

## Modellering van een tandriemaandrijving

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

Vaassen, W. M. H. (1987). *Modellering van een tandriemaandrijving*. (DCT rapporten; Vol. 1987.038). Technische Universiteit Eindhoven.

***Document status and date:***

Gepubliceerd: 01/01/1987

***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.

**MODELLERING VAN EEN  
TANDRIEMAANDRIJVING**

**WFW 87.038**

Verslag in het kader van een stage-opdracht.

Vakgroep WFW  
Faculteit Werktuigbouwkunde  
Technische Universiteit Eindhoven

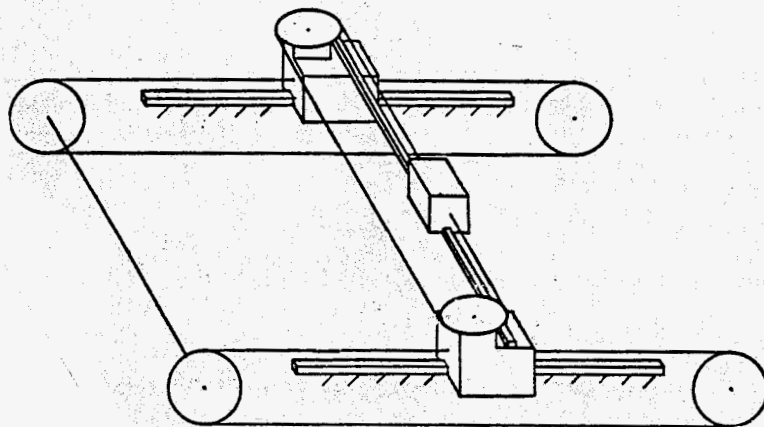
W.M.H. Vaassen

Juni 1987

Dr.ir. A.A.H.J. Sauren en ir. W.P. Koppens  
dank ik voor het begeleiden van de stage.

## Inleiding

In dit verslag wordt de modellering van een tandriemaandrijving beschreven. Deze tandriemaandrijving maakt deel uit van een zgn. X-Y-tafel. Deze bestaat uit een rechthoekig horizontaal frame waarover aan iedere kant in langsrichting een slede kan bewegen. Op deze sledes is een geleiding bevestigd waarover een dwarsslede kan bewegen.



**X-Y-tafel**

figuur 1

De stageopdracht luidt:

**Modelleer 1 tandriemaandrijving**

**Maak een bewegingssimulatie m.b.v. het programmapakket DADS**

**Verifieer de resultaten met een zelfgeschreven programma.**

Er is gekozen voor verwerking met DADS omdat een model met dit pakket gemakkelijk uitgebreid kan worden. Het programma vraagt alleen de invoer van een aantal elementen als 'BODIES', 'CONSTRAINTS', 'INITIAL CONDITIONS' enz. Het opstellen en oplossen van de bewegingsvergelijkingen gebeurt automatisch. Dit heeft echter wel het nadeel dat men weinig zicht heeft op de nauwkeurigheid van de resultaten. Dat is dan ook de reden om hetzelfde probleem ook met een zelfgeschreven programma op te lossen.

Er is een model opgesteld van een tandriem die dient voor het verplaatsen van een langsslede. In dit model is ook de massatraagheid van de motor betrokken.

Allereerst zijn de bewegingsvergelijkingen opgesteld en opgelost met een eigen programma. Daarna is het model ingevoerd in DADS waarna ook hiermee de responsie op hetzelfde ingangssignaal is berekend.

De verschillen tussen de resultaten en de voor- en nadelen van beide methoden worden besproken. Ook de bruikbaarheid van het model komt aan de orde.

INHOUDSOPGAVE

Lijst van gebruikte symbolen	7
------------------------------	---

Hoofdstuk 1: Modellering

1.1 Modelbeschrijving	11
1.2 Fysische achtergrond	13
1.3 Keuze van de coördinaten	14
1.4 Krachten in het model	15
1.5 Bewegingsvergelijkingen	17
1.6 Keuze van de modelparameters	18

Hoofdstuk 2: Simulatie

2.1 Eigen programma	
- differentiaalvergelijkingen	19
- wrijvingsmodel	20
- tolerantie	20
2.2 Verwerking met DADS	
- elementen invoeren	21
- userforces	21
- wrijvingsmodel	22
2.3 Resultaten	
- algemeen	23
- nauwkeurigheid	28
- vergelijking tussen eigen programma en DADS	28

Hoofdstuk 3: Evaluatie van modellering en simulatie

3.1 Fysische relevantie van het model	30
3.2 Aanbevelingen voor vervolgonderzoek	31

Appendices:

A Afleiding van de bewegingsvergelijkingen	35
B Sourcetekst van het eigen programma	38
C DADS-invoerfile + wijzigingen in de hulproutines	46

Lijst van gebruikte symbolen

A	dwarsdoorsnede-oppervlak van de riem	$m^2$
$A_r$	referentielengte linker riemdeel	m
d	expansie van de rol-as-afstand	m
E	over de dwarsdoorsnede gemiddelde elasticiteitsmodulus in langsrichting van het riemmateriaal	$N m^{-2}$
$\underline{E}$	krachtenvektor	
$F_1(\underline{q}, \dot{\underline{q}})$	wrijvingskracht op de slede	N
$F_2(\underline{q})$	reaktiekracht tussen slede en riemblokje	N
$F_3(\underline{q})$	reaktiekracht op de aandrijfrol-as	N
$F_4(\underline{q})$	reaktiekracht op de spanrol-as	N
$F_5(\underline{q})$	riemspankracht	N
$F_6(\underline{q})$	riemspankracht	N
$F_7(\underline{q})$	riemspankracht	N
$F_w$	maximale grootte van de wrijvingskracht op de slede ( $\geq 0$ )	N
i	overbrengingsverhouding	-



$i_3$	traagheidsmoment van aandrijfrol en as	$\text{kg m}^2$
$i_4$	traagheidsmoment van de spanrol	$\text{kg m}^2$
$i_m$	traagheidsmoment van het motor-anker	$\text{kg m}^2$
$k_2$	stijfheid van de riemblokveren	$\text{N m}^{-1}$
$k_3$	stijfheid van de aandrijfrolbevestiging	$\text{N m}^{-1}$
$k_4$	stijfheid van de spanrolbevestiging	$\text{N m}^{-1}$
$k^*$	$k^* \gg k_2$	$\text{N m}^{-1}$
$l_1$	referentie rol-afstand	m
$l_2$	lengte riemblokje	m
$M(t)$	aandrijfmoment op het motor-anker	$\text{N m}$
$[M]$	massamatrix	
$m_1$	massa van de slede	kg
$m_2$	massa van het riemblokje	kg
$m_3$	massa van de aandrijfrol	kg
$m_4$	massa van de spanrol	kg

$\underline{Q}(\underline{q}, \dot{\underline{q}}, t)$	gegeneraliseerde krachtenvektor	
$\underline{q}$	vektor van onafhankelijke coördinaten	
$r$	straal van de riemrollen	m
$s$	speling tussen riemblokveren en slede	m
$T(\underline{q}, \dot{\underline{q}})$	totale kinetische energie	N m
$t$	tijd	sec
$v$	maximale indrukking van een veer van het riemblokje	m

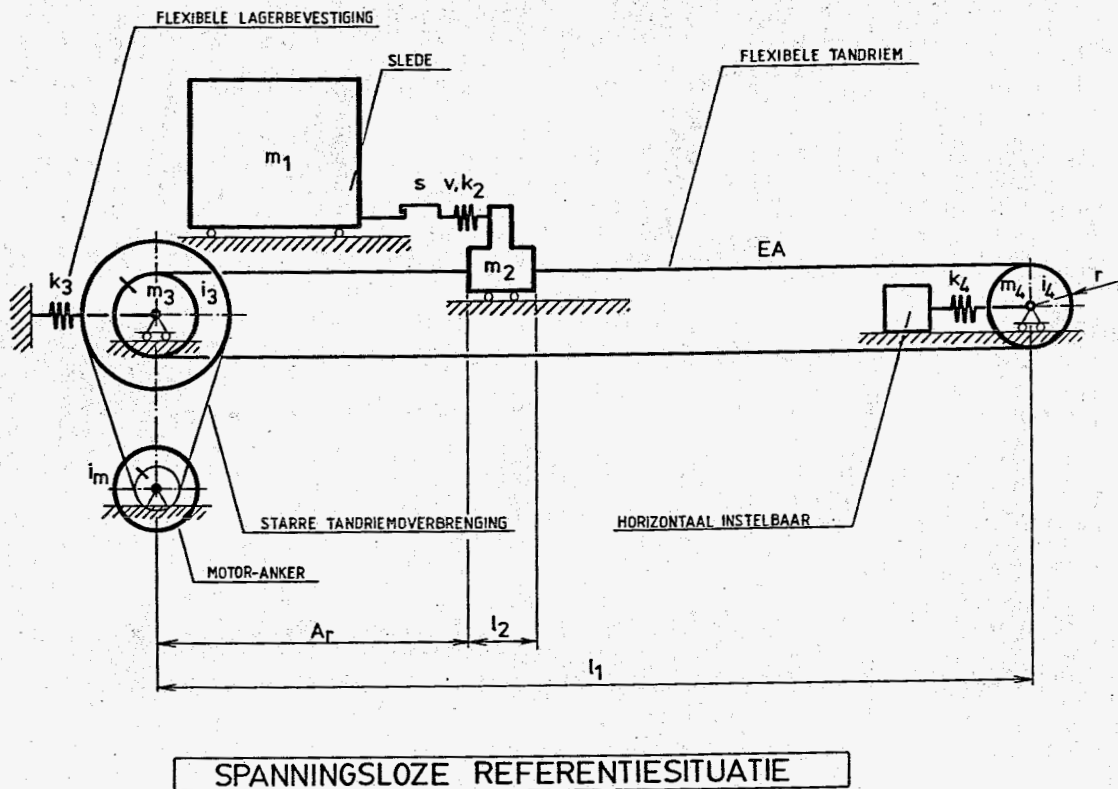
$\underline{x}$	koördinatenvektor	
$x_1$	verplaatsing van de slede t.o.v. de referentiepositie	m
$x_2$	verplaatsing van het riemblokje t.o.v. de referentiepositie	m
$x_3$	verplaatsing van de aandrijfrol t.o.v. de referentiepositie	m
$\varphi_3$	verdraaiing van de aandrijfrol t.o.v. de referentiepositie	rad
$x_4$	verplaatsing van de spanrol t.o.v. de referentiepositie	m
$\varphi_4$	verdraaiing van de spanrol t.o.v. de referentiepositie	rad
$\varphi_m$	verdraaiing van het motor-anker t.o.v. de referentiepositie	rad
$\underline{y}$	$\underline{y}^T = ( \underline{q}^T , \underline{\dot{q}}^T )$	
$\underline{\varphi}(t, \underline{y})$	afgeleide naar de tijd van $\underline{y}$	

## Hoofdstuk 1

### Modellering

#### § 1.1 Beschrijving van het model.

Bij het kiezen van vrijheidsgraden en modelparameters is uitgegaan van de konstruktie die schematisch is weergegeven in onderstaande figuur.



model van de tandriemaandrijving

figuur 2

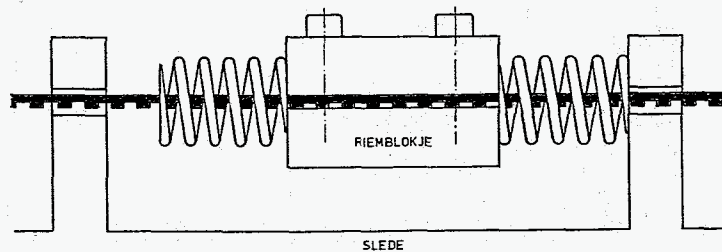
Het model bevat een massa  $m_1$  die flexibel verbonden is met een riemblokje  $m_2$ . Dit riemblokje is star verbonden met de tandriem.

