

Formulering van de bewegingsvergelijkingen in AUTOLEV volgens Kane

Citation for published version (APA):

Sweegers, I. T. A. M., & Nederkoorn, B. (1992). *Formulering van de bewegingsvergelijkingen in AUTOLEV volgens Kane*. (DCT rapporten; Vol. 1992.086). Technische Universiteit Eindhoven.

Document status and date:

Gepubliceerd: 01/01/1992

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

Take down policy

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

openaccess@tue.nl

providing details and we will investigate your claim.

Formulering van de bewegingsvergelijkingen in
AUTOLEV volgens Kane

Ilphons Sweegers Bas Nederkoorn

27 juli 1992

Rapportnummer 92-086

Stagebegeleider: A.Sauren

Vakgroep WFW

Technische Universiteit Eindhoven

Inhoudsopgave

.1	Samenvatting	3
.2	Inleiding	4
.3	Notaties	6
.4	Mathematische symbolen	7
.5	Indices	7
1	Systeemmodellering	
1.1	Systeembeschrijving	8
1.2	Hoeksnelheden	13
1.3	Gegeneraliseerde hoeksnelheden	15
1.4	Partiële hoeksnelheden	19
1.5	Hoekversnellingen	21
1.6	Translatiesnelheden	22
1.7	Versnellingen	29
2	De bewegingsvergelijkingen	
2.1	Het opstellen van de bewegingsvergelijkingen	30
2.2	De bewegingsvergelijkingen volgens Kane	32
2.3	Eulerparameters	32
2.4	De functie AUTOZ	34
2.5	Het genereren van de bewegingsvergelijkingen	36
2.6	Controle van de invoer	39
2.7	Gebruik van de Eulerparameters	41
3	Simulatie	
3.1	Het genereren van een FORTRAN-programma	44
3.2	Invoer van de gegevens	44
4	Conclusies en aanbevelingen	47
A	Literatuurverwijzing	49
B	Uitwerking translatiesnelheid	50
B.1	2 ^e term in het rechterlid van (f39)	50
B.2	3 ^e term in het rechterlid van (f39)	51
B.3	4 ^e term in het rechterlid van (f39)	51
C	Uitdraai slinger.for	54

D Resultaten Simulaties	67
D.1 Voorbeelden output-files	67
D.2 Slinger zonder demping	70
D.3 Slinger met demping	71
D.4 Resultaten DADS versus AUTOLEV	72
E Gebruik van AUTOLEV onder DOS	75
E.1 Installatie	75
E.2 Werken binnen AUTOLEV	75
E.3 Compileren Fortranprogramma	75
E.4 Grafische verwerking van resultaten met Matlab	76
F Schematische aanpak modellering	77

0.1 Samenvatting

We bekijken een holonoom systeem bestaande uit een keten van starre lichamen B^k met $k=1,\dots,n$. We formuleren uitdrukkingen voor de positie, absolute (hoek)snelheden en absolute (hoek)versnellingen ten opzichte van de vaste wereld door gebruik te maken van de relatieve beschrijvingswijze. De representaties hiervan t.o.v. de \bar{e}^0 -basis schrijven we als functie van de gegeneraliseerde snelheden. Als we uitdrukkingen voor de absolute hoeksnelheden en absolute (hoek)versnellingen hebben, dan kunnen we deze gebruiken in de tweede wet van Newton en de impuls-momentvergelijking van Euler.

We schrijven de vergelijkingen van Newton en Euler om naar representaties ervan t.o.v. een inertiaalbasis en substitueren onze eerder gevonden uitdrukkingen voor de absolute (hoek)versnellingen en hoeksnelheden hierin. Na een transformatie worden deze vergelijkingen bij elkaar opgeteld en gesommeerd over alle lichamen. Op deze manier ontstaan n bewegingsvergelijkingen die het dynamisch gedrag van het hele systeem beschrijven.

Met het symbolisch MultiBody-pakket AUTOLEV wordt de theorie toegelicht door het formuleren van bewegingsvergelijkingen voor de dubbelslinger. Hier wordt ook een simulatie mee uitgevoerd. De formulering van de bewegingsvergelijkingen volgens Kane leidt snel tot goede resultaten, die ondersteund worden door een gebruikersvriendelijk AUTOLEV.

0.2 Inleiding

De MULTIBODY-DYNAMICA houdt zich bezig met het modelleren en simuleren van het kinematische en dynamische gedrag van een eindig aantal aan elkaar gekoppelde lichamen. Uit de V.S. is de formulering van de bewegingsvergelijkingen volgens Kane overgewaaid samen met het op die theorie gebaseerde programma AUTOLEV. Omdat over deze theorie steeds meer wordt gehoord, maar ook vanwege de kleine omvang van het programma, willen we graag weten wat de mogelijkheden en beperkingen van die theorie en het programma zijn.

Hiertoe is dit rapport opgesteld, dat ten eerste de bewegingsvergelijkingen volgens Kane in de hier gangbare wiskunde moet presenteren. De lezer zal derhalve matrix- en vectorrekening moeten beheersen en op het gebied van de dynamica onderlegd moeten zijn.

Het tweede doel is om een inleiding te geven op het gebruik van het programma AUTOLEV aan de hand van een voorbeeld. Het programma AUTOLEV past deze theorie toe. Hiervoor is een goede kennis van deze theorie vereist. We zullen daarom na elke theoretische stap een uitwerking geven binnen AUTOLEV, zodat we snel een verband kunnen leggen tussen theorie en praktijk.

De theorie volgens Kane is op de volgende manier opgebouwd:

- opstellen van de kinematische verbindingsbetrekkingen
- opstellen van de kinetische vergelijkingen
- combineren van deze twee tot bewegingsvergelijkingen

Kenmerken van de bewegingsvergelijkingen volgens Kane zijn het gebruik van gegeneraliseerde snelheden en de vervanging van uitwendige vectorprodukten door matrixvermenigvuldigingen.

Ook AUTOLEV heeft deze opbouw met eraan toegevoegd:

- controle van de ingevoerde systeembeschrijving
- simuleren dat wil zeggen het numeriek berekenen van de gevraagde output zoals verdraaiingen, snelheden, krachten en energieën

Dit rapport is op dezelfde manier opgebouwd waarbij de resultaten grafisch worden gepresenteerd, en vergeleken worden met resultaten verkregen met het multibodypakket DADS.

We hebben ons beperkt tot holonome systemen en een eenvoudig, in principe tweedimensionaal voorbeeld: de dubbelslinger. We menen echter dat dit voldoende is om in ieder geval een duidelijk idee te krijgen van de problemen die een rol spelen. AUTOLEV is een symbolisch pakket en eigenlijk niets meer dan een hulpmiddel om formules te genereren, zij

het dat het speciaal is uitgerust voor multibodydynamica. Ten behoeve van de simulatie gegeneert AUTOLEV een Fortranprogramma dat snel is door de efficiënte manier waarop de theorie is opgebouwd.

Om duidelijk het verschil tussen theorie en praktijk weer te geven, zijn voor de stukken die betrekking hebben op de theorie en op AUTOLEV aparte lettertypen gebruikt. Daarnaast zijn theoretische formules met ('fnr.')

 en formules binnen AUTOLEV met ('nr.') aangeduid. Vectoren zullen in principe worden aangeduid met een $\vec{}$. AUTOLEV gebruikt echter een notatie zonder $\vec{}$. Bij het gebruik van die notatie zal uit de context blijken wanneer het om vectoren gaat.

0.3 Notaties

n	Aantal beschouwde lichamen
m	Massa
l	Lengte
g	Gravitatieversnelling
\vec{e}	Basis
ϕ	Hoek
ω	Hoeksnelheid
$\hat{\omega}$	Relatieve hoeksnelheid
α	Hoekversnelling
r, b, c	Posities
v	Snelheid
a	Versnelling
y	Gegeneraliseerde snelheid
Z	Partiële hoeksnelheid
z	Komponent van de partiële hoeksnelheidsmatrix \underline{Z}
Y	Partiële translatiesnelheid
ν	Komponenten van de partiële translatiesnelheidsmatrix \underline{Y}
J	Centrale traagheidstensor
R	Rotatietensor
Ω	Hoeksnelheidstensor
H	Verbindingselement
O	Referentiepunt van een lichaam
N	Verbindingspunt
M^k	Massamiddelpunt
\vec{M} of \underline{M}	Moment
F	Kracht

0.4 Mathematische symbolen

- $\stackrel{def}{\equiv}$ Wordt gedefinieerd als
- $\stackrel{(\cdot)}{\equiv}$ Met behulp van vergelijking (\cdot)
- \dot{a} Tijdsafgeleide
- $\vec{a} * \vec{a}$ Uitwendig produkt
- $\vec{a} \cdot \vec{a}$ Inwendig produkt
- \vec{a} Vector
- a Kolom
- A Matrix
- $\sum_{j=0}^n$ Sommeren over alle j voor $j = 0, \dots, n$

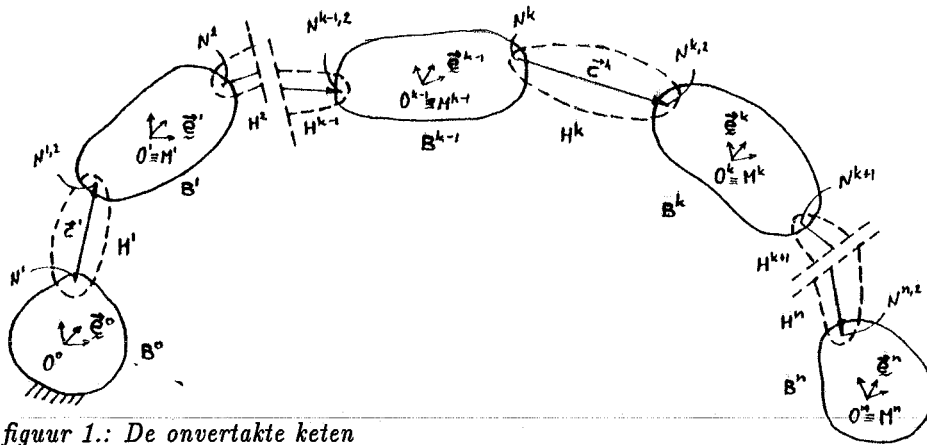
0.5 Indices

- ${}^i a_k^j$ i Referentiebasis met $i = 0, \dots, n-1$
- j Beschouwde basis met $j = 0, \dots, n$
- k Hoofdrichtingskomponent met $k = 1, 2, 3$

1 Systeemmodellering

1.1 Systeembeschrijving

Om de theorie eenvoudig uit te leggen bekijken we een onvertakte keten (boomstructuur) die uit n starre lichamen B^1 t/m B^n bestaat, ten opzichte van de globale referentie of vaste wereld B^0 . Voor de vaste wereld geldt dat de absolute versnelling nul is.



figuur 1.: De onvertakte keten

Voor elk lichaam B^k zijn de volgende grootheden gedefinieerd:

- Er is een lichaamsgebonden rechtsdraaiende orthogonale basis $(\vec{e}^k)^T = [\vec{e}_1^k \ \vec{e}_2^k \ \vec{e}_3^k]$. Vanaf nu zal elke basis rechtsdraaiend en orthogonaal worden gedefinieerd.
- Deze basis heeft zijn oorsprong in het lichaamsgebonden referentiepunt O^k , dat samenvalt met het massamiddelpunt M^k .
- Op elk lichaam B^k bevinden zich twee punten $N^{k,2}$ en $N^{k+1,1}$. Het lichaam B^k is met lichaam B^{k-1} verbonden door verbindingselement H^k en met lichaam B^{k+1} door verbindingselement H^{k+1} . Het punt $N^{k,2}$ is het verbindingspunt tussen H^k en B^k en het punt $N^{k+1,1}$ is het verbindingspunt tussen B^k en H^{k+1} .
- Als verbindingsassenstelsel in de punten $N^{k,2}$ en $N^{k+1,1}$ wordt de basis \vec{e}^k van lichaam B^k gekozen.

