10)

NUMBER/PAR(8,10),TAU(10),LS(10),U0(10)

$$
$$ BEPALEN REFERENTIEWAARDES
$$
PI=3.1415926
TAUG=TAU(X)*180/PI
U0G=U0(X)*180/PI

REFV=1
IF/LS(X)= =1 OR LS(X)= =3,REFV=-1
REFD=1
IF/LS(X)= =2 OR LS(X)= =3,REFD=-1

EXTA=1
IF/PAR(2,X)<0,EXTA=-1

TAUG=-1*REFD*TAUG

FI=ASINF(PAR(1,X)*SINF(TAUG))
UM=ABSF(PAR(2,X))*(TAUG+FI)+U0G

L50:
IFTHEN/UM>=360
UM=UM-360
JUMP/L50:
ENDIF
L60:
IFTHEN/UM<0
UM=UM+360
JUMP/L60:
ENDIF

XTW1=(4/5)*DMACH

YTW1=0

L10:
PRINT/ '
PRINT/'MECHANISMETYPE R003'

PRINT/ '
$$
$$ TANDWIELEN
$$
XL=SQRTF(XTW1*XTW1+YTW1*YTW1)
YL=SQRTF((DMACH-XTW1)*(DMACH-XTW1)+YTW
1*YTW1)

IFTHEN/REFD*REFV*EXTA= =1
R2=YL/(3+ABSF(PAR(2,X)))
R1=R2*ABSF(PAR(2,X))
R3=R2

XTW3=XTW1+((R1+R2)/(R1+3*R2))*(DMACH-XTW1
)
YTW3=((2*R2)/(R1+3*R2))*YTW1

CALL/'TANDWIEL',3,XTW3,YTW3,R3,TW
CALL/'VASTSCHAR',2,XTW3,YTW3
ELSE
R2=YL/(1+ABSF(PAR(2,X)))
R1=R2*ABSF(PAR(2,X))
ENDIF

CALL/'TANDWIEL',1,XTW1,YTW1,R1,TW
CALL/'VASTSCHAR',1,XTW1,YTW1

XTW2=DMACH
YTW2=0

CALL/'TANDWIEL',2,XTW2,YTW2,R2,TW
PRINT/'TANDWIELVERHOUDING r1/r2 =
'+FSTR(R1/R2)

$$
$$ STANGEN
$$

LD=XL
LB=LD/PAR(1,X)
LA=LB+LD+30

LV=2*R2

$$ BEPALEN SCHUIFSTANG A

AGAMMA=ATANF(YTW1/XTW1)
TM=TAUG-AALPHA0+AGAMMA
IF/TM>=360,TM=TM-360
IF/TM<0,TM=TM+360

PRINT/'MONTAGEHOEK KRUK tm =
'+FSTR(TM)

X1STA=0
Y1STA=0
X2STA=COSEF(TAUG+AGAMMA)*LA
Y2STA=SINF(TAUG+AGAMMA)*LA

ST(1)=LINE/X1STA,Y1STA,X2STA,Y2STA

$$ BEPALEN STANG B

```

```

C1 = CIRCLE/LD,0,LB
IFTHEN/Y2STA >= 0
  SP = POINT/YLARGE,INTOF,ST(1),C1
ELSE
  SP = POINT/YSMALL,INTOF,ST(1),C1
ENDIF
OBTAIN/SP,X1STB,Y1STB
DELETE/C1,SP
X2STB = XTW1
Y2STB = YTW1

CALL/'STANG',2,X1STB,Y1STB,X2STB,Y2STB,LB,ST

$$ BEPALEN STANG D

X1STD = X2STB
Y1STD = Y2STB
X2STD = 0
Y2STD = 0

CALL/'STANG',4,X1STD,Y1STD,X2STD,Y2STD,XL,ST

$$ BEPALEN VOLGERSTANG V

D0 = UM + ADELTA0

X1STV = DMACH
Y1STV = 0
X2STV = COSF(D0)*LV + DMACH
Y2STV = SINF(D0)*LV

ST(5) = LINE/X1STV,Y1STV,X2STV,Y2STV

$$ BEPALEN SCHUIF

CALL/'SCHUIF',1,X1STB,Y1STB,TAUG

PRINT/'MONTAGEHOEK VOLGERAS      um =
'+FSTR(UM)