De tandriem wordt opgespannen door een aandrijfrol  $m_3$  en een spanrol  $m_4$ . Beide rollen zijn flexibel aan de vaste wereld verbonden. Het bevestigingspunt van  $m_4$  kan vrij gekozen worden zodat de voorspanning van de riem instelbaar is.

De aandrijfrol is met een starre overbrenging gekoppeld aan het motor-anker.

§ 1.2 De fysische achtergrond.

De slede is gekoppeld met een blokje dat op de tandriem geklemd is. Deze koppeling heeft een instelbare speling. De konstruktieve uitvoering daarvan is weergegeven in figuur 3.



**slede - riembevestiging**

figuur 3

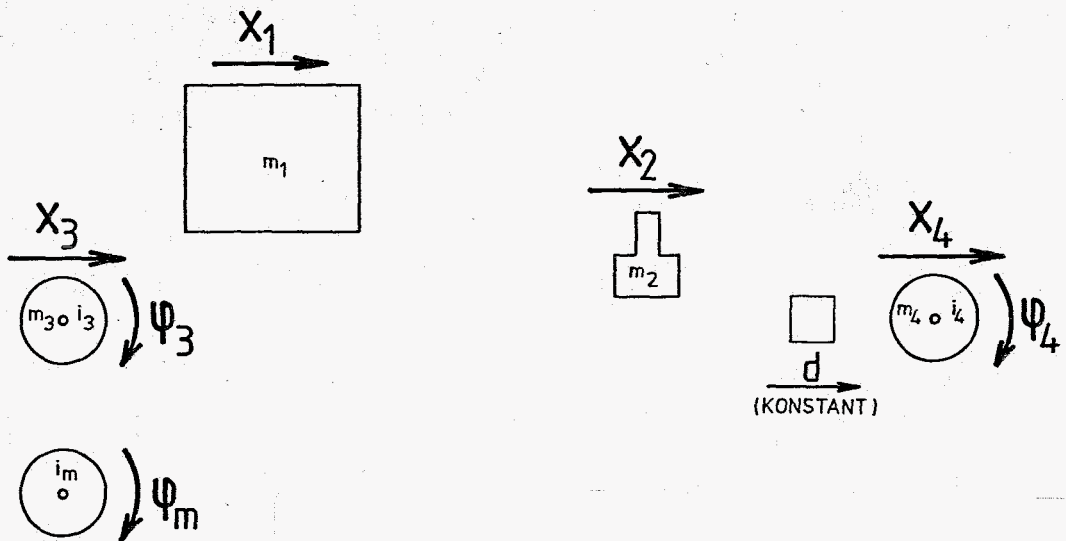
De rollen die de riem geleiden zijn bevestigd op vrije as-einden. De doorbuiging van deze as-einden kan in de orde grootte van 1 mm. zijn. De positie van de asbevestiging van de spanrol  $m_4$  kan in x-richting ingesteld worden.

De aandrijving vindt plaats via een tandriem-overbrenging tussen de motor en de as van de aandrijfrol. Deze riem is kort in vergelijking met de riem die de slede aandrijft en wordt minder zwaar belast. De verlenging van deze riem wordt in het model dan ook verwaarloosd.

§ 1.3 Keuze van de coördinaten.

Als coördinaten worden de volgende grootheden gekozen:

$x_1, x_2, x_3, \varphi_3, x_4, \varphi_4$  en  $\varphi_m$ .



coördinaten van het model

figuur 4

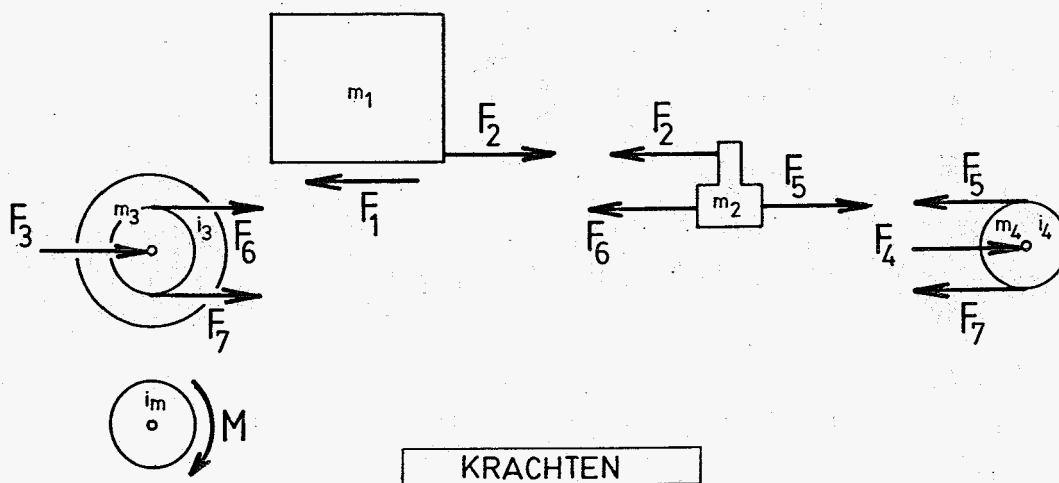
De coördinaten zijn relatief t.o.v. de spanningsloze referentiesituatie gekozen. De keuze van de coördinaten ligt voor de hand; ze stellen alle verplaatsingen of hoekverdraaiingen voor. Bovendien zijn de krachten  $F_2$  t/m  $F_7$  (zie volgende §) gemakkelijk uit te drukken in deze grootheden. Het aantal onafhankelijke coördinaten dat het model beschrijft is 1 kleiner dan het totale aantal coördinaten omdat er geldt:

$$i \varphi_3 - \varphi_m = 0$$

( 1 )

§ 1.4 Krachten in het model.

Op de slede werkt een Coulombse wrijvingskracht  $F_1(\underline{q}, \dot{\underline{q}})$  en de kracht  $F_2(\underline{q})$  van de koppeling met het riemblokje.



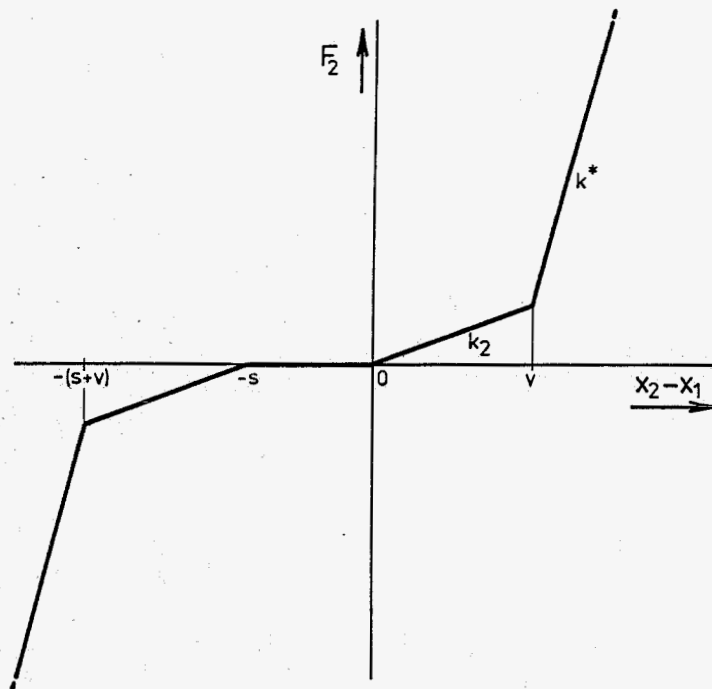
figuur 5

Op het motor-anker werkt een uitwendig koppel  $M(t)$ .  $M(t)$  wordt beschouwd als ingangssignaal.

De verbindingskrachten ten gevolge van (1) zijn niet weergegeven!



De koppeling tussen  $m_1$  en  $m_2$  heeft een konstitutief gedrag als aangegeven in figuur 6:



konstitutief gedrag van de koppeling  
tussen slede en riemblokje

figuur 6

De vergelijkingen voor de krachten zijn vermeld in appendix A.

§ 1.5 De bewegingsvergelijkingen

Enkele definities:

$$\underline{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \varphi_3 \\ x_4 \\ \varphi_4 \end{bmatrix} \quad \underline{Q}(\underline{q}, \dot{\underline{q}}, t) = \begin{bmatrix} -F_1 + F_2 \\ -F_2 + F_5 - F_6 \\ F_3 + F_6 + F_7 \\ (F_6 - F_7) r + i M(t) \\ F_4 - F_5 - F_7 \\ (-F_5 + F_7) r \end{bmatrix}$$

$$[M] = \begin{bmatrix} m_1 & & & & & \\ & m_2 & & & & \\ & & m_3 & & & \\ & & & i_3 + i^2 i_m & & \\ & & & & m_4 & \\ & & & & & i_4 \end{bmatrix} \quad (\text{diagonaalmatrix})$$

Via Lagrange volgt het stelsel bewegingsvergelijkingen:

$$[M] \underline{\ddot{q}} = \underline{Q}(\underline{q}, \dot{\underline{q}}, t) \quad (2)$$

(Zie appendix A)

§ 1.6 De keuze van de modelparameters

Het model is slechts een schematisering van de echte konstruktie. Men kan de modelparameters zo kiezen dat het gedrag van model en konstruktie in een bepaald opzicht optimaal overeenstemmen.