Voor verbindingselement H^k geldt verder nog:

- Verbindingselement H^k wordt gekarakteriseerd door de verbindingsvector \bar{c}^k en de verbindingstensor \mathbf{C}^k . De vector \bar{c}^k is de positievector van $N^{k,2}$ ten opzichte van N^k waarvoor geldt:

$$(f1) \bar{c}^k \stackrel{\text{def}}{=} (\bar{\underline{e}}^{k-1})^T {}^{k-1}\underline{c}^k = (\bar{\underline{e}}^{k-1})^T \begin{bmatrix} {}^{k-1}c_1^k \\ {}^{k-1}c_2^k \\ {}^{k-1}c_3^k \end{bmatrix}$$

- H^k wordt massaloos verondersteld dat wil zeggen dat H^k geen bijdrage kan leveren aan de kinetische energie van het totale systeem. Bij lijn- en bolscharnieren vallen bovendien de punten N^k en $N^{k,2}$ samen.

Voor een overzichtelijke notatie schrijven we in plaats van ${}^{k-1}\underline{c}^k$ in het vervolg \underline{c}^k dus:

$$(f2) {}^{k-1}\underline{c}^k \stackrel{\text{def}}{=} \underline{c}^k$$

De tensor \mathbf{C}^k beschrijft de oriëntatie van basis $\bar{\underline{e}}^k$ ten opzichte van basis $\bar{\underline{e}}^{k-1}$ volgens:

$$(f3) (\bar{\underline{e}}^k)^T \stackrel{\text{def}}{=} \mathbf{C}^k \cdot (\bar{\underline{e}}^{k-1})^T$$

Door substitutie van $\mathbf{C}^k = (\bar{\underline{e}}^k)^T \underline{\mathbf{C}}^k \bar{\underline{e}}^k$ in (f3) volgt:

$$(f4) \bar{\underline{e}}^{k-1} = \underline{\mathbf{C}}^k \bar{\underline{e}}^k$$

$\underline{\mathbf{C}}^k$ is de matrixrepresentatie van de verbindingstensor \mathbf{C}^k ten opzichte van zowel de basis $\bar{\underline{e}}^k$ als de basis $\bar{\underline{e}}^{k-1}$ en is evenals \mathbf{C}^k een functie van de tijd.

Bekijken we nu lichaam B^1 t/m B^k dan geldt hiervoor:

$$(f5) \bar{\underline{e}}^0 = \underline{\mathbf{C}}^1 \bar{\underline{e}}^1; \quad \bar{\underline{e}}^1 = \underline{\mathbf{C}}^2 \bar{\underline{e}}^2; \quad \bar{\underline{e}}^{k-1} = \underline{\mathbf{C}}^k \bar{\underline{e}}^k$$

Successieve substitutie van deze relatieve uitdrukkingen leidt tot een beschrijving van basis $\bar{\underline{e}}^k$ ten opzichte van de vaste wereld:

$$(f6) \bar{\underline{e}}^0 = \underline{\mathbf{C}}^1 \underline{\mathbf{C}}^2 \dots \bar{\underline{e}}^k$$

De oriëntatie van de lokale lichaamsgebonden basis $\bar{\underline{e}}^k$ ten opzichte van de vaste wereld B^0 wordt ook volledig en eenduidig beschreven door de

rotatietensor R^k volgens:

$$(f7) (\tilde{e}^k)^T = R^k \cdot (\tilde{e}^0)^T$$

of in matrixrepresentatie:

$$(f8) \tilde{e}^0 = \underline{R}^k \tilde{e}^k$$

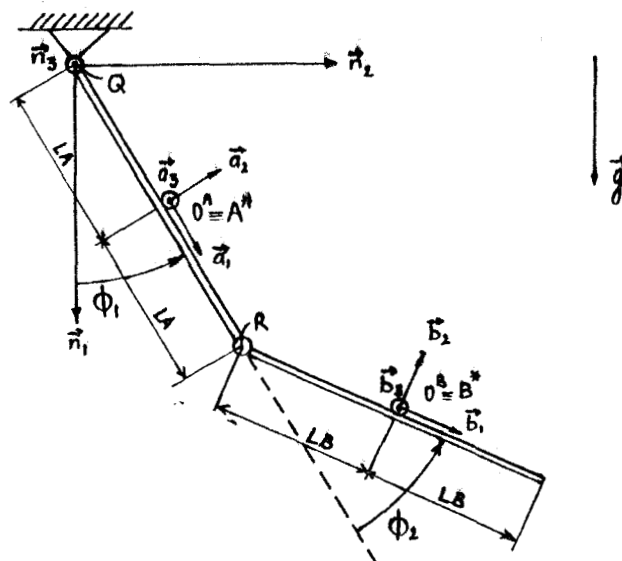
Vergelijken we (f8) en (f6) dan volgt hieruit:

$$(f9) \underline{R}^k = \underline{C}^1 \underline{C}^2 \dots \underline{C}^k$$

Definitie van het probleem

Het systeem wat we gaan beschrijven met AUTOLEV is een dubbelslinger (zie figuur 2). Dit systeem bestaat uit twee starre dunne cilindrische lichamen A en B. Lichaam A is met een lijnscharnier opgehangen aan de vaste wereld N in het punt Q. Het lichaam B is eveneens met een lijnscharnier verbonden met lichaam A in het punt R. Aan de lichamen A en B worden lichaamsgebonden verbindingsassenstelsels (frames) A en B gedacht, waarvan de referentiepunten O^A en O^B samenvallen met de massamiddelpunten A^* respectievelijk B^* . Aan de vaste wereld wordt het frame N toegekend. De frames N, A en B komen overeen met de lichaamsgebonden bases \tilde{e}^0 , \tilde{e}^1 en \tilde{e}^2 uit figuur 1. De lengten van de lichamen A en B zijn respectievelijk $2 \cdot LA$ en $2 \cdot LB$. Het systeem beweegt in een plat vlak dat wordt opgespannen door de vectoren $N1$ en $N2$. De hoek Φ_1 geeft de verdraaiing weer tussen lichaam A en de vaste wereld N en de hoek Φ_2 de verdraaiing tussen de lichamen A en B. De massa's van de lichamen A en B zijn respectievelijk $MASSA$ en $MASSB$.

Figuur 2: De dubbelslinger.



Voor dit systeem zullen worden bepaald de hoekverdraaiingen, de hoeksnelheden, de hoekversnellingen, optredende momenten, de kinetische en potentiële energie als functie van de tijd, voor een vrije beweging in het zwaartekrachtveld vanuit de beginpositie $\Phi_1 = \Phi_2 = 45^\circ$, met en zonder demping in de scharnierpunten.

Nu maken we een begin met het invoeren van het systeem in AUTOLEV. In appendix F staat het schema wat we aanhouden voor het programmeren van de dubbelslinger. Dit kan ook bij andere eenvoudige voorbeelden worden aangehouden.

We kunnen op elk moment tussen het invoeren van gegevens commentaar- en toelichtingsregels invoegen. Dit kan gedaan worden na een uitroep- teken, !. Vaak wordt een programma geopend met de naam van het probleem. Bijvoorbeeld:

(1) ! DUBBELSLINGER

(2) !

Een uitroep teken kan ook worden gebruikt om de regels van elkaar te scheiden. Om overzicht te houden op de invoer worden in het vervolg de commentaarregels van de invoer gescheiden.

Wanneer in de systeemvergelijkingen bepaalde deelexpressies meerdere malen voorkomen, zal AUTOLEV deze automatisch vervangen door hulpvergelijkingen Z1, Z2 etc. Dit doet AUTOLEV om de einduitdrukking niet al te lang te laten worden. Omdat het overzicht zoek raakt wanneer meerdere Z's worden gevormd, zullen we voor de duidelijkheid dan ook geen gebruik maken van deze optie. AUTOLEV gaat er bij het opstarten vanuit dat deze optie gewenst is en we moeten deze uitzetten met het commando:

(3) AUTOZ(OFF)

Als eerste wil AUTOLEV weten hoeveel onafhankelijke graden van vrijheid (Degrees Of Freedom) het systeem heeft. Dit aantal moeten we zelf bepalen. Ons systeem heeft twee graden van vrijheid: bijvoorbeeld Φ_1 en Φ_2 . Dit wordt in AUTOLEV als volgt ingevoerd:

(4) DOF(2)

Vervolgens worden de (starre) lichamen, exclusief de vaste wereld, geïntroduceerd met het commando FRAMES(A,B). Met dit commando

weet AUTOLEV dat er twee lichamen A en B in het systeem voorkomen. Het commando introduceert tevens:

- lichaamsgebonden rechtsdraaiende vectorbases met vectoren A1, A2, A3 respectievelijk B1, B2, B3.
- massamiddelpunten ASTAR en BSTAR.
- lichaamsmassa's MASSA en MASSB.

Voor de vaste (of Newtoniaanse) wereld hoeven we geen frame toe te kennen. Er wordt door AUTOLEV zelf een rechtsdraaiende set lichaamsgebonden orthogonale basisvectoren geïntroduceerd, N1, N2 en N3. Deze komen overeen met de set basisvectoren behorende bij de vaste wereld: $(\vec{e}^0)^T = [\vec{e}_1^0 \quad \vec{e}_2^0 \quad \vec{e}_3^0]$.

(5) FRAMES(A,B)

We kunnen nu de oriëntaties van de verbindingssassenstelsels volgens (f4) invoeren met behulp van het commando SIMPROT (SIMple ROTation). Om de verdraaiing van een lichaam B^k (in dit geval A) t.o.v. een vorig lichaam B^{k-1} (de vaste wereld N) te bepalen moeten het voorgaande en volgende frame worden ingevoerd. Vervolgens moet de as aangegeven worden die tijdens de rotatie gemeenschappelijk blijft (3 van N3 en A3) en als laatste de hoek (PHI1) die de grootte van de rotatie aangeeft.

Dus:

(6) SIMPROT(N,A,3,PHI1)

AUTOLEV reageert hierop met het geven van het antwoord achter het symbool ->, in de vorm van de richtingscosinussen (DIRection COSines) en doet dit als volgt:

-> (7) DIRCOS(N,A,COS(PHI1),-SIN(PHI1),0,SIN(PHI1),
COS(PHI1),0,0,0,1)

Hierin staan de richtingscosinussen voor lichaam A ten opzichte van de vaste wereld N. Deze komen overeen met de componenten van de verbindingsmatrix \underline{C}^1 uit formule (f5) ten opzichte van \vec{e}^1 -basis (frame A) (of \vec{e}^0 (frame N)):

$$\underline{C}^1 = \underline{R}^1 = \begin{bmatrix} \cos \Phi_1 & -\sin \Phi_1 & 0 \\ \sin \Phi_1 & \cos \Phi_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Dit commando moet, op overeenkomstige wijze als onder (6), voor iedere verbinding worden gegeven. De oriëntatie van lichaam B ten opzichte van lichaam A wordt als volgt gevonden:

(8) SIMPROT(A,B,3,PHI2)

AUTOLEV reageert hierop met:

-> (9) DIRCOS(A,B,COS(PHI2),-SIN(PHI2),0,SIN(PHI2),
COS(PHI2),0,0,0,1)

Uitdrukking (9) komt overeen met de componenten van de verbindingsmatrix \underline{C}^2 uit formule (f5) ten opzichte van de \tilde{e}^2 -basis (frame B) of \tilde{e}^1 (frame A). Met het commando DIRCOS(N,B) worden de richtingscosinussen van basis B ten opzichte van de basis N berekend.