CALL/'VASTTW',1,ST(2),TW(1),Y1STB
CALL/'VASTTW',2,ST(5),TW(2),Y2STV

$$
$$ EVENTUEEL WIJZIGEN POS.
VOLGERSCHARNIER
$$

RPAINT
VIEWE/AUTO
L100:
CHOOSE/'WIJZIGEN POSITIE
VOLGERSCHARNIER?',$
'JA','NEE',DEFLT,2,RSP
JUMP/L100:,L999:,,,L999:,RSP

L110:
GPOS/'KIES NIEUWE PUNT',XTW1,YTW1,ZTW1,RSP
JUMP/L110:,L999:,,,L999:,RSP

PRINT/' '
PRINT/'_____',
MASK/ALL
DELETE/ALL
JUMP/L10:

```

## Bijlage 17. Listing T006

```

$$=====
$$
$$ NAAM : T006.GRS (SUBPROGRAMMA)
$$
$$ DIT PROGRAMMA ZET DE
$$ OPLOSSINGSgegevens OM IN EEN
$$ MONTAGESTANDSCHETS IN DE
$$ MACHINETEKENING
$$
$$ PROGRAMMEUR : H.J.M. VAN ABEELLEN
$$
$$ TU EINDHOVEN, VAKGROEP WPA
$$
$$ DATUM : 18-8-93
$$
$$ DECLARATIES=====
$$

PROC/DMACH,EMACH,AALPHA0,X,PAR,LS,TAU,U0

ENTITY/C1,C2,SP,ST(20),TW(10)

NUMBER/PAR(8,10),TAU(10),LS(10),U0(10)

$$
$$ BEPALEN REFERENTIEWAARDES
$$

PI= 3.1415926
TAUG=TAU(X)*180/PI

REFV=1
IF/LS(X)= =1 OR LS(X)= =3,REFV=-1
REFD=1
IF/LS(X)= =2 OR LS(X)= =3,REFD=-1

HAZ=-1
IF/PAR(2,X)<0,HAZ=1

TAUG=-1*REFD*TAUG

RKW=1+PAR(1,X)*PAR(1,X)-2*PAR(1,X)*COSF(TAUG)
)
PSI=ACOSF((RKW+PAR(4,X)*PAR(4,X)-PAR(3,X)*PA
R(3,X))/
(2*PAR(4,X)*SQRTF(RKW)))
IF/REFD<0,PSI=-1*PSI
BETA=ATANF((PAR(1,X)*SINF(TAUG))/(1-PAR(1,X)*
COSF(TAUG)))
SM=ABSF(PAR(2,X))*((PSI+BETA)*PI/180)+U0(X)

XTW1=SM

R1=ABSF(PAR(2,X))
IF/THEN/REFD*REFV*HAZ= =-1
YTW1=EMACH+R1
ELSE
YTW1=EMACH-R1
ENDIF

L10:
PRINT/ '
PRINT/'MECHANISMETYPE T006'

PRINT/ '
$$
$$ TANDWIEL EN HEUGEL
$$
XL=SQRTF(XTW1*XTW1+YTW1*YTW1)

CALL/'TANDWIEL',1,XTW1,YTW1,R1,TW
CALL/'VASTSCHAR',1,XTW1,YTW1

CALL/'HEUGEL',1,SM,EMACH,0

PRINT/'MONTAGESTAND HEUGEL          sm =
'+FSTR(SM)

PRINT/'EXCENTRICITEIT          e =
'+FSTR(EMACH)

$$
$$ STANGEN
$$

LD=XL
LB=LD*PAR(4,X)
LA=LD*PAR(1,X)
LC=LD*PAR(3,X)

$$ BEPALEN STANG A

AGAMMA=ATANF(YTW1/XTW1)
TM=TAUG-AALPHA0+AGAMMA
IF/TM> =360,TM=TM-360
IF/TM<0,TM=TM+360

PRINT/'MONTAGEHOEK KRUK          tm =
'+FSTR(TM)

X1STA=0
Y1STA=0
X2STA=COSF(TAUG+AGAMMA)*LA
Y2STA=SINF(TAUG+AGAMMA)*LA

CALL/'STANG',1,X1STA,Y1STA,X2STA,Y2STA,LA,ST

$$ BEPALEN STANG C

X1STC=X2STA
Y1STC=Y2STA

C1=CIRCLE/X1STC,Y1STC,LC
C2=CIRCLE/XTW1,YTW1,LB
IF/THEN/REFD= =1
SP=POINT/YLARGE,INTOF,C1,C2
ELSE
SP=POINT/YSMALL,INTOF,C1,C2
ENDIF
OBTAIN/SP,XSP,YSP
DELETE/SP,C1,C2