Hier is gemakshalve gekozen voor parameterwaarden die slechts een indikatie zijn van de grootte van de fysisch meest zinvolle waarden.

$$\begin{array}{ll} m_1 = 5 & \text{kg} \\ m_2 = 0.1 & \text{kg} \\ m_3 = 0.05 & \text{kg} \\ m_4 = 0.05 & \text{kg} \end{array} \qquad \begin{array}{ll} i_3 = & 10^{-4} \text{ kg m}^2 \\ i_4 = & 10^{-5} \text{ kg m}^2 \\ i_m = 2.25 & 10^{-5} \text{ kg m}^2 \end{array}$$

$$\begin{array}{ll} k_2 = 5 \cdot 10^3 & \text{N m}^{-1} \\ k_3 = & 10^6 \text{ N m}^{-1} \\ k_4 = 2 \cdot 10^5 & \text{N m}^{-1} \\ k^* = & 10^5 \text{ N m}^{-1} \\ EA = 5 \cdot 10^4 & \text{N} \end{array}$$

$$\begin{array}{ll} d = 0 & \text{m} \\ s = 0 & \text{m} \\ l_1 = 1.35 & \text{m} \\ l_2 = 0.04 & \text{m} \\ v = 0.02 & \text{m} \\ A_r = 0.1 & \text{m} \end{array}$$

$$i = 4.625 \quad (\text{zie vergelijking (1)})$$

De parameter  $F_w$  wordt per simulatie gewijzigd.

## Hoofdstuk 2

### Simulatie

#### § 2.1 Het eigen programma

##### Differentiaalvergelijkingen:

Uitgaande van de bewegingsvergelijking (2) is een stelsel differentiaalvergelijkingen opgesteld met de volgende vorm:

$$\begin{bmatrix} \dot{\underline{q}} \\ \underline{\ddot{q}} \end{bmatrix} = \begin{bmatrix} \dot{\underline{q}} \\ [M]^{-1} Q(\underline{q}, \dot{\underline{q}}, t) \end{bmatrix} \quad (3)$$

Definieer  $\underline{y} = \begin{bmatrix} \underline{q} \\ \dot{\underline{q}} \end{bmatrix}$ ; Dan kan het stelsel (3) geschreven worden als

$$\dot{\underline{y}} = \underline{\varphi}(t, \underline{y}) \quad (4)$$

Als er beginvoorwaarden  $\underline{y}(t=0)$  gegeven zijn en  $\underline{\varphi}(t, \underline{y})$  is continu in  $\underline{y}$ , dan kan men een numerieke benadering voor de oplossing van (4) maken met behulp van routines uit de NAG-library.

Het wrijvingsmodel:

In het model werkt een Coulombse wrijvingskracht op de slede. Als  $\dot{x}_1 > 0$  dan heeft deze kracht de waarde  $F_w$ . Als  $\dot{x}_1 < 0$  dan heeft deze kracht de waarde  $-F_w$ .

$\varphi(t, \underline{y})$  is dus niet continu in  $\underline{y}$  als  $F_w > 0$ , want  $\dot{x}_1$  is een element van  $\underline{y}$ .

Voor de wijze waarop dit probleem omzeild is zij verwezen naar appendix B. Hier is de sourcetekst van het eigen programma in opgenomen.

Tolerantie:

Het verschil tussen numerieke benadering en exakte oplossing van (4) met beginwaarde  $\underline{y}(t=0)$  is kleiner dan  $10^{-4}$  t.a.v. de 1<sup>e</sup> komponent van  $\underline{y}$ . De verplaatsing van de slede (in het model) wordt dus op 1/10 mm. nauwkeurig berekend.

§ 2.2 Verwerking met DADS

Invoer van de elementen:

Voor verwerking met DADS wordt er met behulp van de DADS-preprocessor een inputfile opgesteld. Hierin wordt o.a. opgegeven welke starre lichamen er in het model opgenomen zijn, een aantal constraints en de beginwaarden van de coördinaten die het systeem beschrijven.

DADS kent een aantal standaard 'joints' waaronder ook een riemaandrijving. Deze riem is echter rekloos en dus niet bruikbaar voor dit model.

Er zijn in dit geval niet veel mogelijkheden om gebruik te maken van de standaard-elementen van DADS.

Userforces:

De krachten die de riem uitoefent op de verschillende starre lichamen kunnen niet met behulp van standaard-elementen in rekening gebracht worden. In dit geval worden deze krachten berekend en opgeteld bij een array dat de globale gegeneraliseerde krachten bevat. Dit gebeurt middels wijzigingen in enkele hulproutines van DADS; 'INV32\_2D.FOR' en 'FRC32\_2D.FOR'. Dat deze routines gewijzigd zijn wordt aangegeven in de zgn. geformateerde invoerfile van DADS.

Het wrijvingsmodel:

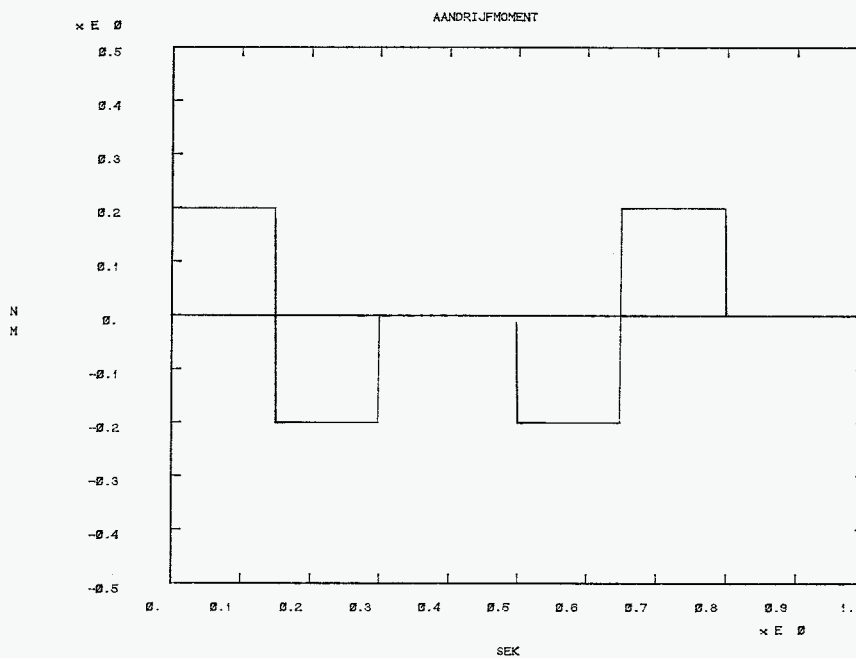
Om het kontinu zijn van  $\varphi(t, \underline{y})$  in  $\underline{y}$  te garanderen is hier gekozen voor een afwijkend wrijvingsmodel:

$$\begin{aligned} F_1(\underline{q}, \dot{\underline{q}}, t) &= F_w && \text{als } \dot{q}_1 \geq 10^{-4} \text{ m sec}^{-1} \\ & -F_w && \text{als } \dot{q}_1 \leq -10^{-4} \text{ m sec}^{-1} \\ & 10^4 \dot{q}_1 F_w && \text{als } |\dot{q}_1| < 10^{-4} \text{ m sec}^{-1} \end{aligned}$$

§ 2.3 Resultaten

Algemeen:

Voor de bewegingssimulatie is een ingangssignaal gekozen dat globaal overeenkomt met een maximale belasting van de motor. Bovendien wordt de aandrijving afwisselend in beide richtingen belast.

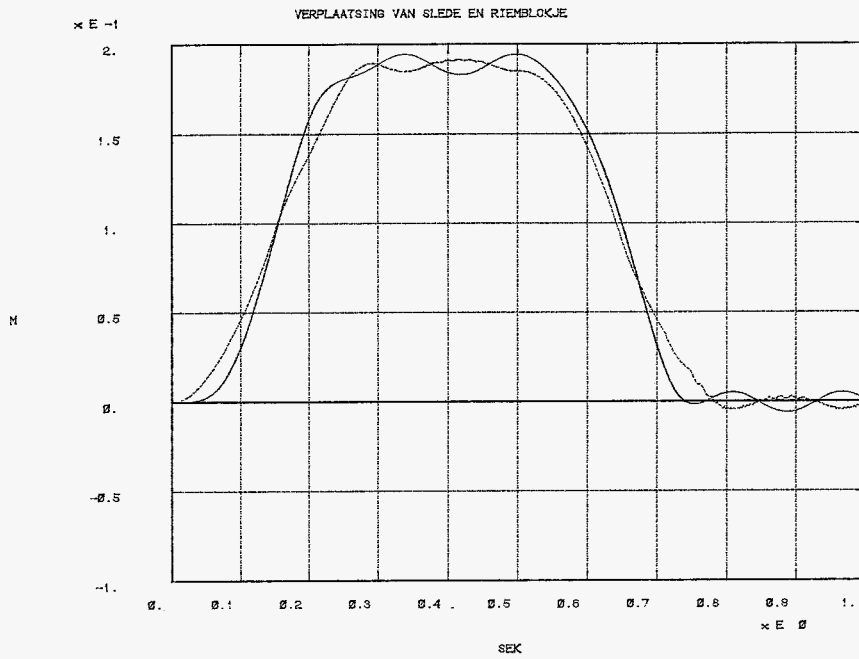


Als begintoestand is eenvoudig gekozen voor de referentietoestand (zie figuur 2). Dit houdt in:

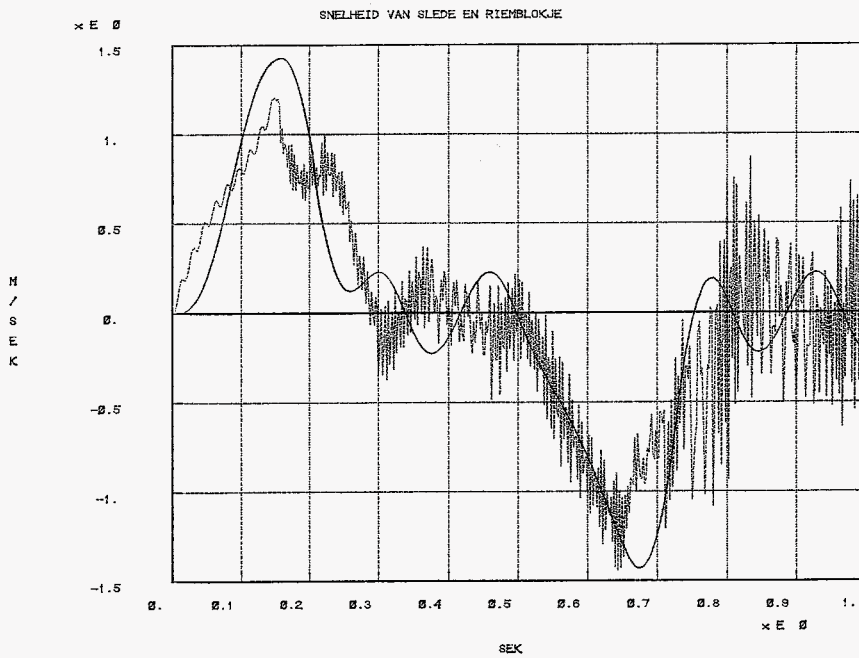
$$\underline{y}(t=0) = \underline{0} \quad (5)$$