(10) DIRCOS(N,B)

-> (11) DIRCOS(N,B,COS(PHI1+PHI2),-SIN(PHI1+PHI2),0,
SIN(PHI1+PHI2),COS(PHI1+PHI2),0,0,0,1)

Deze komen overeen met de componenten van de rotatiematrix \underline{R}^2 uit formule (f9) met:

$$\underline{R}^2 = \begin{bmatrix} \cos(\Phi_1 + \Phi_2) & -\sin(\Phi_1 + \Phi_2) & 0 \\ \sin(\Phi_1 + \Phi_2) & \cos(\Phi_1 + \Phi_2) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

1.2 Hoeksnelheden

Nu we de relatieve en absolute oriëntatie van de frames hebben vastgelegd en de vrijheidsgraden zijn gedefinieerd, kunnen we beginnen met het formuleren van de hoeksnelheden. De translatiesnelheden komen later aan bod, omdat hiervoor ook de hoeksnelheden bekend moeten zijn.

Teneinde de kinetische en potentiële energie, de beweging en het krachterspel van het systeem te kunnen beschrijven, zijn er uitdrukkingen voor de hoek- en translatie- snelheden vereist. Hiervoor moeten zij echter wel uitgedrukt zijn in de componenten ten opzichte van de \tilde{e}^0 -basis wil er een handelbare vergelijking ontstaan. Daarom zal zowel bij de hoeksnelheden als later bij de translatiesnelheden een overgang moeten worden gemaakt van relatieve naar absolute snelheden.

We definiëren $\vec{\omega}^{k-1}$ en $\vec{\omega}^k$ als de absolute hoeksnelheidsvectoren van lichaam B^{k-1} respectievelijk B^k . De relatieve hoeksnelheidsvector van B^k ten opzichte van B^{k-1} wordt als volgt gedefinieerd:

$$(f10) \hat{\vec{\omega}}^k \stackrel{\text{def}}{=} \vec{\omega}^k - \vec{\omega}^{k-1}$$

Deze relatieve hoeksnelheidsvector heeft dus betrekking op verbindings-element H^k .

$$(f11) {}^{k-1}\hat{\omega}^k = \vec{e}^{k-1} \cdot \hat{\vec{\omega}}^k$$

waarin ${}^{k-1}\hat{\omega}^k$ de kolomrepresentatie, is van de relatieve hoeksnelheid van lichaam B^k ten opzichte van B^{k-1} uitgedrukt in de basis \vec{e}^{k-1} . Evenals bij \vec{e}^k noteren we ook voor ${}^{k-1}\hat{\omega}^k$ in het vervolg $\hat{\omega}^k$, dus:

$$(f12) {}^{k-1}\hat{\omega}^k \stackrel{\text{def}}{=} \hat{\omega}^k = \begin{bmatrix} \hat{\omega}_1^k \\ \hat{\omega}_2^k \\ \hat{\omega}_3^k \end{bmatrix}$$

Voor $k=0$ bestaat er in feite geen relatieve hoeksnelheidsvector, maar we definiëren $\hat{\vec{\omega}}^0 = \vec{0}$. Zodat nu ook geldt:

$$(f13) \vec{\omega}^1 = \hat{\vec{\omega}}^1$$

De absolute hoeksnelheid van lichaam B^k is nu te schrijven als:

$$(f14) \vec{\omega}^k = \hat{\vec{\omega}}^1 + \hat{\vec{\omega}}^2 + \dots + \hat{\vec{\omega}}^k$$

Er geldt verder:

$$(f15) (\vec{e}^k)^T = \vec{\omega}^k * (\vec{e}^k)^T$$

Dit uitwendig produkt kunnen we omzetten naar een eenvoudige tensorvermenigvuldiging door:

$$(f16) \vec{\omega}^k * (\vec{e}^k)^T = \Omega^k \cdot (\vec{e}^k)^T = \Omega^k \cdot \mathbf{R}^k \cdot (\vec{e}^0)^T$$

met :

$$(f17) \Omega^k \stackrel{\text{def}}{=} \dot{\mathbf{R}}^k \cdot (\mathbf{R}^k)^c$$

De vector $\vec{\omega}^k$ is dus de axiaalvector van de tensor Ω^k .

De matrixrepresentatie van de hoeksnelheidstensor Ω^k ten opzichte van de \vec{e}^0 -basis luidt nu:

$$(f18) \quad \underline{\Omega}^k = (\vec{e}^0)^T \underline{\Omega}^k \vec{e}^0$$

met:

$$(f19) \quad \underline{\Omega}^k = \begin{bmatrix} 0 & -{}^0\omega_3^k & {}^0\omega_2^k \\ {}^0\omega_3^k & 0 & -{}^0\omega_1^k \\ -{}^0\omega_2^k & {}^0\omega_1^k & 0 \end{bmatrix}$$

Drukken we de matrixrepresentaties van de tensoren Ω^k en \mathbf{R}^k uit ten opzichte van de basis \vec{e}^0 dan volgt uit (f16), (f18) en $\mathbf{R}^k = (\vec{e}^0)^T \underline{\mathbf{R}}^k \vec{e}^0$ de matrixvermenigvuldiging:

$$(f20) \quad \dot{\vec{e}}^k = (\underline{\mathbf{R}}^k)^T (\underline{\Omega}^k)^T \vec{e}^0$$

De hoeksnelheidsvector $\vec{\omega}^k$ is nu ook met (f19) uit te drukken ten opzichte van de \vec{e}^0 -basis volgens:

$$(f21) \quad \vec{\omega}^k = [\vec{e}_1^0 \quad \vec{e}_2^0 \quad \vec{e}_3^0] \begin{bmatrix} {}^0\omega_1^k \\ {}^0\omega_2^k \\ {}^0\omega_3^k \end{bmatrix} = (\vec{e}^0)^T {}^0\omega^k$$

Opmerking: de differentiatie van de basis \vec{e}^k naar de tijd is dus omgezet naar een matrixvermenigvuldiging (f20). Voor computerberekeningen is dit een efficiënt algoritme.

1.3 Gegeneraliseerde hoeksnelheden

Het doel waar we heen willen met deze multi-bodytheorie is om per lichaam te komen tot de bewegingsvergelijkingen volgens Newton en Euler. Deze luiden als volgt:

$$\text{Newton : } \vec{F}^k = m^k \vec{a}^k$$

$$\text{Euler : } \vec{M}^k = \mathbf{J}^k \cdot \vec{\alpha}^k + \vec{\omega}^k * (\mathbf{J}^k \cdot \vec{\omega}^k)$$

met:

- \vec{a}^k de absolute versnelling van M^k
- m^k de massa van B^k

- J^k de centrale traagheidstensor van B^k
- $\vec{\alpha}^k$ de absolute hoekversnelling van B^k
- $\vec{\omega}^k$ de absolute hoeksnelheid van B^k

Hieruit volgt het belang van uitdrukkingen voor de absolute versnellingen, hoekversnellingen en hoeksnelheden. De vorm waarin deze worden aangedragen heeft veel invloed op de complexiteit van de vergelijkingen. De volgende keuze van een kolom gegeneraliseerde hoeksnelheden maakt het mogelijk om per deelsysteem (lichaam B^k en verbindingselementen H^k en H^{k+1}) de bewegingsvergelijkingen op te stellen. Achteraf worden deze bewegingsvergelijkingen via een transformatie omgezet naar vergelijkingen in componenten ten opzichte van de vaste \vec{e}^0 -basis. Deze kolom gegeneraliseerde hoeksnelheden ziet er als volgt uit:

$$(f22) \underline{y} = \begin{bmatrix} \hat{\omega}^1 \\ \hat{\omega}^2 \\ \vdots \\ \hat{\omega}^k \\ \vdots \\ \hat{\omega}^n \end{bmatrix} \quad \begin{array}{l} \text{Hierin staan } 3n \text{ componenten die} \\ \text{worden gevormd door de represen-} \\ \text{tatie van de relatieve hoeksnelheids-} \\ \text{vectoren } \vec{\omega}^k \text{ over element } H^k \text{ ten op-} \\ \text{zichte van de voorgaande basis } \vec{e}^{k-1} \\ \text{voor } k=1 \text{ t/m } n. \end{array}$$

Voor de componenten van de kolom gegeneraliseerde hoeksnelheden is uit (f11) eenvoudig af te leiden dat geldt:

$$(f23) \hat{\omega}^k = {}^{k-1}\hat{\omega}^k = \vec{e}^{k-1} \cdot \hat{\omega}^k$$

We gaan nu de kolom gegeneraliseerde hoeksnelheden in AUTOLEV definiëren. De gegeneraliseerde hoeksnelheden moeten worden aangeduid in de vorm U_k met 'k' het nummer behorende bij de k^e graad van vrijheid. Vanaf nu worden de gegeneraliseerde hoeksnelheden U_k genoemd. Aan iedere graad van vrijheid wordt een gegeneraliseerde hoeksnelheid gekoppeld. Voor de dubbelslinger worden U_1 en U_2 voor de eerste respectievelijk tweede gegeneraliseerde hoeksnelheid in AUTOLEV gedefinieerd. Deze worden gevormd door de componenten van de relatieve hoeksnelheidsvectoren ten opzichte van de basis van het voorgaande lichaam volgens (f22) en f23). De U_k 's zijn door het commando DOF automatisch gedefinieerd als variabelen. In het geval van de dubbelslinger volgt voor de kolom y :

$$\underline{y} = \begin{bmatrix} 0 \\ 0 \\ U1 \\ 0 \\ 0 \\ U2 \end{bmatrix}$$

Het aantal Uk's in deze kolom mag niet groter zijn dan het aantal graden van vrijheid (dof) en moeten de naam krijgen U1...Udof.

Vervolgens moet aan AUTOLEV opgegeven worden hoe we de Uk's hebben gekozen. Bij de formulering hiervan in AUTOLEV mag bij de definitie van een Uk een tijdsafgeleide voorkomen. Dit is alleen toegestaan bij de definities van de Uk's. De Uk's moeten aan het begin worden ingevoerd, zodat uitdrukkingen met tijdsafgeleiden door AUTOLEV zelf omgezet kunnen worden in uitdrukkingen met Uk's. In het geval van de dubbelslinger zijn de Uk's de tijdsafgeleiden van de twee relatieve hoeken Φ_1 en Φ_2 volgens:

$$(12) \text{ PHI1}' = U1$$

waarin $\text{PHI1} = \Phi_1$ en $' = d/dt$.

$$(13) \text{ PHI2}' = U2$$

waarin $\text{PHI2} = \Phi_2$.

Nu kunnen we met AUTOLEV de relatieve hoeksnelheden bepalen door zelf de vergelijking in te voeren voor de relatieve hoeksnelheid. Een hoeksnelheid in AUTOLEV moet worden aangeduid in de vorm $W[\]\{\}$ met $[\]$ de naam van het frame dat wordt beschouwd en $\{\}$ de naam van het referentiefraam. Grootheden zoals (hoek)snelheden, (hoek)versnellingen, vectoren e.d. moeten volgens een door AUTOLEV bepaalde lettercombinatie worden opgeschreven. Men kan deze 'syntactical forms' zoals AUTOLEV ze noemt, opvragen door het geven van het commando FORMS. Voor de relatieve hoeksnelheid van lichaam A ten opzichte van de vaste wereld N geldt:

$$(14) \text{ WAN} = \text{PHI1}' * N3$$

als WAN de relatieve hoeksnelheid van lichaam A is ten opzichte van de vaste wereld N, uitgedrukt in de basis N. WAN komt overeen met (f11) met $k=1$.