X2STC=XSP
Y2STC=YSP

```

CALL/'STANG',3,X1STC,Y1STC,X2STC,Y2STC,LC,ST

\$\$ BEPALEN STANG B

X1STB=X2STC

Y1STB=Y2STC

X2STB=XTW1

Y2STB=YTW1

CALL/'STANG',2,X1STB,Y1STB,X2STB,Y2STB,LB,ST

\$\$ BEPALEN STANG D

X1STD=X2STB

Y1STD=Y2STB

X2STD=0

Y2STD=0

CALL/'STANG',4,X1STD,Y1STD,X2STD,Y2STD,XL,ST

CALL/'VASTTW',1,SI(2),TW(1),Y1STB-YTW1

\$\$

\$\$ EVENTUEEL WIJZIGEN POS.

VOLGERSCHARNIER

\$\$

RPAINI

VIEWE/AUTO

L100:

CHOOSE/'WIJZIGEN POSITIE

VOLGERSCHARNIER?',\$

'JA','NEE',DEFLT,2,RSP

JUMP/L100:,L999:,,,L999:,RSP

L110:

GPOS/'KIES NIEUWE

PUNT',XTW1,YDUM,ZDUM,RSP

JUMP/L110:,L999:,,RSP

PRINT/' '

PRINT/'\_\_\_\_\_'

MASK/ALL

DELETE/ALL

JUMP/L10:

L999:

RETURN



**Bijlage 18. Listing STANG.GRS**

```
$$=====
$$
$$ NAAM : STANG.GRS (SUBROUTINE)
$$
$$ DIT PROGRAMMA TEKENT DE STANGEN IN
$$ EEN MONTAGESTANDSCHETS
$$
$$ PROGRAMMEUR : H.J.M. VAN ABELEN
$$
$$ TU EINDHOVEN, VAKGROEP WPA
$$
$$ DATUM : 08-06-94
$$
$$ DECLARATIES=====
$$

PROC/N,X1ST,Y1ST,X2ST,Y2ST,LST,ST

ENTITY/ST(20)

ST(N)=LINE/X1ST,Y1ST,X2ST,Y2ST
PRINT/'STANGLENGTE
'+CHRSTR(96+N)+' = '+FSTR(LST)
&ENTCLR=&WHITE
NOTE/(X1ST+X2ST)/2,(Y1ST+Y2ST)/2,CHRSTR(96+N
)
&ENTCLR=&MAGENT

RETURN
```

**Bijlage 19. Listing TANDWIEL.GRS**

```

$$=====
$$
$$ NAAM : TANDWIEL.GRS (SUBROUTINE)
$$
$$ DIT PROGRAMMA TEKENT DE TANDWIELEN
$$ IN EEN MONTAGESTANDSCHETS
$$
$$ PROGRAMMEUR : H.J.M. VAN ABELEN
$$
$$ TU EINDHOVEN, VAKGROEP WPA
$$
$$ DATUM : 08-06-94
$$
$$ DECLARATIES=====
$$

PROC/N,XTW,YTW,RTW,TW

ENTITY/TW(10),RA(10),RA1(10),RA2(10)

STRING/STR(2)

TW(N)=CIRCLE/XTW,YTW,RTW
&FONT(TW(N))=&DASHED
&DECPL=0
STR=FSR(N)
&DECPL=2
PRINT/'TANDWIELSTRAAL          r'+STR+' =
'+FSR(RTW)
XR=XTW+0.707*RTW
YR=YTW+0.707*RTW
&ENTCLR=&WHITE
&CNMODE=&VIEW
RA(N)=LINE/XTW,YTW,XR,YR
RA1(N)=LINE/XR,YR,XR-3,YR-2
RA2(N)=LINE/XR,YR,XR-2,YR-3
NOTE/XTW+0.35*RTW,YTW+0.35*RTW,'r'+STR
&ENTCLR=&MAGENT
&CNMODE=&MODEL

RETURN

```

**Bijlage 20. Listing SCHUIF.GRS**

```
$$=====
$$
$$ NAAM : SCHUIF.GRS (SUBROUTINE)
$$
$$ DIT PROGRAMMA TEKENT DE SCHUIVEN IN
$$ EEN MONTAGESTANDSCHETS
$$
$$ PROGRAMMEUR : H.J.M. VAN ABELEN
$$
$$ TU EINDHOVEN, VAKGROEP WPA
$$
$$ DATUM : 08-06-94
$$
$$ DECLARATIES=====
$$

PROC/N,XSCH,YSCH,ASCH

ENTITY/L1(10),L2(10),L3(10),L4(10),SCH(10)

NUMBER/MAT1(12),MAT2(12)