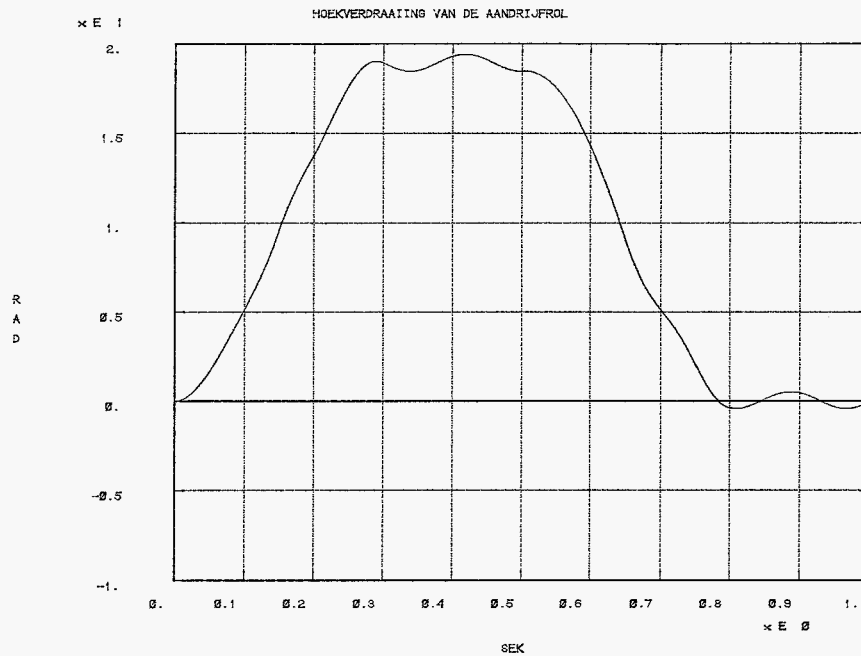
Simulatie met  $F_w = 0$



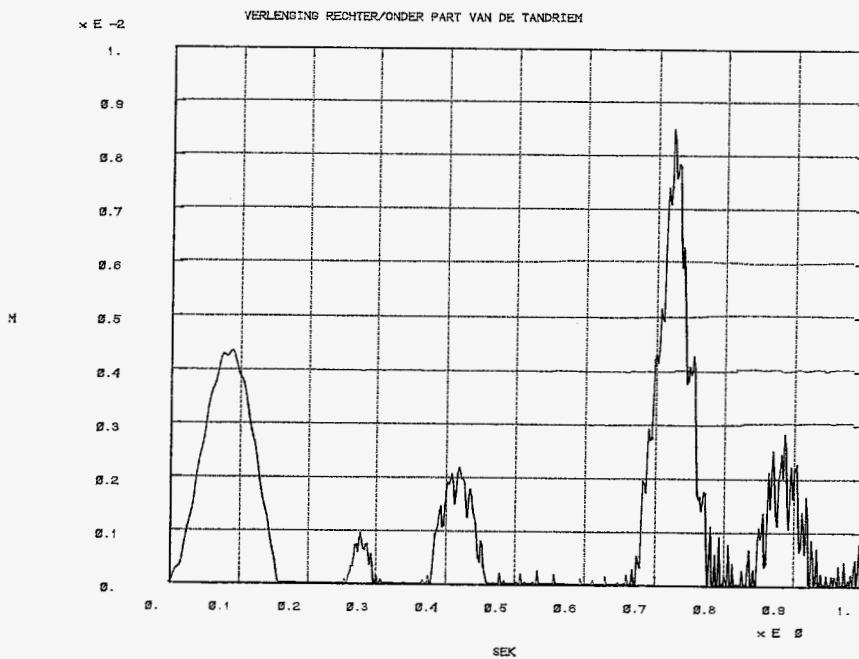
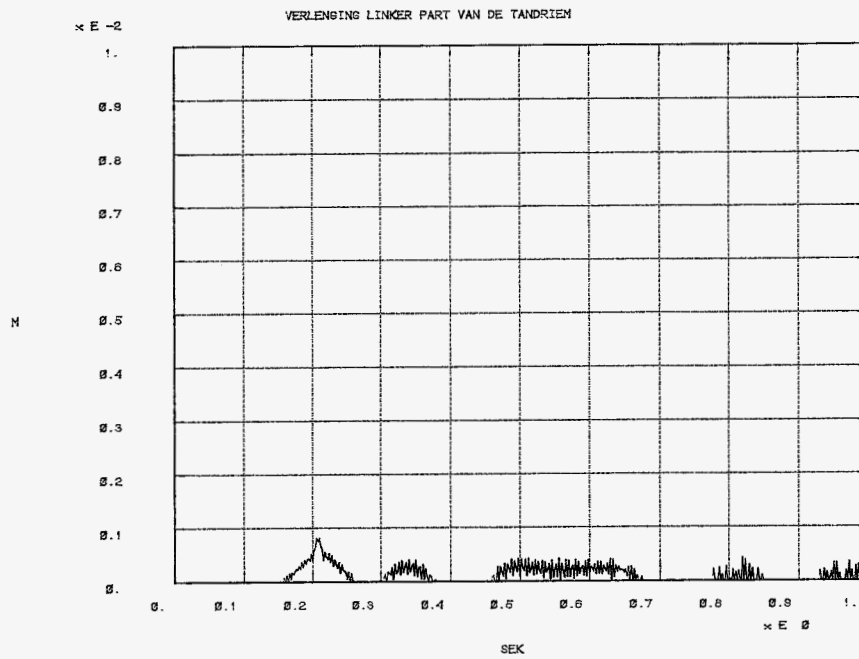
De verplaatsing van de slede vertoont een voorspelbaar verloop; na 0,8 sec. is deze weer in de beginpositie terug. De slede en de aandrijving blijven oscilleren om deze positie.



De snelheid van het riemblokje vertoont een snel oscillerend gedrag. Dit is te wijten aan de vrij stijve riem en de relatief geringe massa van het blokje. Hoewel minder zichtbaar geldt dit natuurlijk ook voor de verplaatsing.



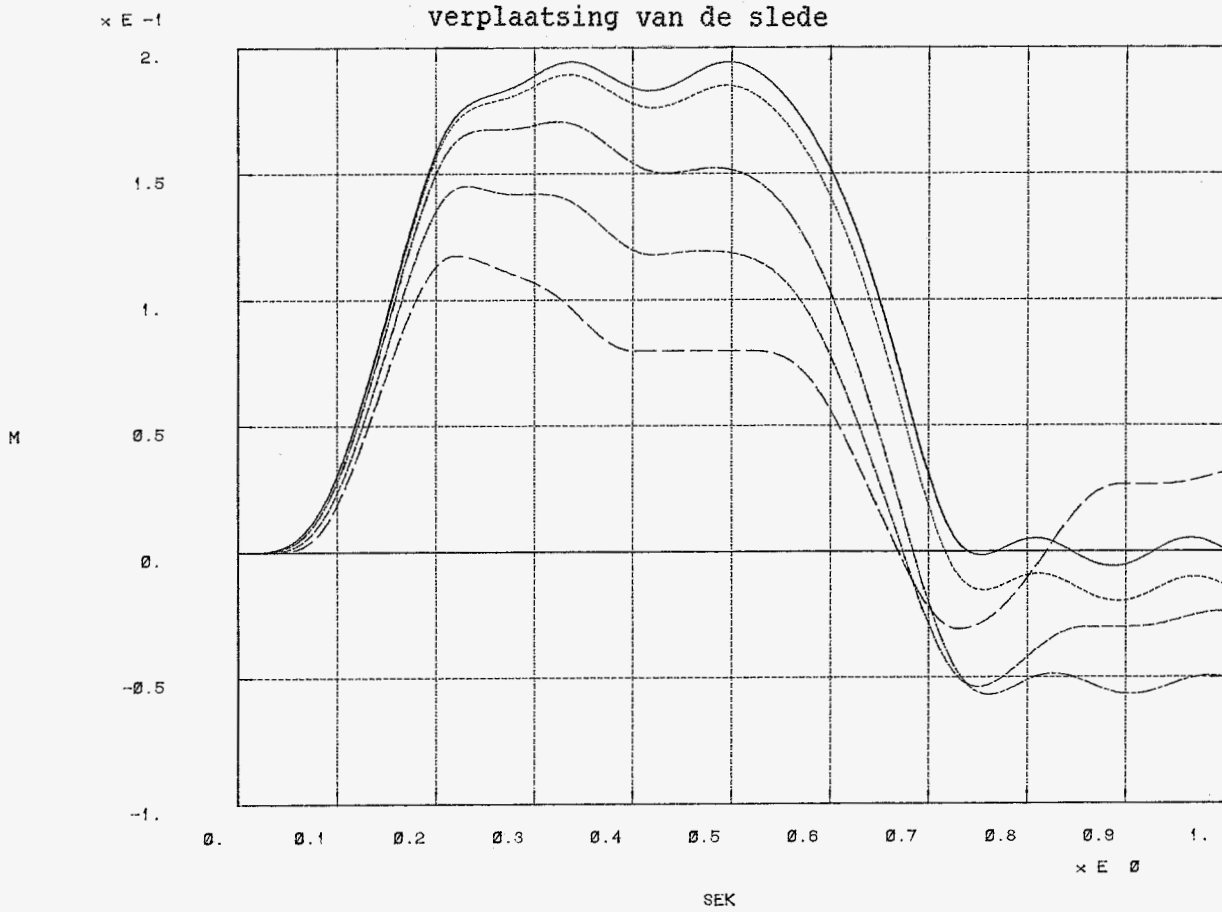
De hoekverdraaiing van de aandrijfrol lijkt evenals de verplaatsing van de slede een glad verloop te hebben. De traagheidsterm  $i_3 + i_m^2$  is van dezelfde orde grootte als  $m_1$ .



De verlenging van het lange riempart is vanzelfsprekend veel groter dan die van het korte part.

In de volgende figuur zijn de responsies op het eerder genoemde ingangssignaal te zien bij verschillende waarden van de grootte van de Coulombse wrijvingskracht.

Deze waarden zijn achtereenvolgens 0, 1, 5, 15 en 30 N.