AUTOLEV reageert met behulp van (12) met:

$$\rightarrow (15) \text{ WAN} = \text{U1} * \text{N3}$$

Voor de relatieve hoeksnelheid van lichaam B ten opzichte van lichaam A geldt:

$$(16) \text{ WBA} = \text{PHI2}' * \text{A3}$$

als WBA de relatieve hoeksnelheid is van lichaam B ten opzichte van lichaam A, uitgedrukt in de basis A. WBA komt overeen met (f11) met $k=2$.

AUTOLEV reageert met behulp van (13) met:

$$\rightarrow (17) \text{ WBA} = \text{U2} * \text{A3}$$

Om de absolute hoeksnelheid van lichaam B ten opzichte van de vaste wereld N te krijgen, kunnen we volgens (f14) de relatieve hoeksnelheden van lichaam A en B bij elkaar optellen. Met de routine $\text{WBN} = \text{ADD}(\text{WAN}, \text{WBA})$ worden de relatieve hoeksnelheidsvectoren WAN en WBA opgeteld.

$$(18) \text{ WBN} = \text{ADD}(\text{WAN}, \text{WBA})$$

waarin WBN overeenkomt met (f21) met $k=2$.

$$\rightarrow (19) \text{ WBN} = \text{U1} * \text{N3} + \text{U2} * \text{A3}$$

Met het commando $\text{EXPRESS}(\text{WBN}, \text{N})$ wordt het rechterlid van de vergelijking voor WBN uitgedrukt in de basis N overeenkomend met formule (f21).

$$(20) \text{ EXPRESS}(\text{WBN}, \text{N})$$

$$\rightarrow (21) \text{ WBN} = (\text{U1} + \text{U2}) * \text{N3}$$

Opmerking: Het commando $\text{EXPRESS}(\text{WAN}, \text{N})$ hoeft niet gegeven te worden, omdat WAN tevens de absolute hoeksnelheid is van lichaam A, overeenkomend met (f13).

Nu zijn alle absolute hoeksnelheden $\vec{\omega}^k$ ($k=1,2$) ten opzichte van de vaste wereld bekend.

1.4 Partiële hoeksnelheden

Voor de absolute hoeksnelheid $\vec{\omega}^k$ van lichaam B^k geldt volgens (f21):

$$(f21) \quad \vec{\omega}^k = [\vec{e}_1^0 \quad \vec{e}_2^0 \quad \vec{e}_3^0] \begin{bmatrix} {}^0\omega_1^k \\ {}^0\omega_2^k \\ {}^0\omega_3^k \end{bmatrix} = (\vec{e}^0)^T {}^0\omega^k$$

De kolom ${}^0\omega^k$ gaan we nu uitdrukken in termen van de gegeneraliseerde hoeksnelheden in y .

Uit formule (f10) en (f14) volgt:

(f24)

$$\begin{aligned} \vec{\omega}^1 &= \hat{\omega}^1 \\ \vec{\omega}^2 &= \vec{\omega}^1 + \hat{\omega}^2 = \hat{\omega}^1 + \hat{\omega}^2 \\ &\vdots \\ \vec{\omega}^k &= \hat{\omega}^1 + \dots + \hat{\omega}^k \end{aligned}$$

Met behulp van (f10) en (f8) komen we tot:

$$(f25) \quad \hat{\omega}^k = (\vec{e}^{k-1})^T \hat{\omega}^k = (\vec{e}^0)^T \underline{R}^{k-1} \hat{\omega}^k \Rightarrow {}^0\hat{\omega}^k = \underline{R}^{k-1} \hat{\omega}^k$$

Uit (f24) en (f25) volgt:

$$(f26) \quad \begin{aligned} {}^0\omega^k &= {}^0\hat{\omega}^1 + {}^0\hat{\omega}^2 + \dots + {}^0\hat{\omega}^k \\ &= \underline{R}^0 \hat{\omega}^1 + \underline{R}^1 \hat{\omega}^2 + \dots + \underline{R}^{k-1} \hat{\omega}^k \end{aligned}$$

Met de laatste vergelijking en de definitie van de kolom gegeneraliseerde hoeksnelheden zijn de globale componenten van de hoeksnelheden $\vec{\omega}^k$ ($k=1, \dots, n$) uit te drukken in termen van de gegeneraliseerde hoeksnelheden y :

$$(f27) \quad {}^0\omega^k = [\underline{R}^0 \quad \underline{R}^1 \quad \dots \quad \underline{R}^{k-1} \quad : \quad \underline{0}] y$$

Anders geschreven:

$$(f28) \quad {}^0\omega^k \stackrel{\text{def}}{=} \underline{Z}^k y$$

waarbij de matrix \underline{Z}^k de zogenaamde partiële hoeksnelheden bevat.

Opmerkingen: $\underline{Z}^k \neq \underline{Z}^{k+1}$ oftewel elk lichaam B^k heeft zijn eigen matrix \underline{Z}^k met partiële snelheden. \underline{Z}^{k+1} is echter wel eenvoudig te verkrijgen

door bij \underline{z}^k de volgende matrix op te tellen:

$$(f29) \left[\begin{array}{c|c|c} \underline{0} & \underline{R}^k & \underline{0} \\ \leftarrow 3 \times 3k \rightarrow & \leftarrow 3 \times 3 \rightarrow & \leftarrow 3 \times (6n-3(k-1)) \rightarrow \end{array} \right]$$

Opmerking: Partiële hoeksnelheden zijn evenals de matrices \underline{R}^k dimensieloos!

Met het commando PANGVEL(FRAME) (Partial ANGular VELOCITY) worden alle partiële hoeksnelheden in het opgegeven frame als volgt berekend:

(22) PANGVEL(B)

waarin B het frame is waarin de partiële hoeksnelheden moeten worden bepaald.

AUTOLEV reageert hierop met het geven van de partiële hoeksnelheden ten opzichte van het frame B.

-> (23) WBN1=N3

WBN1 is de eerste partiële hoeksnelheid uit (19).

-> (24) WBN2=A3

WBN2 is de tweede partiële hoeksnelheid uit (19).

De uitdrukking $WBN1*U1 + WBN2*U2$ geeft dan de absolute hoeksnelheid van lichaam B weer. Voordat we inzien dat dit overeenkomt met de theorie moeten we ons het volgende realiseren. Het is voor AUTOLEV niet van belang in welke basis een grootheid is uitgedrukt. Bovendien zien we dat AUTOLEV de uitkomsten in vectornotaties laat staan, in tegenstelling tot onze theorie, die representaties ten opzichte van bases gebruikt. Formule (f27) uitgewerkt voor lichaam B levert:

$${}^N \underline{\omega}^B = \begin{bmatrix} 1 & 0 & 0 & \cos(\Phi_1) & -\sin(\Phi_1) & 0 \\ 0 & 1 & 0 & \sin(\Phi_1) & \cos(\Phi_1) & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \underline{y}$$

Vermenigvuldigen we bovenstaande uitdrukking vóór met basis N ($(\underline{\bar{e}}^0)^T$) en schrijven we WBN1 en WBN2 beide ten opzichte van de N-basis ($WBN2=A3=N3$) dan komt dit overeen met $WBN1*U1 + WBN2*U2$.

Wanneer we geïnteresseerd zijn in één van de partiële hoeksnelheden kunnen we deze laten berekenen met PANGVEL(FRAME,n) waarin n de n^e partiële snelheid is van FRAME ten opzichte van N.

1.5 Hoekversnellingen

Een uitdrukking voor de absolute componenten van de hoekversnellingen is makkelijk te vinden door uitdrukking (f28) te differentiëren naar de tijd.

$$(f30) \quad {}^0\dot{\omega}^k = \dot{\omega}^k = \dot{Z}^k \underline{y} + Z^k \dot{\underline{y}}$$

$$(f31) \quad \dot{Z}^k = \begin{bmatrix} \dot{R}^0 & \dot{R}^1 & \dots & \dot{R}^{k-1} & \vdots & \underline{0} \\ \underline{\Omega}^0 \underline{R}^0 & \underline{\Omega}^1 \underline{R}^1 & \dots & \underline{\Omega}^{k-1} \underline{R}^{k-1} & \vdots & \underline{0} \end{bmatrix}$$

(zie ook (f17)).

Met behulp van het commando DERIV(VEC,X,F) voert AUTOLEV een partiële differentiatie uit van VEC naar de variabele X in frame F. Wanneer deze variabele de tijd is wordt de vector gewoon naar de tijd gedifferentieerd. Bij gebruik van dit commando moeten we hierbij vooraf een naam aan de uitkomst geven.

Hoekversnellingen zijn net als de hoeksnelheden gebonden aan naamgeving en wel als volgt: ALF[]{ } is de hoekversnelling van frame [] in frame { }. Volgens (f30) en (f21) met k=1 geldt dan:

$$(25) \quad \text{ALFAN} = \text{DERIV}(\text{WAN}, \text{T}, \text{N})$$

waarin ALFAN de absolute hoekversnelling is van lichaam A ten opzichte van de vaste wereld N en N de basis is waarin gedifferentieerd wordt naar de tijd (N, A of B invullen maakt niets uit).

$$\rightarrow (26) \quad \text{ALFAN} = \text{U1}' * \text{N3}$$

waarin ' = d/dt.

Hier zien we dat AUTOLEV reageert met een tijdsafgeleide in het rechterlid. Het betreft hier echter een tijdsafgeleide van een Uk wat wél is toegestaan.

Ook de absolute hoekversnelling van lichaam B moet worden bepaald. Dit gaat op overeenkomstige wijze als bij (25).

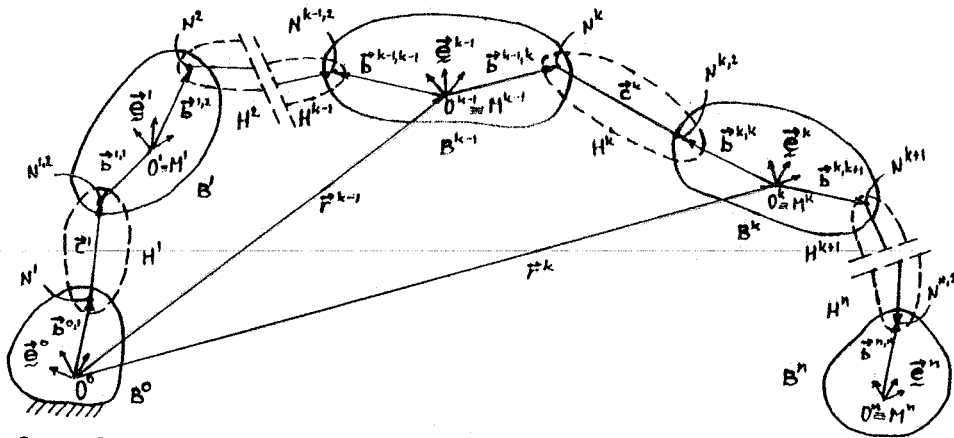
(27) ALFBN=DERIV(WBN,T,N)

-> (28) ALFBN = (U1'+U2')*N3

ALFBN volgens (f30) met k=2.