L1(N)=LINE/15,6,15,-6
L2(N)=LINE/15,-6,-15,-6
L3(N)=LINE/-15,-6,-15,6
L4(N)=LINE/-15,6,15,6
SCH(N)=GROUP/L1(N),L2(N),L3(N),L4(N)
MAT1=MATRIX/XYROT,ASCH
SCH(N+1)=TRANSF/MAT1,SCH(N)
DELETE/SCH(N)
MAT2=MATRIX/TRANSL,XSCH,YSCH,0
SCH(N+2)=TRANSF/MAT2,SCH(N+1)
DELETE/SCH(N+1)

RETURN
```

**Bijlage 21. Listing HEUGEL.GRS**

```

$$=====
$$
$$ NAAM : HEUGEL.GRS (SUBROUTINE)
$$
$$ DIT PROGRAMMA TEKENT DE HEUGELS IN
$$ EEN MONTAGESTANDSCHETS
$$
$$ PROGRAMMEUR : H.J.M. VAN ABBELEN
$$
$$ TU EINDHOVEN, VAKGROEP WPA
$$
$$ DATUM : 08-06-94
$$
$$ DECLARATIES=====
$$

PROC/N,XSCH,YSCH,ASCH

ENTITY/L1(10),L2(10),L3(10),L4(10),L5(10),L6(10),SCH(1
0)

NUMBER/MAT1(12),MAT2(12)

L1(N)=LINE/35,6,35,-6
L2(N)=LINE/35,-6,-35,-6
L3(N)=LINE/-35,-6,-35,6
L4(N)=LINE/-35,6,35,6
L5(N)=LINE/0,6,0,-6
L6(N)=LINE/-35,0,35,0
&FONT(L6(N))=&DASHED
SCH(N)=GROUP/L1(N),L2(N),L3(N),L4(N),L5(N),L6(N)
MAT1=MATRIX/XYROT,ASCH
SCH(N+1)=TRANSF/MAT1,SCH(N)
DELETE/SCH(N)
MAT2=MATRIX/TRANSL,XSCH,YSCH,0
SCH(N+2)=TRANSF/MAT2,SCH(N+1)
DELETE/SCH(N+1)

RETURN

```

**Bijlage 22. Listing VASTTW.GRS**

```
$$=====
$$
$$ NAAM : VASTSTW.GRS (SUBROUTINE)
$$
$$ DIT PROGRAMMA TEKENT DE VERBINDING
$$ TUSSEN STANG EN TANDWIEL IN EEN
$$ MONTAGESTANDSCHETS
$$
$$ PROGRAMMEUR : H.J.M. VAN ABELEN
$$
$$ TU EINDHOVEN, VAKGROEP WPA
$$
$$ DATUM : 08-06-94
$$
$$ DECLARATIES=====
$$
```

```
PROC/N,ST,TW,YST
```

```
ENTITY/VT(10),ST,TW,SP
```

```
&CNMODE=&VIEW
```

```
IFTHEN/YST>=0
```

```
SP=POINT/YLARGE,INTOF,ST,TW
```

```
ELSE
```

```
SP=POINT/YSMALL,INTOF,ST,TW
```

```
ENDIF
```

```
OBTAIN/SP,XVT,YVT
```

```
VT(N)=CIRCLE/XVT,YVT,3
```

```
DELETE/SP
```

```
&CNMODE=&MODEL
```

```
RETURN
```

**Bijlage 23. Listing VASTSCHAR.GRS**

```

$$=====
$$
$$ NAAM : VASTSCHAR.GRS (SUBROUTINE)
$$
$$ DIT PROGRAMMA TEKENT DE VASTE
$$ SCHARNIEREN IN EEN
$$ MONTAGESTANDSCHETS
$$
$$ PROGRAMMEUR : H.J.M. VAN ABELEN
$$
$$ TU EINDHOVEN, VAKGROEP WPA
$$
$$ DATUM : 08-06-94
$$
$$ DECLARATIES=====
$$

PROC/N,XVS,YVS

ENTITY/VS(20),VS1(20),VS2(20),VS3(20),VS4(20),VSS(20
)

&CNMODE=&VIEW
&ENTCLR=&WHITE
VS(N)=CIRCLE/XVS,YVS,4
VS1(N)=LINE/XVS,YVS+4,XVS,YVS-4
VS2(N)=LINE/XVS+4,YVS,XVS-4,YVS
VS3(N)=LINE/XVS+3.8,YVS+1.4,XVS-3.8,YVS-1.4
VS4(N)=LINE/XVS+2.8,YVS+2.8,XVS-2.8,YVS-2.8
VSS(N)=LINE/XVS+1.4,YVS+3.8,XVS-1.4,YVS-3.8
&ENTCLR=&MAGENT
&CNMODE=&MODEL

RETURN

```