Nauwkeurigheid:

De numerieke benadering van de oplossing van (4) bevat een fout die kleiner is dan  $10^{-4}$  m. t.a.v. de verplaatsing van de slede  $x_1$ . Dit wordt bereikt door de variabele TOL (zie appendix B, hoofdprogramma) de waarde  $10^{-7}$  te geven en de simulatie in 500 opeenvolgende deelintervallen uit te voeren. De waarde van TOL is een maat voor de lokale fout.

Dat deze nauwkeurigheid wordt bereikt met de genoemde waarde van TOL is vastgesteld na het doorrekenen van het probleem met steeds kleinere waarden voor TOL. Aangenomen is dat de benadering dan naar de exakte oplossing convergeert.

De onnauwkeurigheid van de andere coördinaten is echter groter dan  $10^{-4}$  m. Dit geldt met name voor  $x_2$  en  $(r\phi_4)$  !

Vergelijking tussen de uitvoering met respektievelijk het eigen programma en het programmapakket DADS:

Het maken van het eigen programma kost meer tijd dan het invoeren van het model in DADS. In het 1<sup>e</sup> geval moeten immers de bewegingsvergelijkingen omgewerkt worden tot een stelsel 1<sup>e</sup> orde differentiaalvergelijkingen. Het opstellen van de bewegingsvergelijkingen en het afleiden van het stelsel differentiaalvergelijkingen is in dit geval nog relatief eenvoudig; de massamatrix is onafhankelijk van de coördinaten en heeft bovendien een zuivere diagonaalvorm.

Vervolgens moeten er geschikte routines gekozen en getest worden om het probleem op te lossen. Ten slotte moet er een goed leesbare uitvoerfile gemaakt worden.

Bij verwerking met DADS speelt dit allemaal geen rol. Dit spaart tijd, mede omdat de kans op fouten veel geringer is.

Hier tegenover staat dat in het eigen programma de differentiaalvergelijkingen worden opgelost met routines die specifiek voor dit probleem gekozen zijn. Bij DADS is er geen keuze mogelijk!

Dit uit zich dan ook in het verschil in rekentijd; het eigen programma vraagt ca. 25 min. terwijl een simulatie met DADS het 4-voudige kost. (De berekeningen zijn uitgevoerd met de VAX-computer)

Ook neemt DADS een veelvoud van de door het eigen programma benodigde geheugenruimte in beslag.

Ondanks de verschillende wrijvingsmodellen die bij het eigen programma resp. bij simulatie met DADS zijn gehanteerd leveren beide methoden toch binnen de genoemde tolerantie een overeenkomstig resultaat!

Een voordeel van het werken met DADS is dat het model gemakkelijker over te dragen is; de methode van invoeren van een model staat beschreven in de handleiding van het programmapakket. Het uitbreiden van het model of het koppelen aan een ander model van een dynamisch systeem is relatief eenvoudig.

Alleen het eventueel reduceren van het aantal vrijheidsgraden van het model lijkt problemen op te leveren.

### Hoofdstuk 3

#### Evaluatie van modellering en simulatie

##### § 3.1 Fysische relevantie

Uit de plotjes van de simulatie blijkt dat in het model een snel fluctuerende beweging optreedt. Let bijvoorbeeld op de snelheid van het riemblokje.

Bij het verkleinen van de fouttolerantie convergeert de benadering van de snelheid van het riemblokje veel langzamer naar de exacte oplossing dan de benadering van de snelheid van de slede. Dit geldt ook voor de coördinaten  $x_3$ ,  $x_4$  en  $\varphi_4$ . Het vermoeden rijst dan ook dat de snelle fluctuaties in deze coördinaten fysisch van weinig betekenis zijn, althans voor de verplaatsing van de slede.

Bij simulaties met verschillende waarden voor de wrijvingskracht kunnen de respektievelijke bewegingen van de slede sterk verschillen. Bedenk dat de verhouding wrijvingskracht/aandrijfkracht op de slede in de orde van 1/10 is.

De aanname dat de wrijvingskracht zuiver Coulombs en niet afhankelijk van  $t$  en  $x_1$  is lijkt bovendien niet erg reëel. Het is moeilijk om een model op te stellen dat wel goed overeenkomt met de realiteit omdat een zinnige beschrijving  $F_w = F_w(q, \dot{q}, t)$  niet te verwezenlijken is.

§ 3.2 Aanbeveling voor vervolgonderzoek

Om de benodigde rekentijd voor een simulatie te verminderen is het aan te bevelen het aantal onafhankelijke coördinaten in het model te reduceren. Bij het afleiden van de bewegingsvergelijkingen is gebruik gemaakt van de vergelijking:

$$\frac{d}{dt} \left[ \frac{\partial T(\underline{q}, \underline{\dot{q}})}{\partial \underline{\dot{q}}} \right] - \left[ \frac{\partial T(\underline{q}, \underline{\dot{q}})}{\partial \underline{q}} \right] = \underline{Q}^T(\underline{q}, \underline{\dot{q}}, t)$$

Het is interessant te onderzoeken of de volgende werkwijze mogelijk en zinvol is:

Definieer de kolommen:

$$\underline{q}_1^T = (q_1, q_2) \quad \text{onafhankelijke coördinaten}$$

$$\underline{q}_2^T = (q_3, q_4, q_5, q_6) \quad \text{afhankelijke coördinaten}$$

$$(\underline{q}^0)^T = (\underline{q}_1^T, \underline{q}_2^T)$$



Beschrijf de volgende functies:

$$\underline{q}_2 = \underline{q}_2(\underline{q}_1)$$

$$\underline{q}^0 = \underline{q}^0(\underline{q}_1)$$

$$T^0 = T^0(\underline{q}_1, \dot{\underline{q}}_1) \quad \text{totale kinetische energie}$$

$$\underline{Q}^0 = \underline{Q}^0(\underline{q}_1, \dot{\underline{q}}_1, t) \quad \text{gegeneraliseerde krachtenvektor}$$

Maak nu gebruik van de volgende vergelijking:

$$\frac{d}{dt} \left[ \frac{\partial T^0(\underline{q}_1, \dot{\underline{q}}_1)}{\partial \dot{\underline{q}}_1} \right] - \left[ \frac{\partial T^0(\underline{q}_1, \dot{\underline{q}}_1)}{\partial \underline{q}_1} \right] = (\underline{Q}^0)^T(\underline{q}_1, \dot{\underline{q}}_1, t) \frac{\partial \underline{q}^0}{\partial \underline{q}_1}$$

Hieruit volgt dan een bewegingsvergelijking met de vorm:

$$[M(\underline{q}_1)] \ddot{\underline{q}}_1 + \underline{h}(\underline{q}_1, \dot{\underline{q}}_1) + \underline{H}(\underline{q}_1) M(t) = \underline{Q}$$

$[M(\underline{q}_1)]$  is een (2\*2) koördinaatafhankelijke massamatrix.

$\underline{h}(\underline{q}_1, \dot{\underline{q}}_1)$  is een (2\*1) vektor.

$\underline{H}(\underline{q}_1)$  is een (2\*1) ingangsmatrix.

Als impliciete beschrijving voor  $\underline{q}_2 = \underline{q}_2(\underline{q}_1)$  kan men bijvoorbeeld het volgende stelsel vergelijkingen kiezen (zie ook fig 5):

$$\begin{aligned} - F_2 + F_5 - F_6 &= 0 \\ F_3 + F_6 + F_7 &= 0 \\ F_4 - F_5 - F_7 &= 0 \\ - F_5 + F_7 &= 0 \end{aligned}$$

Voor de vergelijkingen voor de krachten  $F_i$  wordt verwezen naar appendix A.

Er kunnen 2 uitspraken gedaan worden:

- 1) Als  $F_6 > 0$  en  $F_7 > 0$ , dan is het stelsel lineair in  $\underline{q}_2$ .
- 2) Als  $F_7 = 0$ , dan is het stelsel singulier.

Om de lineariteit van het stelsel in  $\underline{q}_2$  en het regulier zijn te garanderen is het noodzakelijk de parameter  $d$  voldoende groot te kiezen, voor zo ver er al een keuze gemaakt kan worden.

Wegens de beperkte tijdsduur van de stage is dit probleem niet meer in beschouwing genomen.

APPENDICES

APPENDIX A

Afleiding van de bewegingsvergelijkingen

$$T = \frac{1}{2} \{ m_1(\dot{x}_1)^2 + m_2(\dot{x}_2)^2 + m_3(\dot{x}_3)^2 + i_3(\dot{\psi}_3)^2 + i_m(\dot{\psi}_m)^2 + \\ + m_4(\dot{x}_4)^2 + i_4(\dot{\psi}_4)^2 \}$$

Met vergelijking (1) volgt:

$$T = \frac{1}{2} \{ m_1(\dot{x}_1)^2 + m_2(\dot{x}_2)^2 + m_3(\dot{x}_3)^2 + i_3(\dot{\psi}_3)^2 + i_m(i\dot{\psi}_3)^2 + \\ + m_4(\dot{x}_4)^2 + i_4(\dot{\psi}_4)^2 \} = \\ = \frac{1}{2} \{ m_1(\dot{q}_1)^2 + m_2(\dot{q}_2)^2 + m_3(\dot{q}_3)^2 + (i_3 + i^2 i_m)(\dot{q}_4)^2 + \\ + m_4(\dot{q}_5)^2 + i_4(\dot{q}_6)^2 \}$$

$$\underline{F} = \begin{bmatrix} -F_1 + F_2 \\ -F_2 + F_5 - F_6 \\ F_3 + F_6 + F_7 \\ (F_6 - F_7) r \\ F_4 - F_5 - F_7 \\ (-F_5 + F_7) r \\ M(t) \end{bmatrix}$$

$$\underline{x}^T = ( \underline{q}^T, \varphi_m )$$

Er geldt:

$$\frac{d}{dt} \left[ \frac{\partial T(\underline{q}, \dot{\underline{q}})}{\partial \dot{\underline{q}}} \right] - \left[ \frac{\partial T(\underline{q}, \dot{\underline{q}})}{\partial \underline{q}} \right] = \underline{F}^T \frac{\partial \underline{x}}{\partial \underline{q}}$$

Dit is een stelsel van 6 gekoppelde 2<sup>e</sup> orde differentiaalvergelijkingen:

$$\begin{aligned} m_1 \ddot{q}_1 &= -F_1 + F_2 \\ m_2 \ddot{q}_2 &= -F_2 + F_5 - F_6 \\ m_3 \ddot{q}_3 &= F_3 + F_6 + F_7 \\ (i_3 + i^2 i_m) \ddot{q}_4 &= (F_6 - F_7) r + i M(t) \\ m_4 \ddot{q}_5 &= F_4 - F_5 - F_7 \\ i_4 \ddot{q}_6 &= (-F_5 + F_7) r \end{aligned}$$