1.6 Translatiesnelheden

Nu de uitdrukkingen van de hoeksnelheden en -versnellingen bekend zijn voor alle lichamen B^k , kunnen we overgaan tot het berekenen van de translatiesnelheden.



figuur 3

Op het lichaam B^k worden nog twee grootheden gedefinieerd (zie ook figuur 3):

- Een vector $\vec{b}^{k,k}$ beschrijft de positie van verbindingspunt $N^{k,2}$ ten opzichte van het massamiddelpunt O^k .
- Een vector $\vec{b}^{k,k+1}$ beschrijft de positie van verbindingspunt N^{k+1} ten opzichte van het massamiddelpunt O^k .

Bovendien beschrijft de vector \vec{r}^k de positie van het referentiepunt O^k (= massamiddelpunt M^k) van lichaam B^k ten opzichte van het punt O^0 op de vaste wereld B^0 .

De vector \vec{r}^k (met $k=1, \dots, n$) naar lichaam B^k wordt samengesteld door telkens de absolute positievector \vec{r}^{k-1} van lichaam B^{k-1} op te tellen bij de vectoren die de positie O^k ten opzichte van O^{k-1} beschrijven (zie figuur 3).

In formulevorm ziet dit er als volgt uit:

(f32)

$$\begin{aligned}\vec{r}^0 &= \vec{0} \\ \vec{r}^1 &= \vec{r}^0 + \vec{b}^{0,1} + \vec{c}^1 - \vec{b}^{1,1} \\ &\vdots \\ \vec{r}^k &= \vec{r}^{k-1} + \vec{b}^{k-1,k} + \vec{c}^k - \vec{b}^{k,k}\end{aligned}$$

Vervang omwille van de overzichtelijkheid de indices (k-1) en k door i respectievelijk j, dan geldt dus voor de positievector \vec{r}^j van punt O^0 naar O^j :

$$(f33) \quad \vec{r}^j = \vec{r}^i + \vec{b}^{i,j} + \vec{c}^j - \vec{b}^{j,j}$$

De positievector van massamiddelpunt O^j is dus nu bekend ten opzichte van de vaste wereld. Door differentiëren naar de tijd volgt dan voor de absolute snelheid \vec{v}^j van O^j :

$$(f34) \quad \vec{v}^j = \dot{\vec{r}}^j = \dot{\vec{r}}^i + \dot{\vec{b}}^{i,j} + \dot{\vec{c}}^j - \dot{\vec{b}}^{j,j}$$

Voor de tweede term van het rechterlid van (f34) geldt:

$$(f35) \quad \dot{\vec{b}}^{i,j} = \vec{\omega}^i * \vec{b}^{i,j}$$

Dit is de snelheid van verbindingpunt N^j ten opzichte van referentiepunt O^i die het gevolg is van de absolute hoeksnelheid van lichaam B^i .

Analoog geldt voor de vierde term:

$$(f36) \quad \dot{\vec{b}}^{j,j} = \vec{\omega}^j * \vec{b}^{j,j}$$

Voor de derde term $\dot{\vec{c}}^j$ geldt:

(f37)

$$\begin{aligned}\vec{c}^j &= (\vec{e}^i)^T {}^i \underline{c}^j \\ \dot{\vec{c}}^j &= (\dot{\vec{e}}^i)^T {}^i \underline{c}^j + (\vec{e}^i)^T {}^i \dot{\underline{c}}^j\end{aligned}$$

en met (f15) volgt:

$$(f15) (\vec{e}^i)^T = \vec{\omega}^i * (\vec{e}^i)^T$$

$$(f38) \dot{\vec{c}}^j = \vec{\omega}^i * \vec{c}^j + (\vec{e}^i)^T \dot{\vec{c}}^j$$

Substitutie van (f35), (f36) en (f38) in (f34) geeft voor de snelheid \vec{v}^j de volgende uitdrukking:

$$(f39) \vec{v}^j = \vec{v}^i + \vec{\omega}^i * (\vec{b}^{i,j} + \vec{c}^j) + (\vec{e}^i)^T \dot{\vec{c}}^j - \vec{\omega}^j * \vec{b}^{j,j}$$

Bovenstaande uitdrukking is onhandelbaar en niet te gebruiken voor computerdoeleinden. De eerder gedefinieerde kolom gegeneraliseerde hoeksnelheden (f22) wordt nu uitgebreid met een kolom gegeneraliseerde translatiesnelheden. Dit levert op:

$$(f40) \underline{y} = \begin{bmatrix} \dot{\omega}^1 \\ \dot{\omega}^2 \\ \vdots \\ \dot{\omega}^k \\ \vdots \\ \dot{\omega}^n \\ \hline \dot{c}^1 \\ \dot{c}^2 \\ \vdots \\ \dot{c}^k \\ \vdots \\ \dot{c}^n \end{bmatrix}$$

Met de tweede 3n componenten: de representatie van de relatieve (translatie)snelheidsvectoren $\dot{\vec{c}}^k$ ten opzichte van de basis \vec{e}^{k-1} . Zij geven de relatieve snelheid weer van de punten $N^{k,2}$ ten opzichte van N^k over verbindingselementen H^k voor $k=1$ t/m n .

Voor de tweede set van 3n componenten uit de kolom gegeneraliseerde snelheden geldt:

$$(f41) \dot{c}^k \stackrel{\text{def}}{=} {}^{k-1}\dot{\vec{c}}^k = \vec{e}^{k-1} \cdot \dot{\vec{c}}^k$$

De kolom \underline{y} wordt vanaf nu de kolom gegeneraliseerde snelheden genoemd, deze bevat dus zowel gegeneraliseerde hoek- als translatiesnelheden. Volgens appendix B kunnen we nu (f39) omschrijven in de vorm:

$$(f42) {}^0\underline{v}^k = \underline{Y}^k \underline{y} \quad \text{met: } {}^0\underline{v}^k = \vec{e}^0 \vec{v}^k$$

en dit is wel een handelbare vergelijking voor de computer.

Met ${}^0y^k$ de representatie van de snelheid van het k^e lichaam ten opzichte van de vaste wereld uitgedrukt in de basis \vec{e}^0 . De matrix \underline{Y}^k bevat de partiële snelheden vergelijkbaar met de matrix \underline{Z}^k .

$$(f43) \underline{Y}^k = \sum_{i=1}^n \left(\underbrace{{}^0G^i \underline{Z}^{i-1}}_{1^e} + \underbrace{\underline{L}^i}_{2^e} + \underbrace{{}^0K^i \underline{Z}^i}_{3^e} \right)$$

Overeenkomend met de opbouw van (f34) heeft (f43) componenten die voortkomen uit de rotaties van de lichamen (vandaar het verschijnen van de matrices \underline{Z}^{i-1} en \underline{Z}^i in het eerste en derde gedeelte) en de verlengsnelheden van verbindingselement H^i (tweede gedeelte).

Voor deze matrices geldt (zie appendix B):

$$(f43a) \underbrace{{}^0G^i}_{3i} = \begin{bmatrix} 0 & -{}^0g_3^i & {}^0g_2^i \\ {}^0g_3^i & 0 & -{}^0g_1^i \\ -{}^0g_2^i & {}^0g_1^i & 0 \end{bmatrix}$$

$$(f43b) \underline{L}^i = -\underline{R}^{i-1} ({}^{i-1}b^{i-1,i} + {}^{i-1}c^i)$$

$$(f43c) \underline{Z}^i = [\underbrace{\underline{R}^0 \ \underline{R}^1 \ \dots \ \underline{R}^{i-1}}_{3i} \ ; \ \underbrace{\underline{0}}_{3(2n-i)}]$$

$$(f43d) \underline{L}^i = [\underbrace{\underline{0}}_{3(n+i-1)} \ ; \ \underbrace{\underline{R}^{i-1}}_3 \ ; \ \underbrace{\underline{0}}_{3(n-i)}]$$

$$(f43e) \underbrace{{}^0K^i}_{3i} = \begin{bmatrix} 0 & -{}^0b_3^{i,i} & {}^0b_2^{i,i} \\ {}^0b_3^{i,i} & 0 & -{}^0b_1^{i,i} \\ -{}^0b_2^{i,i} & {}^0b_1^{i,i} & 0 \end{bmatrix}$$

$$(f43f) \underline{b}^{i,i} = \underline{R}^i \underline{b}^{i,i}$$

De translatiesnelheden, die van belang zijn voor de bewegingsvergelijkingen, zijn de absolute snelheden VASTAR en VBSTAR van de massamiddelpunten ASTAR (theorie: $O^1 = M^1$) en BSTAR (theorie $O^2 = M^2$), voor lichaam A respectievelijk B. Aangezien de translatiesnelheden in ons geval afhankelijk zijn van de gegeneraliseerde hoeksnelheden U1 en U2, kunnen en moeten zij worden uitgedrukt in termen van de Uk's. In de uitdrukkingen voor de translatiesnelheden kunnen variabelen en/of constante parameters voorkomen. Met het commando VAR(X1,X2,...) worden de symbolen X1, X2,... als variabelen gedefinieerd. De Uk's, de tijd T en linkerzijden van vergelijkingen hoeven niet nog eens als variabele te worden benoemd. AUTOLEV neemt dit automatisch aan. Met het commando CONST(C1,C2,...),

worden de de symbolen C_1, C_2, \dots als constanten gedefinieerd. In het geval van de dubbelslinger komen geen variabelen maar wel constanten voor. Deze voeren we als volgt in (zie figuur 2):

(29) $\text{CONST}(L_A, L_B)$

waarin L_A en L_B de halve lengten van de staven A respectievelijk B zijn.

Voor de dubbelslinger introduceren we (massaloze) hulppunten op de scharnieren zodat we de translatiesnelheden gemakkelijk kunnen invoeren.

(30) $\text{POINTS}(Q, R, S)$

waarin Q (fig.1: $N^1 \equiv N^{1,2}$) en R (fig.1: $N^2 \equiv N^{2,2}$) de scharnierpunten zijn van lichaam A met de vaste wereld respectievelijk lichaam B met lichaam A. Het punt S is het eindpunt van lichaam B.

Omdat de punten massaloos zijn, moeten we dat aangeven met het commando MASSLESS volgens:

(31) $\text{MASSLESS}(Q, R, S)$

Nu kunnen alle snelheden worden bepaald. Met behulp van de geïntroduceerde (hulp)grootheden kunnen we de uitdrukking invoeren voor de snelheid van massamiddelpunt A^* .

(32) $\text{VASTARN} = \text{PHI1}' * L_A * A_2$

waarin VASTARN de translatie snelheid is van het massamiddelpunt A^* ten opzichte van de vaste wereld N en A_2 de basisvector is die de richting van de snelheid VASTARN aangeeft.

-> (33) $\text{VASTARN} = U_1 * L_A * A_2$

We kunnen zelf eenvoudig een uitdrukking invoeren voor de snelheid van massamiddelpunt B^* (zoals voor VASTAR), maar we kunnen dit ook laten formuleren door AUTOLEV . Hiervoor moeten we de positie van punt B^* differentiëren naar de tijd. Eerst bepalen we de positie van B^* m.b.v de hulppunten.

(34) $\text{PQASTAR} = L_A * A_1$

waarin P_QASTAR de vector is van Punt Q naar het punt ASTAR, uitgedrukt in de lichaamsgebonden basis A.

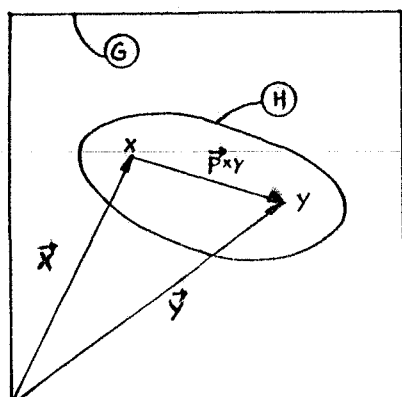
$$(35) \text{ PASTARR} = \text{LA} * \text{A1}$$

waarin P_ASTAR de vector is van Punt ASTAR naar het punt R, uitgedrukt in de basis A.

$$(36) \text{ PRBSTAR} = \text{LB} * \text{B1}$$

waarin P_RSTAR de vector is van Punt R naar het punt BSTAR, uitgedrukt in de lichaamsgebonden basis B.

We zullen aan de hand van figuur 4 een uitdrukking afleiden voor de snelheid van een punt ten opzichte van een ander punt in een frame G, als beide punten vast verbonden zijn met een star frame H.



figuur 4

We definiëren:

\vec{g} en \vec{h} als lichaamsgebonden bases bij lichaam G respectievelijk H

$$\vec{p}^{xy} = \vec{y} - \vec{x}$$

met x en y punten op het starre lichaam H. Er geldt:

$$\vec{x} = \vec{h}^T H \underline{x} \quad \text{en voor } y: \quad \vec{y} = \vec{h}^T H \underline{y}$$

Differentiëren naar de tijd levert:

$$\dot{\vec{x}} = \dot{\vec{h}}^T H \underline{x} + \vec{h}^T H \dot{\underline{x}} \quad \text{en} \quad \dot{\vec{y}} = \dot{\vec{h}}^T H \underline{y} + \vec{h}^T H \dot{\underline{y}}$$

\vec{p}^{xy} naar de tijd differentiëren levert nu:

$$\begin{aligned} \dot{\vec{p}}^{xy} &= \dot{\vec{y}} - \dot{\vec{x}} \\ &= \dot{\vec{h}}^T H \dot{\vec{y}} + \vec{h}^T H \dot{\vec{y}} - \dot{\vec{h}}^T H \dot{\vec{x}} - \vec{h}^T H \dot{\vec{x}} \\ &= \dot{\vec{h}}^T H \dot{\vec{p}}^{xy} + \vec{h}^T H \dot{\vec{p}}^{xy} \\ &= {}^G \vec{\omega}^H * \vec{p}^{xy} \end{aligned}$$

met ${}^H \dot{\vec{p}}^{xy} = 0$

De snelheid van punt y kunnen we dus berekenen met:

$${}^G \vec{v}^y = {}^G \vec{v}^x + {}^G \vec{\omega}^H * \vec{p}^{xy}$$

- ${}^G \vec{v}^y$ is de snelheid van punt y ten opzichte van frame G (= VYG)
- ${}^G \vec{v}^x$ is de snelheid van punt x ten opzichte van frame G (= VXG)
- ${}^G \vec{\omega}^H$ is de hoeksnelheid van frame H ten opzichte van frame G (= WGH)
- \vec{p}^{xy} is de plaatsvector van punt x naar y (= PXY)

AUTOLEV berekent op deze manier met het commando V2PTS(G,H,x,y) de snelheid VYG indien VXG, WGH en PXY bekend zijn.

VASTARN, WAN en PASTARR zijn bekend dus we kunnen het volgende commando geven:

(37) V2PTS(N,A,ASTAR,R)

waarin N de referentie basis is waarbinnen de beweging plaatsvindt, A het starre lichaam, ASTAR het massamiddelpunt en R het punt op het lichaam A waarvan de snelheid nu wordt bepaald.

-> (38) VRN=U1*(2*LA)*A2

waarin VRN de absolute snelheid is van het punt R ten opzichte van de vaste wereld N.

Nu kunnen we VBSTARN bepalen door V2PTS uit te rekenen voor het punt BSTAR ten opzichte van het punt R.

(39) V2PTS(N,B,R,BSTAR)

$$\rightarrow (40) \text{VBSTARN} = ((\dots)) * N1 + ((\dots)) * N2 + ((\dots)) * A2$$

$$(41) \text{EXPRESS}(\text{VBSTARN}, B)$$

$$\rightarrow (42) \text{VBSTARN} = ((\dots)) * B1 + ((\dots)) * B2$$

We geven de uitdrukkingen voor VBSTARN niet helemaal weer vanwege hun lengte. Volgens formule (f39) met $i=1$ en $j=2$ geldt:

$$(f39) \vec{v}^2 = \vec{v}^1 + \vec{\omega}^1 * (\vec{b}^{1,2} + \vec{c}^2) + (\vec{e}^1)^T \dot{\vec{c}}^2 - \vec{\omega}^2 * \vec{b}^{2,2}$$

- $\vec{v}^1 = \text{VASTARN} = \text{LA} * \text{U1} * \text{A2}$
- $\vec{\omega}^1 = \text{WAN} = \text{U1} * \text{N3} = \text{U1} * \text{A3}$
- $\vec{b}^{1,2} = \text{PASTARR} = \text{LA} * \text{A1}$
- $\vec{c}^2 = {}^1\vec{c}^2 = \vec{0}$
- $\vec{\omega}^2 = \text{WBN} = (\text{U1} + \text{U2}) * \text{N3} = (\text{U1} + \text{U2}) * \text{B3}$
- $\vec{b}^{2,2} = -\text{LB} * \text{B1}$

Uitwerken volgens (f39) levert (40).

1.7 Versnellingen

Voor de kolompresentatie van de absolute versnellingen ${}^0\dot{a}^k$ ten opzichte van de \vec{e}^0 -basis volgt:

$$(f44) {}^0\dot{a}^k = \dot{v}^k = \dot{Y}^k y + Y^k \dot{y}$$

De versnellingen die gelden voor de massamiddelpunten ASTAR en BSTAR worden gevonden door de translatiesnelheden VASTARN respectievelijk VBSTARN naar de tijd te differentiëren. Met AUTOLEV wordt dit weer gedaan met het commando DERIV.

$$(43) \text{AASTARN} = \text{DERIV}(\text{VASTARN}, T, N)$$

$$\rightarrow (44) \text{AASTARN} = -\text{COS}(\text{PHI1}) * \text{LA} * \text{U1} * \text{U1} * \text{N1} - \text{LA} * \text{SIN}(\text{PHI1}) * \text{U1} * \text{U1} * \text{N2} + \text{LA} * \text{U1}' * \text{A2}$$

$$(45) \text{ABSTARN} = \text{DERIV}(\text{VBSTARN}, T, N)$$

$$\rightarrow (46) \text{ABSTARN} = ((\dots))$$

2 De bewegingsvergelijkingen

2.1 Het opstellen van de bewegingsvergelijkingen

In het voorgaande zijn we gekomen tot een beschrijving van de representaties van de absolute snelheden en versnellingen ten opzichte van de $\bar{\varepsilon}^0$ -basis als functie van de gegeneraliseerde snelheden en hun tijdsafgeleiden volgens:

$$(f28) \quad {}^0\omega^k = \underline{Z}^k \dot{y}$$

waarin de matrix \underline{Z}^k de partiële hoeksnelheden bevat.

$$(f30) \quad {}^0\alpha^k = {}^0\dot{\omega}^k = \dot{\underline{Z}}^k \dot{y} + \underline{Z}^k \ddot{y}$$

$$(f42) \quad {}^0v^k = \underline{Y}^k \dot{y} \quad \text{met: } {}^0v^k = \bar{\varepsilon}^0 \bar{v}^k$$

waarin de matrix \underline{Y}^k de partiële translatiesnelheden bevat.

$$(f44) \quad {}^0a^k = \dot{v}^k = \dot{\underline{Y}}^k \dot{y} + \underline{Y}^k \ddot{y}$$

We gaan bovenstaande vergelijkingen in de bewegingsvergelijkingen volgens Newton en Euler gebruiken.

$$(f45) \quad \text{Newton: } \vec{F}^k = m^k \vec{a}^k$$

$$(f46) \quad \text{Euler: } \vec{M}^k = \mathbf{J}^k \cdot \vec{\alpha}^k + \vec{\omega}^k * (\mathbf{J}^k \cdot \vec{\omega}^k)$$

met:

- $\vec{a}^k = (\bar{\varepsilon}^0)^T {}^0a^k$ de absolute versnelling van M^k
- m^k de massa van B^k
- \mathbf{J}^k de centrale traagheidstensor van B^k
- $\vec{\alpha}^k = (\bar{\varepsilon}^0)^T {}^0\alpha^k$ de absolute hoekversnelling van B^k
- $\vec{\omega}^k = (\bar{\varepsilon}^0)^T {}^0\omega^k$ de absolute hoeksnelheid van B^k

De matrixrepresentatie van (f45) en (f46) t.o.v de \bar{e}^0 -basis is:

$$(f47) \quad {}^0F^k = m^k {}^0a^k$$

$$(f48) \quad {}^0M^k = {}^0J^k {}^0a^k + \underline{\Omega}^k ({}^0J^k {}^0\omega^k)$$

De in totaal $6n$ kolommen in:

$$(f49) \quad \underline{Z}^k = [z^{k1} \quad z^{k2} \quad \dots \quad z^{k6n}]$$

respectievelijk:

$$(f50) \quad \underline{Y}^k = [y^{k1} \quad y^{k2} \quad \dots \quad y^{k6n}]$$

zijn op te vatten als de matrixrepresentatie van de partiële hoeksnelheidsvectoren $\bar{z}^{ki} = (\bar{e}^0)^T \underline{z}^{ki}$ en de partiële translatiesnelheidsvectoren $\bar{v}^{ki} = (\bar{e}^0)^T \underline{v}^{ki}$ (voor $i=1, \dots, 6n$; $k=1, \dots, n$) ten opzichte van de basis \bar{e}^0 .

Vermenigvuldigen we (f47) en (f48) vóór met $(\underline{Z}^k)^T$ respectievelijk $(\underline{Y}^k)^T$, dan volgt hieruit:

$$(f51) \quad m^k (\underline{Y}^k)^T {}^0a^k = (\underline{Y}^k)^T {}^0F^k$$

$$(f52) \quad (\underline{Z}^k)^T {}^0J^k {}^0a^k = (\underline{Z}^k)^T {}^0M^k - (\underline{Z}^k)^T \underline{\Omega}^k {}^0J^k {}^0\omega^k$$

Substitutie van (f28), (f30) en (f44) in (f51) en (f52) en het optellen van de zo verkregen uitdrukkingen levert dan:

$$(f53) \quad m^k (\underline{Y}^k)^T (\dot{\underline{Y}}^k \underline{y} + \underline{Y}^k \dot{\underline{y}}) + (\underline{Z}^k)^T {}^0J^k (\dot{\underline{Z}}^k \underline{y} + \underline{Z}^k \dot{\underline{y}}) = (\underline{Y}^k)^T {}^0F^k + (\underline{Z}^k)^T {}^0M^k - (\underline{Z}^k)^T \underline{\Omega}^k {}^0J^k \dot{\underline{Z}}^k \underline{y}$$

Het overbrengen van de termen met $\dot{\underline{y}}$ naar het linker- en \underline{y} naar het rechterlid en sommeren over alle lichamen geeft:

$$(f54) \quad \sum_{k=1}^n \{ m^k (\underline{Y}^k)^T \dot{\underline{Y}}^k + (\underline{Z}^k)^T {}^0J^k \dot{\underline{Z}}^k \} \dot{\underline{y}} = \sum_{k=1}^n \{ (\underline{Y}^k)^T {}^0F^k + (\underline{Z}^k)^T {}^0M^k \} - \sum_{k=1}^n \{ m^k (\underline{Y}^k)^T \dot{\underline{Y}}^k + (\underline{Z}^k)^T {}^0J^k \dot{\underline{Z}}^k + (\underline{Z}^k)^T \underline{\Omega}^k {}^0J^k \dot{\underline{Z}}^k \} \underline{y}$$

of:

$$(f55) \quad \underline{A} \dot{\underline{y}} = \underline{f}$$

2.2 De bewegingsvergelijkingen volgens Kane

De vorm waarin we de wetten van Newton en Euler hebben gepresenteerd, is uitgewerkt door de Amerikaanse dynamicus Kane. Het programma AUTOLEV is mede door hem ontwikkeld en maakt gebruik van de voorafgaande theorie. Er zijn echter een klein aantal verschillen tussen onze bewegingsvergelijkingen en die volgens Kane.