Het stelsel is te schrijven als:

$$[M] \ddot{\underline{q}} = \underline{Q}(\underline{q}, \dot{\underline{q}}, t) \quad (2)$$

Vergelijkingen voor  $F_1$  t/m  $F_7$

$$\begin{aligned} F_1 = & F_W && \text{als } \{ ( \dot{q}_1 > 0 ) \text{ of } ( \dot{q}_1 = 0 \text{ en } F_2 \geq F_W ) \} \\ & -F_W && \text{als } \{ ( \dot{q}_1 < 0 ) \text{ of } ( \dot{q}_1 = 0 \text{ en } F_2 \leq -F_W ) \} \\ & F_2 && \text{anders} \end{aligned}$$

Voor  $F_2$  wordt verwezen naar figuur 6 en appendix B, Subroutine FORCES.

$$F_3 = -k_3 q_3$$

$$F_4 = k_4 ( d - q_5 )$$

$$F_5 = \text{MAX} ( 0 , \frac{q_5 + q_6 r - q_2}{l_1 - ( A_r + l_2 + q_2 )} EA )$$

$$F_6 = \text{MAX} ( 0 , \frac{q_2 - ( q_3 + q_4 r )}{A_r + q_2} EA )$$

$$F_7 = \text{MAX} ( 0 , \frac{( q_5 - q_6 r ) - ( q_3 - q_4 r )}{l_1} EA )$$

APPENDIX B

Eigen programma:

HOOFDPROGRAMMA:

```
PROGRAM RESPONSIE
C  RESPONSIE VAN EEN TANDRIEMAANDRIJVING.

C  KONSTANTEN:
    INCLUDE 'CONST.DAT'

C  COMMON BLOCKS:
    COMMON /CASE/      MOVE

C  INVOER:
    REAL*8  TOL, TEIND
    INTEGER NSTEPS,    ! AANTAL PLOTPUNTEN
&          NREPRT    ! AANTAL RAPPORTERINGEN (ONGEVEER)

C  UITVOER:
    REAL*8  OUTPUT(NSTMAX+1,NJOUT)
C          OUTPUT = ( NVERG TOESTANDSGR. , SIGNAAL , TIJD )

C  VARIABELEN:
    REAL*8  TSTAP, TIJD,
&          HMAX,    ! =< HNAG (CONSTANT)
&          THMAX,  ! TIJD+HMAX
&          WORK(NVERG,IWORK), Y(NVERG)
    INTEGER FAIL,
&          COUNT  ! NR VAN DE TE BEREKENEN STAP
    CHARACTER*3 MOVE

C  SUBROUTINES:
    EXTERNAL FUNAFG, ! BEREKENT DE TOESTANDS-TIJDAFGELEIDE
&          SIGNAL,  ! STUURSIGNAAL (M) ALS FUN. VAN TIJD
&          OUT,     ! MAAKT EEN LEESBARE UITVOERFILE
&          PARAMREAD, ! LEEST DE PARAMETERWAARDEN
&          PLOTTEN, ! MAAKT PLAATJES
&          UPDATE,  ! OUTPUT UPDATEN EN PRINTEN OP SCHERM
&          CURVES   ! CURVEFILES MAKEN

C  FUNCTIONS:
    EXTERNAL G,      ! G = ABS(F2) - FW
&          F2       ! ZIE SUBROUTINE 'FORCES'
```

```
CALL PARAMREAD      ! LEEST DE PARAMETERWAARDEN

C  BEGIN INVOER VAN SCHERM
    WRITE(6, 10001) NSTMAX  ! TIJD, AANTAL STAPPEN, TOLERANTIE
    READ(5, *) TEIND, NSTEPS, NREPRT, TOL
C  EINDE

C  BEGIN INITIALISEREN;
    HMAX = DMIN1( HNAG, TEIND/NSTEPS/2 )
    TIJD = 0
    TSTAP = 0
    COUNT = 0
    FAIL = 4
    DO 100 J = 1, NVERG
        Y(J) = 0
100  CONTINUE
    MOVE = 'NOT'
C  EINDE

C  BEGIN RESPONSIE;
200  CONTINUE
    IF ( FAIL .EQ. 0 ) THEN ! SNELHEID = 0
C  BEGIN MOVE BEPALEN
        IF ( G(T,Y) .GE. 0 ) THEN ! VERSNELLING <> 0
            IF ( F2(T,Y) .GT. 0 ) THEN
                MOVE = 'POS'
            ELSE
                MOVE = 'NEG'
            END IF
        ELSE ! VERSNELLING = 0
            MOVE = 'NOT'
        END IF
C  EINDE MOVE BEPALEN
C  BEGIN VOORBIJ ' SNELHEID = 0 ' INTEGREREN
    FAIL = 1
    TOL = ABS(TOL)
    THMAX = DMIN1( TIJD+HMAX, TSTAP )
    IF ( DVSTIJF ) THEN
        CALL DO2EAF(TIJD, THMAX, NVERG, Y, TOL, FUNAFG,
&                                WORK, IWORK, FAIL )
    ELSE
        CALL DO2CAF(TIJD, THMAX, NVERG, Y, TOL, FUNAFG,
&                                WORK, FAIL )
    END IF
    IF ( FAIL .EQ. 0 ) THEN
        FAIL = 4
    END IF
C  EINDE VOORBIJ ' SNELHEID = 0 ' INTEGREREN
END IF
```



```
IF ( FAIL .EQ. 4 ) THEN
  CALL UPDATE(TSTAP,MOVE, Y, OUTPUT,
&          COUNT, TEIND, NSTEPS )
ELSE IF ( FAIL .NE. 0 ) THEN
  WRITE(6, 10002) FAIL
  STOP ! NOODSTOP!!!!!!
ENDIF
IF ( COUNT .GT. NSTEPS ) GO TO 300
  FAIL = 1
  TOL = ABS( TOL )
  IF ( MOVE .EQ. 'NOT' .AND. G(T,Y) .LT. 0 ) THEN
    IF ( DVSTIJF ) THEN
      CALL D02EHF(TIJD, TSTAP, NVERG, Y, TOL, IRELAB,
&          HMAX, FUNAFG, MPED, PEDERV, G, WORK,
&          IWORK, FAIL )
    ELSE
      CALL D02CHF(TIJD, TSTAP, NVERG, Y, TOL, IRELAB,
&          HMAX, FUNAFG, G, WORK, FAIL )
    END IF
  ELSE
    IF ( Y(NXD1) .GT. 0 ) THEN
      MOVE = 'POS'
    ELSE
      MOVE = 'NEG'
    END IF
    IF ( DVSTIJF ) THEN
      CALL D02EGF(TIJD, TSTAP, NVERG, Y, TOL,
&          HMAX, NXD1, 0, FUNAFG, WORK, IWORK, FAIL )
    ELSE
      CALL D02CGF(TIJD, TSTAP, NVERG, Y, TOL, HMAX,
&          NXD1, 0, FUNAFG, WORK, FAIL )
    END IF
  END IF
  GO TO 200
300 CONTINUE
C   EINDE RESPONSIE
```

```
CALL CURVES( OUTPUT, NSTEPS )      ! CURVEFILES MAKEN
CALL OUT(OUTPUT, NSTEPS, NREPRT, TOL) ! UITVOERFILE MAKEN
CALL PLOTTEN( NSTEPS, OUTPUT )     ! KWALITATIEVE PLOTJES
```

```
C   BEGIN FORMATS:
```

```
10001 FORMAT('$Dit programma maakt een responsieberekening
& van een model',/, '$van een tandriemaandrijving van een
& XY-besturing',/,
&'$Geef Eindtijd, Aantal plotpunten ( < ',I4,' ),
& Aantal rapporteringen',/, '$en de Tolerantie
& van de simulatie',/)
10002 FORMAT('$Het programma is voortijdig beëindigd; FAIL = ',I1,/)
C   EINDE
```

```
CALL EXIT
END
```

ENKELE SUBROUTINES:

```
      REAL*8 FUNCTION G( T, Y )
C     G = ABS(F2) - FW

C     COMMON BLOCKS:
      INCLUDE 'PARAM.COMMON'

C     INVOER:
      REAL*8  Y(12), T

C     FUNCTIONS:
      EXTERNAL F2

      G = ABS( F2(T,Y) ) - FW

      RETURN
      END

      SUBROUTINE FORCES(Y, T, F)
C     BEREKENT DE KRACHTEN ALS FUNKTIE VAN TOESTAND EN T

C     KONSTANTEN
      INCLUDE 'CONST.DAT'

C     COMMON BLOCKS:
      INCLUDE 'PARAM.COMMON'
      COMMON /CASE/ MOVE
      CHARACTER*3 MOVE

C     INVOER:
      REAL*8  Y(NVERG), T

C     UITVOER:
      REAL*8  F(NFORC)

C     VARIABELEN:
      REAL*8  HULP, FPOS, FNEG

C     BETEKENIS VAN DE TOESTANDSVEKTOR Y:
C
C     1   2   3   4   5   6   7   8   9   10  11  12
C     X1  X2  X3  Fi3  X4  Fi4  X1D  X2D  X3D  Fi3D  X4D  Fi4D
```

```
C      BEGIN F2;
      HULP = Y(2) - Y(1)

      IF ( V .LT. HULP ) THEN
        FPOS = KSTER * ( HULP - V ) + K2 * V
      ELSE IF ( 0 .LT. HULP ) THEN
        FPOS = K2 * HULP
      ELSE
        FPOS = 0
      ENDIF

      IF ( HULP .LT. -(V+S) ) THEN
        FNEG = KSTER * ( HULP + V+S ) - K2 * V
      ELSE IF ( HULP .LT. -S ) THEN
        FNEG = K2 * ( HULP + S )
      ELSE
        FNEG = 0
      ENDIF

      F(2) = FPOS + FNEG
C      EINDE

C      BEGIN F1
      IF ( MOVE .EQ. 'POS' ) THEN
        F(1) = FW
      ELSE IF ( MOVE .EQ. 'NEG' ) THEN
        F(1) = -FW
      ELSE
        F(1) = F(2)
      END IF
C      EINDE

C      BEGIN F3 EN F4
      F(3) = -K3 * Y(3)
      F(4) = K4 * ( D - Y(5) )
C      EINDE

C      BEGIN F5
      HULP = Y(5) + Y(6) * R - Y(2)
      IF ( HULP .GT. 0 ) THEN
        F(5) = HULP / ( L1 - ( AR + L2 + Y(2) ) ) * EA
      ELSE
        F(5) = 0
      END IF
C      EINDE
```

```
C      BEGIN F6
      HULP = Y(2) - ( Y(3) + Y(4) * R )
      IF ( HULP .GT. 0 ) THEN
        F(6) = HULP / ( AR + Y(2) ) * EA
      ELSE
        F(6) = 0
      END IF
C      EINDE

C      BEGIN F7
      HULP = ( Y(5) - Y(6) * R ) - ( Y(3) - Y(4) * R )
      IF ( HULP .GT. 0 ) THEN
        F(7) = HULP / L1 * EA
      ELSE
        F(7) = 0
      END IF
C      EINDE

      RETURN

END
```

```
SUBROUTINE FUNAFG( T, Y, AFG)
C   BEREKENING VAN DE TIJDSAFGELIJDE VAN DE VEKTOR Y

C   KONSTANTEN
      INCLUDE 'CONST.DAT'

C   PARAMETERNAMEN
      INCLUDE 'PARAM.COMMON'

C   INVOER:
      REAL*8  T, Y(NVERG)

C   UITVOER:
      REAL*8  AFG(NVERG)

C   SUBROUTINES:
      EXTERNAL FORCES, ! KRACHTEN ALS FUN. VAN TOESTAND EN T
&      SIGNAL ! STUURSIGNAAL (M) ALS FUNKTIE VAN T

C   VARIABELEN
      REAL*8 F(NFORC), M

C   BEGIN BEREKENING;
      CALL FORCES(Y, T, F)
      CALL SIGNAL(T,M)
      AFG(1) = Y(7)
      AFG(2) = Y(8)
      AFG(3) = Y(9)
      AFG(4) = Y(10)
      AFG(5) = Y(11)
      AFG(6) = Y(12)
      AFG(7) = ( - F(1) + F(2) ) / M1
      AFG(8) = ( - F(2) + F(5) - F(6) ) / M2
      AFG(9) = ( F(3) + F(6) + F(7) ) / M3
      AFG(10) = ( R*( F(6)-F(7) ) + M*IO ) / ( J3 + IO**2*JM )
      AFG(11) = ( F(4) - F(5) - F(7) ) / M4
      AFG(12) = R*( F(7) - F(5) ) / J4

C   EINDE

      RETURN

END
```