Kane maakt onderscheid tussen *opgelegde* krachten en momenten en *schijnkrachten* en -momenten. Alle krachten en momenten worden voorvermenigvuldigd met $(\underline{Z}^k)^T$ respectievelijk $(\underline{Y}^k)^T$ volgens:

$$(f56) \underline{F} = \sum_{k=1}^n \{(\underline{Y}^k)^T {}^0 \underline{F}^k + (\underline{Z}^k)^T {}^0 \underline{M}^k\}$$

voor de opgelegde krachten en momenten,

$$(f57) \underline{F}^* = \sum_{k=1}^n \{(\underline{Y}^k)^T {}^0 \underline{F}^{k*} + (\underline{Z}^k)^T {}^0 \underline{M}^{k*}\}$$

voor de schijnkrachten en -momenten. Voor de laatste vergelijking geldt:

$$(f58) {}^0 \underline{F}^{k*} = -m^k {}^0 \underline{g}^k$$

en

$$(f59) {}^0 \underline{M}^{k*} = -{}^0 \underline{J}^k {}^0 \underline{\alpha}^k - {}^0 \underline{\Omega}^k {}^0 \underline{J}^k {}^0 \underline{\omega}^k$$

De bewegingsvergelijkingen volgens Kane zien er dan als volgt uit:

$$(f60) \underline{F} + \underline{F}^* = 0$$

Na substitutie van (f58), (f59), (f56), (f57) en (f28), (f30) en (f44) in (f60) komt men weer uit op (f54). Dit zijn n eerste orde differentiaalvergelijkingen (met n het aantal graden van vrijheid) die numeriek kunnen worden opgelost.

Na elke tijdstap, als de gegeneraliseerde snelheden zijn uitgerekend, worden meteen de hoeken Φ_1 en Φ_2 meegeïntegreerd.

In ons geval is het mogelijk om op deze manier het verloop van de hoeken als functie van de tijd te berekenen. Er is namelijk sprake van een vlakke beweging, zodat de oriëntatie met de twee gekozen hoeken eenduidig vastligt.

2.3 Eulerparameters

Voor het beschrijven van de oriëntatie van een lichaam in de ruimte, zijn in principe drie parameters nodig. In sommige situaties is de eenduidig-

heid van de beschrijving niet meer gewaarborgd. Met de introductie van vier *Eulerparameters* kan dit niet gebeuren. De keuze van deze parameters zijn gebaseerd op het theorema van Euler:

Twee willekeurig geöriënteerde orthonormale vectorbases \vec{e}^{k-1} en \vec{e}^k met dezelfde oorsprong O kan men laten samenvallen door de dragers van de vectoren van één der bases over een bepaalde hoek, θ_k , om een bepaalde as door O met eenheidsrichtingsvector $\vec{\lambda}_k$ te roteren (zie literatuur [4], hoofdstuk 1.2.4).

Met andere woorden: er is altijd een as door een punt O aan te wijzen waar men een basis \vec{e}^k over een hoek θ_k om kan laten draaien, zodat basis \vec{e}^k op basis \vec{e}^{k-1} valt. De Eulerparameters definiëren we als volgt (literatuur [1]):

$$(f61) \quad \varepsilon_{ki} = \lambda_{ki} \sin(\theta_k/2) \quad i = 1, 2, 3, \quad \varepsilon_{k4} = \cos(\theta_k/2)$$

met: λ_{ki} de componenten van $\vec{\lambda}_k$ ten opzichte van een willekeurige basis. De Eulerparameters zijn onderling afhankelijk volgens:

$$(f62) \quad \varepsilon_{k1}^2 + \varepsilon_{k2}^2 + \varepsilon_{k3}^2 + \varepsilon_{k4}^2 = 1$$

Op de volgende manier kunnen we de rotatiematrix (overeenkomend met (f8)) met Eulerparameters opbouwen:

$$(f63) \quad \underline{R}^k =$$

$$\begin{bmatrix} (\varepsilon_{k1}^2 - \varepsilon_{k2}^2 - \varepsilon_{k3}^2 + \varepsilon_{k4}^2) & 2(\varepsilon_{k1}\varepsilon_{k2} - \varepsilon_{k3}\varepsilon_{k4}) & 2(\varepsilon_{k1}\varepsilon_{k3} + \varepsilon_{k2}\varepsilon_{k4}) \\ 2(\varepsilon_{k1}\varepsilon_{k2} + \varepsilon_{k3}\varepsilon_{k4}) & (-\varepsilon_{k1}^2 + \varepsilon_{k2}^2 - \varepsilon_{k3}^2 + \varepsilon_{k4}^2) & 2(\varepsilon_{k2}\varepsilon_{k3} - \varepsilon_{k1}\varepsilon_{k4}) \\ 2(\varepsilon_{k1}\varepsilon_{k3} - \varepsilon_{k2}\varepsilon_{k4}) & 2(\varepsilon_{k2}\varepsilon_{k3} + \varepsilon_{k1}\varepsilon_{k4}) & (-\varepsilon_{k1}^2 - \varepsilon_{k2}^2 + \varepsilon_{k3}^2 + \varepsilon_{k4}^2) \end{bmatrix}$$

De tijdsafgeleiden van de Eulerparameters zijn lineair afhankelijk van de relatieve hoeksnelheden volgens:

$$(f64) \quad \begin{aligned} \dot{\varepsilon}_{k1} &= \frac{1}{2}(\varepsilon_{k4}\dot{\omega}_{k1} + \varepsilon_{k3}\dot{\omega}_{k2} - \varepsilon_{k2}\dot{\omega}_{k3}) \\ \dot{\varepsilon}_{k2} &= \frac{1}{2}(-\varepsilon_{k3}\dot{\omega}_{k1} + \varepsilon_{k4}\dot{\omega}_{k2} + \varepsilon_{k1}\dot{\omega}_{k3}) \\ \dot{\varepsilon}_{k3} &= \frac{1}{2}(\varepsilon_{k2}\dot{\omega}_{k1} - \varepsilon_{k1}\dot{\omega}_{k2} + \varepsilon_{k4}\dot{\omega}_{k3}) \\ \dot{\varepsilon}_{k4} &= \frac{1}{2}(-\varepsilon_{k1}\dot{\omega}_{k1} - \varepsilon_{k2}\dot{\omega}_{k2} - \varepsilon_{k3}\dot{\omega}_{k3}) \end{aligned}$$

Zoals hierboven duidelijk te zien is, zijn de afgeleiden van de Eulerparameters lineair in de relatieve hoeksnelheden. Aangezien AUTOLEV hiervan toch al gebruik maakt in de vorm van gegeneraliseerde snelheden, zijn de Eulerparameters met weinig extra moeite uit te rekenen.

2.4 De functie AUTOZ

In het begin van dit verslag hebben we reeds de functie AUTOZ behandeld. Het was overzichtelijker om deze functie uit te zetten omdat er dan geen gebruik van hulpvergelijkingen werd gemaakt. Op een gegeven moment worden de vergelijkingen echter te lang, zodat zij niet meer in het geheugen passen (AUTOLEV geeft dan een melding: Error, out of string space).

We voeren de commando's opnieuw in (zie appendix E), nu met de functie AUTOZ aan. Bekijken we de uitvoer dan zien we dat we nu 30 regels meer hebben dan voorheen. Alle commando's met uitvoer t/m het commando ABSTARN=DERIV(...) staan hieronder nog een keer weergegeven.

```

(1) ! DUBBELSLINGER
(2) !
(3) AUTOZ(ON)
(4) DOF(2)
(5) FRAMES(A,B)
(6) SIMPROT(N,A,3,PHI1)
-> (7) DIRCOS(N,A,COS(PHI1),-SIN(PHI1),0,SIN(PHI1),
    COS(PHI1),0,0,0,1)
(8) SIMPROT(A,B,3,PHI2)
-> (9) DIRCOS(A,B,COS(PHI2),-SIN(PHI2),0,SIN(PHI2),
    COS(PHI2),0,0,0,1)
(10) DIRCOS(N,B)
-> (11) DIRCOS(N,B,COS(PHI1+PHI2),-SIN(PHI1+PHI2),
    0,SIN(PHI1+PHI2),COS(PHI1+PHI2),0,0,0,1)
(12) PHI1'=U1
(13) PHI2'=U2
(14) WAN=PHI1'*N3
-> (15) WAN=U1*N3
(16) WBA=PHI2'*A3
-> (17) WBA=U2*A3
(18) WBN=ADD(WAN,WBA)
-> (19) WBN=U1*N3+U2*A3
(20) EXPRESS(WBN,N)
-> (21) WBN=(U1+U2)*N3
(22) ALFAN=DERIV(WAN,T,N)
-> (23) ALFAN=U1'*N3
(24) ALFBN=DERIV(WBN,T,N)
-> (25) ALFBN=(U1'+U2')*N3
(26) CONST(LA,LB)
(27) POINTS(Q,R,S)

```

```

(28) MASSLESS(Q,R,S)
(29) VASTARN=PHI1*LA*A2
-> (30) VASTARN=U1*LA*A2
(31) PQASTAR=LA*A1
(32) PASTARR=LA*A1
(33) PRBSTAR=LB*B1
(34) V2PTS(N,A,ASTAR,R)
-> (35) Z1=LA*SIN(PHI1)
-> (36) Z2=COS(PHI1)*LA
-> (37) VRN=-U1*Z1*N1+U1*Z2*N2+LA*U1*A2
(38) V2PTS(N,B,R,BSTAR)
-> (39) Z3=LB*SIN(PHI1+PHI2)+Z1
-> (40) Z4=LB*SIN(PHI1+PHI2)
-> (41) Z5=COS(PHI1+PHI2)*LB+Z2
-> (42) Z6=COS(PHI1+PHI2)*LB
-> (43) VBSTARN=(-U1*Z3-U2*Z4)*N1+(U1*Z5+U2*Z6)*N2+
    LA*U1*A2
(44) EXPRESS(VBSTARN,B)
-> (45) Z7=COS(PHI1+PHI2)*Z3-LA*SIN(PHI2)-SIN(PHI1+
    PHI2)*Z5
-> (46) Z8=COS(PHI1+PHI2)*Z4-SIN(PHI1+PHI2)*Z6
-> (47) Z9=COS(PHI1+PHI2)*Z5+COS(PHI2)*LA+
    SIN(PHI1+PHI2)*Z3
-> (48) Z10=COS(PHI1+PHI2)*Z6+SIN(PHI1+PHI2)*Z4
-> (49) VBSTARN=(-U1*Z7-U2*Z8)*B1+(U1*Z9+U2*Z10)*B2
(50) AASTARN=DERIV(VASTARN,T,N)
-> (51) Z11=LA*U1
-> (52) VASTARN=Z11*A2
-> (53) Z12=COS(PHI1)*U1*Z11
-> (54) Z13=SIN(PHI1)*U1*Z11
-> (55) AASTARN=-Z12*N1-Z13*N2+LA*U1*A2
(56) ABSTARN=DERIV(VBSTARN,T,N)
-> (57) Z14=COS(PHI1)*LA*U1
-> (58) Z15=LA*SIN(PHI1)*U1
-> (59) Z16=(U1+U2)*COS(PHI1+PHI2)*LB+Z14
-> (60) Z17=(U1+U2)*COS(PHI1+PHI2)*LB
-> (61) Z18=(U1+U2)*LB*SIN(PHI1+PHI2)+Z15
-> (62) Z19=(U1+U2)*LB*SIN(PHI1+PHI2)
-> (63) Z20=(U1+U2)*COS(PHI1+PHI2)*Z5+(U1+U2)*
    SIN(PHI1+PHI2)*Z3-COS(PHI1+PHI2)*Z16+
    COS(PHI2)*LA*U2- SIN(PHI1+PHI2)*Z18
-> (64) Z21=(U1+U2)*COS(PHI1+PHI2)*Z6+(U1+U2)*
    SIN(PHI1+PHI2)*Z4-COS(PHI1+PHI2)*Z17-
    SIN(PHI1+PHI2)*Z19