APPENDIX C

Verwerking met DADS

Invoerfile:

CREATE HEADER

RESPONSIE VAN EEN TANDRIEM-AANDRIJVING

.  
T.b.v. de modellering van een XY-tafel wordt de  
aandrijving van alleen een massa m.b.v. van 1 tandriem  
beschouwd.

.  
De riemrollen zijn flexibel opgesteld.  
De riem is flexibel.  
De massa is flexibel aan de riem bevestigd.

ANALYSIS

CREATE SYSTEM.DATA

UNITS	:= 'SI'
ANALYSIS.TYPE	:= 'DYNAMIC'
STARTING.TIME	:= '0.0'
ENDING.TIME	:= '1'
PRINT.INTERVAL	:= '0.01'
GRAVITY.SEA.LEVEL	:= '9.80665'
X.GRAVITY	:= '0.0'
Y.GRAVITY	:= '-1.0'
SCALE.GRAVITY.COEF	:= '0'
MATRIX.OPERATIONS	:= 'SPARSE'
REDUNDANCY.CHECK	:= 'TRUE'
LU.TOL	:= '1.0D-12'
ASSEMBLY.TOL	:= '1E-4'
BYPASS.ASSEMBLY	:= 'FALSE'
OUTPUT.FILE	:= 'BOTH'
REFERENCE.FRAME	:= 'LOCAL'
DEBUG.FLAG	:= 'TRUE'

```
CREATE DYNAMIC.DATA
  REACTION.FORCES           := 'FALSE'
  FORCE.COORDINATES         := 'GLOBAL'
  PRINT.METHOD            := 'INTERPOLATED'
  MAX.INT.STEP              := '1E-3'
  SOLUTION.TOL              := '1E-4'
  INTEGRATION.TOL          := '1E-6'
```

CONSTRAINTS

```
CREATE POSITION.CONSTRAINT
  NAME                       := 'POS.Y.M1'
  BODY.NAME                   := 'M1'
  TYPE.CONSTRAINT            := 'Y'
  CONSTRAINT.VALUE           := '0.0'
  P.ON.BODY                   := ( 0.0, 0.0 )
  Q.ON.BODY                   := ( 1.0, 0.0 )
  ANGULAR.UNITS              := 'DEGREES'
```

```
CREATE POSITION.CONSTRAINT
  NAME                       := 'POS.PHI.M1'
  BODY.NAME                   := 'M1'
  TYPE.CONSTRAINT            := 'PHI'
  CONSTRAINT.VALUE           := '0.0'
  P.ON.BODY                   := ( 0.0, 0.0 )
  Q.ON.BODY                   := ( 1.0, 0.0 )
  ANGULAR.UNITS              := 'RADIANS'
```

```
CREATE POSITION.CONSTRAINT
  NAME                       := 'POS.Y.M2'
  BODY.NAME                   := 'M2'
  TYPE.CONSTRAINT            := 'Y'
  CONSTRAINT.VALUE           := '0.0'
  P.ON.BODY                   := ( 0.0, 0.0 )
  Q.ON.BODY                   := ( 1.0, 0.0 )
  ANGULAR.UNITS              := 'DEGREES'
```

```
CREATE POSITION.CONSTRAINT
  NAME                       := 'POS.PHI.M2'
  BODY.NAME                   := 'M2'
  TYPE.CONSTRAINT            := 'PHI'
  CONSTRAINT.VALUE           := '0.0'
  P.ON.BODY                   := ( 0.0, 0.0 )
  Q.ON.BODY                   := ( 1.0, 0.0 )
  ANGULAR.UNITS              := 'RADIANS'
```



```
CREATE POSITION.CONSTRAINT
NAME                := 'POS.Y.M3'
BODY.NAME           := 'M3'
TYPE.CONSTRAINT     := 'Y'
CONSTRAINT.VALUE    := '0.0'
P.ON.BODY           := ( 0.0, 0.0 )
Q.ON.BODY           := ( 1.0, 0.0 )
ANGULAR.UNITS       := 'DEGREES'
```

```
CREATE POSITION.CONSTRAINT
NAME                := 'POS.Y.M4'
BODY.NAME           := 'M4'
TYPE.CONSTRAINT     := 'Y'
CONSTRAINT.VALUE    := '0.0'
P.ON.BODY           := ( 0.0, 0.0 )
Q.ON.BODY           := ( 1.0, 0.0 )
ANGULAR.UNITS       := 'DEGREES'
```

#### JOINTS

```
CREATE BELT.JOINT
NAME                := 'BELT.MOTORANKER'
GROUP.NAME          := 'RIEM'
BODY.NAME           := 'MOTORANKER'
ROT.DIRECTION       := 'NORMAL'
RADIUS              := '0.01'
```

```
CREATE BELT.JOINT
NAME                := 'BELT.M3'
GROUP.NAME          := 'RIEM'
BODY.NAME           := 'M3'
ROT.DIRECTION       := 'NORMAL'
RADIUS              := '0.04625'
```

```
CREATE REVOLUTE.JOINT
NAME                := 'REV.MOTORANKER'
BODY.1.NAME         := 'MOTORANKER'
BODY.2.NAME         := 'FRAME'
P.ON.BODY.1         := ( 0, 0 )
P.ON.BODY.2         := ( 0, -0.15 )
Q.ON.BODY.1         := ( 1, 0 )
Q.ON.BODY.2         := ( 1, -0.15 )
NODE.1              := '0'
NODE.2              := '0'
```

CREATE BODY

NAME	:= 'M1'
CENTER.OF.GRAVITY	:= ( 0.0, 0.0 )
PHI	:= '0.0'
FIXED.TO.GROUND	:= 'FALSE'
MASS	:= '5'
INERTIA	:= '1.0'
XG.FORCE	:= '0.0'
YG.FORCE	:= '0.0'
TORQUE.CONSTANT	:= '0.0'
CURVE.XGF	:= 'NONE'
CURVE.YGF	:= 'NONE'
CURVE.TORQUE	:= 'NONE'
OUTLINE.SHAPE	:= 'NONE'
SHAPE.CENTER	:= ( 0.0, 0.0 )
ANGULAR.UNITS	:= 'DEGREES'
FLEXIBLE	:= 'FALSE'
SUPERELEMENT	:= 'FALSE'

CREATE BODY

NAME	:= 'M2'
CENTER.OF.GRAVITY	:= ( 0.0, 0.0 )
PHI	:= '0.0'
FIXED.TO.GROUND	:= 'FALSE'
MASS	:= '1E-1'
INERTIA	:= '1.0'
XG.FORCE	:= '0.0'
YG.FORCE	:= '0.0'
TORQUE.CONSTANT	:= '0.0'
CURVE.XGF	:= 'NONE'
CURVE.YGF	:= 'NONE'
CURVE.TORQUE	:= 'NONE'
ANGULAR.UNITS	:= 'DEGREES'
FLEXIBLE	:= 'FALSE'
SUPERELEMENT	:= 'FALSE'

CREATE BODY

NAME	:= 'M3'
CENTER.OF.GRAVITY	:= ( 0.0, 0.0 )
PHI	:= '0.0'
FIXED.TO.GROUND	:= 'FALSE'
MASS	:= '5E-2'
INERTIA	:= '1E-4'
XG.FORCE	:= '0.0'
YG.FORCE	:= '0.0'
TORQUE.CONSTANT	:= '0.0'
CURVE.XGF	:= 'NONE'
CURVE.YGF	:= 'NONE'
CURVE.TORQUE	:= 'NONE'

```
OUTLINE.SHAPE           := 'NONE'  
SHAPE.CENTER            := ( 0.0, 0.0 )  
ANGULAR.UNITS           := 'RADIANS'  
FLEXIBLE                 := 'FALSE'  
SUPERELEMENT            := 'FALSE'
```

CREATE BODY

```
NAME                     := 'M4'  
CENTER.OF.GRAVITY       := ( 0.0, 0.0 )  
PHI                      := '0.0'  
FIXED.TO.GROUND         := 'FALSE'  
MASS                     := '5E-2'  
INERTIA                  := '1E-5'  
XG.FORCE                 := '0.0'  
YG.FORCE                 := '0.0'  
TORQUE.CONSTANT         := '0.0'  
CURVE.XGF                := 'NONE'  
CURVE.YGF                := 'NONE'  
CURVE.TORQUE            := 'NONE'  
ANGULAR.UNITS           := 'RADIANS'  
FLEXIBLE                 := 'FALSE'  
SUPERELEMENT            := 'FALSE'
```

CREATE BODY

```
NAME                     := 'FRAME'  
CENTER.OF.GRAVITY       := ( 0.0, 0.0 )  
PHI                      := '0.0'  
FIXED.TO.GROUND         := 'TRUE'  
MASS                     := '1.0'  
INERTIA                  := '1.0'  
XG.FORCE                 := '0.0'  
YG.FORCE                 := '0.0'  
TORQUE.CONSTANT         := '0.0'  
CURVE.XGF                := 'NONE'  
CURVE.YGF                := 'NONE'  
CURVE.TORQUE            := 'NONE'  
OUTLINE.SHAPE           := 'NONE'  
SHAPE.CENTER            := ( 0.0, 0.0 )  
ANGULAR.UNITS           := 'DEGREES'  
FLEXIBLE                 := 'FALSE'  
SUPERELEMENT            := 'FALSE'
```

CREATE BODY

```
NAME                     := 'MOTORANKER'  
CENTER.OF.GRAVITY       := ( 0.0, 0.0 )  
PHI                      := '0.0'  
FIXED.TO.GROUND         := 'FALSE'  
MASS                     := '1.0'  
INERTIA                  := '2.25E-5'
```

XG.FORCE := '0.0'  
YG.FORCE := '0.0'  
TORQUE.CONSTANT := '0.0'  
CURVE.XGF := 'NONE'  
CURVE.YGF := 'NONE'  
CURVE.TORQUE := 'NONE'  
ANGULAR.UNITS := 'RADIANS'  
FLEXIBLE := 'FALSE'  
SUPERELEMENT := 'FALSE'

CREATE INITIAL.CONDITION

NAME := 'INIT.M1.X'  
BODY.1.NAME := 'M1'  
BODY.2.NAME := 'NONE'  
TYPE.INITIAL.COND := 'X'  
INITIAL.VALUE := '0.0'  
TIME.DERIVATIVE := '0.0'  
P.ON.BODY.1 := ( 0.0, 0.0 )  
P.ON.BODY.2 := ( 0.0, 0.0 )  
EXTRA.COORD := '0'  
ANGULAR.UNITS := 'DEGREES'

CREATE INITIAL.CONDITION

NAME := 'INIT.M2.X'  
BODY.1.NAME := 'M2'  
BODY.2.NAME := 'NONE'  
TYPE.INITIAL.COND := 'X'  
INITIAL.VALUE := '0.0'  
TIME.DERIVATIVE := '0.0'  
P.ON.BODY.1 := ( 0.0, 0.0 )  
P.ON.BODY.2 := ( 0.0, 0.0 )  
EXTRA.COORD := '0'  
ANGULAR.UNITS := 'DEGREES'

CREATE INITIAL.CONDITION

NAME := 'INIT.M3.X'  
BODY.1.NAME := 'M3'  
BODY.2.NAME := 'NONE'  
TYPE.INITIAL.COND := 'X'  
INITIAL.VALUE := '0.0'  
TIME.DERIVATIVE := '0.0'  
P.ON.BODY.1 := ( 0.0, 0.0 )  
P.ON.BODY.2 := ( 0.0, 0.0 )  
EXTRA.COORD := '0'  
ANGULAR.UNITS := 'DEGREES'

CREATE INITIAL.CONDITION

```
NAME                := 'INIT.M3.PHI'  
BODY.1.NAME         := 'M3'  
BODY.2.NAME         := 'NONE'  
TYPE.INITIAL.COND  := 'PHI'  
INITIAL.VALUE       := '0.0'  
TIME.DERIVATIVE    := '0.0'  
P.ON.BODY.1        := ( 0.0, 0.0 )  
P.ON.BODY.2        := ( 0.0, 0.0 )  
EXTRA.COORD        := '0'  
ANGULAR.UNITS      := 'RADIANS'
```

CREATE INITIAL.CONDITION

```
NAME                := 'INIT.M4.X'  
BODY.1.NAME         := 'M4'  
BODY.2.NAME         := 'NONE'  
TYPE.INITIAL.COND  := 'X'  
INITIAL.VALUE       := '0.0'  
TIME.DERIVATIVE    := '0.0'  
P.ON.BODY.1        := ( 0.0, 0.0 )  
P.ON.BODY.2        := ( 0.0, 0.0 )  
EXTRA.COORD        := '0'  
ANGULAR.UNITS      := 'DEGREES'
```

CREATE INITIAL.CONDITION

```
NAME                := 'INIT.M4.PHI'  
BODY.1.NAME         := 'M4'  
BODY.2.NAME         := 'NONE'  
TYPE.INITIAL.COND  := 'PHI'  
INITIAL.VALUE       := '0.0'  
TIME.DERIVATIVE    := '0.0'  
P.ON.BODY.1        := ( 0.0, 0.0 )  
P.ON.BODY.2        := ( 0.0, 0.0 )  
EXTRA.COORD        := '0'  
ANGULAR.UNITS      := 'RADIANS'
```

Hulproutine INV32\_2D.FOR

C IN32: Reads and stores input data for user-defined forces (2D).

.  
.  
.

C---COMMON blocks-----

INCLUDE '[WWTMWV.XYNAG]PARAM.COMMON'

COMMON/BODYNUMMERS/ NM1, NM2, NM3, NM4, NMOT  
INTEGER NM1, NM2, NM3, NM4, NMOT

.  
.  
.

C---Functions and subroutines-----

.  
.  
.

C SUBROUTINE OM PARAMETERS TE LEZEN  
EXTERNAL PARAMREAD

.  
.  
.

C=====Process Block=====

.  
.  
.

C LEES DE PARAMETERWAARDEN  
CALL PARAMREAD

C BEGIN BODYNUMMERS BEPALEN  
NM1 = GETNUM(3, 'M1')  
NM2 = GETNUM(3, 'M2')  
NM3 = GETNUM(3, 'M3')  
NM4 = GETNUM(3, 'M4')  
NMOT = GETNUM(3, 'MOTORANKER')

C EINDE

.  
.  
.

.  
.  
.

RETURN  
END

C SUBROUTINE OM PARAMETERWAARDEN TE LEZEN  
INCLUDE '[WWTMWV.XYNAG]PARAMREAD.FOR'

END

Hulproutine FRC32\_2D.FOR

C FRC32: Force calculations for user-defined force elements (2D).

.  
.  
.

C---COMMON blocks-----

INCLUDE 'USER3:[WWTMWV.XYNAG]PARAM.COMMON'

COMMON/BODYNUMMERS/ NM1,NM2,NM3,NM4,NMOT  
INTEGER NM1, NM2, NM3, NM4, ! BODYNUMMERS M1 T/M M4  
& NMOT ! BODYNUMMER VAN MOTORANKER

C---Local variables-----

REAL\*8 HULP, FPOS, FNEG, ! HULPVARIABLEN  
& VDELTA, ! GRENSSNELHEID  
& MOMENT, ! AANDRIJFMOMENT  
& F1, F2, F3, F4, F5, F6, F7 ! OVERIGE USERFORCES

C VDELTA heeft een positieve waarde. Als de snelheid van  
C de slede in absolute zin groter is dan VDELTA wordt het  
C Coulomb-wrijvings-model gehanteerd.

C---Functions and subroutines-----

C SUBROUTINE VOOR HET BEREKENEN VAN HET AANDRIJFMOMENT:  
EXTERNAL SIGNAL

.  
.  
.



C=====Process Block=====

C---First zero out the error condition flag to indicate that no errors  
C have occurred yet.

ERRCOD = 0

C BEGIN BEREKENING KRACHTEN

C BEGIN F1

VDELTA = 1D-3

IF ( QD(1,NM1) .GT. VDELTA ) THEN

F1 = FW

ELSE IF ( QD(1,NM1) .LT. -VDELTA ) THEN

F1 = -FW

ELSE

F1 = QD(1,NM1)/VDELTA \* FW

END IF

C EINDE

C BEGIN F2;

HULP = Q(1,NM2) - Q(1,NM1)

IF ( V .LT. HULP ) THEN

FPOS = KSTER \* ( HULP - V ) + K2 \* V

ELSE IF ( 0 .LT. HULP ) THEN

FPOS = K2 \* HULP

ELSE

FPOS = 0

ENDIF

IF ( HULP .LT. -(V+S) ) THEN

FNEG = KSTER \* ( HULP + V+S ) - K2 \* V

ELSE IF ( HULP .LT. -S ) THEN

FNEG = K2 \* ( HULP + S )

ELSE

FNEG = 0

ENDIF

F2 = FPOS + FNEG

C EINDE

C BEGIN F3 EN F4

F3 = - K3 \* Q(1,NM3)

F4 = K4 \* ( D - Q(1,NM4) )

C EINDE

```
C      BEGIN F5
      HULP = Q(1,NM4) + Q(3,NM4) * R - Q(1,NM2)
      IF ( HULP .GT. 0 ) THEN
        F5 = HULP / ( L1 - ( AR + L2 + Q(1,NM2) ) ) * EA
      ELSE
        F5 = 0
      END IF
C      EINDE

C      BEGIN F6
      HULP = Q(1,NM2) - ( Q(1,NM3) + Q(3,NM3) * R )
      IF ( HULP .GT. 0 ) THEN
        F6 = HULP / ( AR + Q(1,NM2) ) * EA
      ELSE
        F6 = 0
      END IF
C      EINDE

C      BEGIN F7
      HULP = ( Q(1,NM4) - Q(3,NM4) * R )
&      - ( Q(1,NM3) - Q(3,NM3) * R )
      IF ( HULP .GT. 0 ) THEN
        F7 = HULP / L1 * EA
      ELSE
        F7 = 0
      END IF
C      EINDE
C      EINDE BEREKENING KRACHTEN

C      BEGIN AANDRIJFMOMENT
      CALL SIGNAL(TIME,MOMENT)
C      EINDE

C      BEGIN AANPASSEN KRACHTENSOMMEN
      FRC(3,NMOT) = FRC(3,NMOT) + MOMENT
      FRC(1,NM1) = FRC(1,NM1) - F1 + F2
      FRC(1,NM2) = FRC(1,NM2) - F2 + F5 - F6
      FRC(1,NM3) = FRC(1,NM3) + F3 + F6 + F7
      FRC(3,NM3) = FRC(3,NM3) + ( F6 - F7 ) * R
      FRC(1,NM4) = FRC(1,NM4) + F4 - F5 - F7
      FRC(3,NM4) = FRC(3,NM4) + ( -F5 + F7 ) * R
C      EINDE

      RETURN

      END
```

```
C  SUBROUTINE VOOR HET BEREKENEN VAN HET AANDRIJFMOMENT:  
   INCLUDE 'USER3:[WWTMWV.XYNAG]SIGNAL.FOR'  
  
   END
```