```

```

-> (65) Z22=(U1+U2)*COS(PHI1+PHI2)*Z3-
      (U1+U2)*SIN(PHI1+PHI2)*Z5-COS(PHI1+PHI2)*Z18-
      LA*SIN(PHI2)*U2+SIN(PHI1+PHI2)*Z16
-> (66) Z23=(U1+U2)*COS(PHI1+PHI2)*Z4-(U1+U2)*
      SIN(PHI1+PHI2)*Z6-COS(PHI1+PHI2)*Z19+
      SIN(PHI1+PHI2)*Z17
-> (67) Z24=U1*Z7+U2*Z8
-> (68) Z25=U1*Z9+U2*Z10
-> (69) VBSTAR=-Z24*B1+Z25*B2
-> (70) Z26=U1+U2
-> (71) WBN=Z26*N3
-> (72) Z27=(COS(PHI1+PHI2)*Z25-SIN(PHI1+PHI2)*Z24)*Z26
-> (73) Z28=(-COS(PHI1+PHI2)*Z24-SIN(PHI1+PHI2)*Z25)*Z26
-> (74) Z29=U1*Z20+U2*Z21
-> (75) Z30=U1*Z22+U2*Z23
-> (76) ABSTAR=-Z27*N1+Z28*N2+(-U1'*Z7-U2'*Z8+
      Z29)*B1+(U1'*Z9+U2'*Z10+Z30)*B2

```

2.5 Het genereren van de bewegingsvergelijkingen

We gaan nu verder met de formulering van de bewegingsvergelijkingen volgens Euler en Newton.

De absolute versnelling, hoekversnelling en hoeksnelheid van de massamiddelpunten ASTAR en BSTAR zijn inmiddels uitgerekend. Om de vergelijking voor ${}^0M^{**}$ volgens (f59) te berekenen, moeten we de matrixrepresentatie van de traagheidstensor van lichaam A en B met de componenten $IA(i,j)$ respectievelijk $IB(i,j)$ ten opzichte van de basis A respectievelijk B, met i en $j \in [1,2,3]$, beschrijven. $IA(i,j)$ en $IB(i,j)$ zijn met het commando (5): `FRAMES(A,B)` als 'syntactical forms' door AUTOLEV zelf bij de lichamen/frames A en B geïntroduceerd. Omdat we de lokale assenstelsels A1, A2, A3 en B1, B2, B3 zodanig hebben gekozen dat de werklijnen van de basisvectoren hoofdtraagheidsassen zijn, is $IA(i,j)$ en $IB(i,j)$ alleen voor $i=j$ ongelijk aan nul. Dit kunnen we binnen AUTOLEV aangeven met het commando:

```
(77) PRINCIPAL(A,B)
```

De overgebleven termen $IA(i,i)$ en $IB(i,i)$ met $i=1,2,3$ worden vervangen door $IA(i)$ respectievelijk $IB(i)$ met $i=1,2,3$. Deze worden later bij het specificeren van het systeem ingevoerd.

De gegeneraliseerde traagheidskrachten F^* volgens (f57) kunnen dan worden opgesteld volgens formule (f47) met het commando:

(78) FRSTAR

Dit levert twee vergelijkingen op die voldoen aan (f59) vermenigvuldigd met de \tilde{e}^0 -basis, nl:

$$\rightarrow (79) TSTARA = -IA(3) * U1' * A3$$

$$\rightarrow (80) TSTARB = (-IB(3) * U1' - IB(3) * U2') * B3$$

Verder levert (78) met de introductie van een nieuwe reeks Z's vergelijkingen op voor F1STAR:

$$\rightarrow (81) Z31 = (-Z7 * Z7 - Z9 * Z9) * MASSB - IA(3) - IB(3) - LA * LA * MASSA$$

$$\rightarrow (82) Z32 = (-Z10 * Z9 - Z7 * Z8) * MASSB - IB(3)$$

$$\rightarrow (83) Z33 = (-\cos(\text{PHI1}) * LA * Z13 + LA * \sin(\text{PHI1}) * Z12) * MASSA + (\cos(\text{PHI1} + \text{PHI2}) * Z27 * Z7 + \cos(\text{PHI1} + \text{PHI2}) * Z28 * Z9 + \sin(\text{PHI1} + \text{PHI2}) * Z27 * Z9 - \sin(\text{PHI1} + \text{PHI2}) * Z28 * Z7 - Z29 * Z7 + Z30 * Z9) * MASSB$$

$$\rightarrow (84) F1STAR = Z31 * U1' + Z32 * U2' - Z33$$

en F2STAR:

$$\rightarrow (85) Z34 = (-Z10 * Z10 - Z8 * Z8) * MASSB - IB(3)$$

$$(86) Z35 = (\cos(\text{PHI1} + \text{PHI2}) * Z10 * Z28 + \cos(\text{PHI1} + \text{PHI2}) * Z27 * Z8 + \sin(\text{PHI1} + \text{PHI2}) * Z10 * Z27 - \sin(\text{PHI1} + \text{PHI2}) * Z28 * Z8 + Z10 * Z30 - Z29 * Z8) * MASSB$$

$$\rightarrow (87) F2STAR = Z32 * U1' + Z34 * U2' - Z35$$

die voldoen aan (f58) vermenigvuldigd met de \tilde{e}^0 -basis.

Gegeneraliseerde opgelegde krachten

We willen tijdens de simulatie krachten en momenten opleggen, voortkomend uit demping en zwaartekracht op het systeem. Hiervoor zijn de commando's FORCE en TORQUE beschikbaar. AUTOLEV kent hiervoor de volgende invoer:

- FORCE(<>)- uitwendige kracht uitgeoefend op het punt <>
- TORQUE(<>)- uitwendig koppel uitgeoefend op het frame <>

- FORCE(<> /{ }) - kracht uitgeoefend door punt <> op het punt { }
- TORQUE(<> /{ }) - koppel uitgeoefend door frame <> op frame { }

Het scharnier bij punt Q heeft een dempingskoppel TA rond N3 of A3, het scharnier bij punt R heeft een dempingskoppel TAB, dus:

$$(88) \text{ TORQUE}(N/A) = TA * A3$$

$$(89) \text{ TORQUE}(A/B) = TAB * A3$$

Met behulp van het commando CONTROLS(C1,C2,...,Cn) kunnen de parameters worden aangegeven die worden gebruikt bij de beschrijving van de krachten en koppels (met de commando's FORCE resp TORQUE) en die een functie van de kinematische variabelen zijn.

$$(90) \text{ CONTROLS}(TA, TAB)$$

Om de demping te beschrijven voeren we verder nog twee constanten in nl:

$$(91) \text{ CONST}(WDA, WDAB)$$

Voor de dempingskoppels TA en TAB geldt dan:

$$(92) TA = -WDA * PHI1'$$

$$\rightarrow (93) TA = -WDA * U1$$

$$(94) TAB = -WDAB * PHI2'$$

$$\rightarrow (95) TAB = -WDAB * U2$$

Ons systeem is onderhevig aan het zwaartekrachtveld dat een constante kracht in de richting N1 geeft. De valversnelling \vec{g} voeren we in als constante G en op massamiddelpunten ASTAR en BSTAR werken dan de volgende uitwendige krachten:

$$(96) \text{ CONST}(G)$$

$$(97) \text{ FORCE(ASTAR)} = G * \text{MASSA} * N1$$

$$(98) \text{ FORCE(BSTAR)} = G * \text{MASSB} * N1$$

Omdat de zwaartekracht \vec{g} en de massa's toch constant zijn (dat wil zeggen geen functie van de kinematische variabelen), hoeven we deze niet aan te duiden in het CONTROLS-commando. Nadat alle opgelegde krachten en momenten zijn beschreven, gebeurt het opstellen van de vergelijking volgens (f56) eenvoudig met het geven van het commando:

$$(99) \text{ FR}$$

Dit leidt dan tot twee vergelijkingen (vergelijk dit met de uitvoer van het commando (78) FRSTAR) voor F1 en F2:

$$\rightarrow (100) F1 = -\text{COS}(\text{PHI1} + \text{PHI2}) * G * \text{MASSB} * Z7 - G * \text{LA} * \text{MASSA} * \text{SIN}(\text{PHI1}) - G * \text{MASSB} * \text{SIN}(\text{PHI1} + \text{PHI2}) * Z9 + \text{TA}$$

$$\rightarrow (101) F2 = -G * \text{MASSB} * Z4 + \text{TAB}$$

F1 en F2 hebben natuurlijk dezelfde eenheden als die van F1STAR en F2STAR.

De twee bewegingsvergelijkingen volgens Kane (f58) waaraan de twee lichamen A en B moeten voldoen worden nu opgesteld met het volgende commando:

$$(102) \text{ KANE}$$

Met als responsie:

$$\rightarrow (103) R = 1: Z31 * U1' + Z32 * U2' - Z33 - \text{COS}(\text{PHI1} + \text{PHI2}) * G * \text{MASSB} * Z7 - G * \text{LA} * \text{MASSA} * \text{SIN}(\text{PHI1}) - G * \text{MASSB} * \text{SIN}(\text{PHI1} + \text{PHI2}) * Z9 + \text{TA} = 0$$

$$\rightarrow (104) R = 2: Z32 * U1' + Z34 * U2' - Z35 - G * \text{MASSB} * Z4 + \text{TAB} = 0$$

2.6 Controle van de invoer

Het systeem is nu volledig beschreven maar we zouden graag een controle willen hebben ten aanzien van de juistheid van de systeembeschrijving. Hiervoor heeft AUTOLEV twee mogelijkheden, waarvan

één voor ons van toepassing is. De eerste betreft het opstellen van vergelijkingen voor het impulsmoment dat voor sommige systemen constant moet zijn. Dit is voor ons systeem niet het geval. De tweede mogelijkheid betreft het opstellen van een vergelijking voor de kinetische, potentële en totale energie, die in ons geval een voorspelbaar verloop heeft. Zo zal bij ons systeem zonder demping de totale energie constant moeten blijven en, ook in het geval met demping, een (plaatselijk) maximum voor de potentiële energie moeten worden bereikt. De afgeleide van de totale energie moet bovendien gelijk aan nul zijn als de kinetische energie gelijk aan nul is. Door de parameters te veranderen kan dit voor verschillende gevallen worden bekeken. Het gebruik van deze mogelijkheid leidt tot kolommen met getallen in de uitvoerfiles die we met behulp van Matlab grafisch zullen weergeven.

AUTOLEV rekt zelf met het commando KE (Kinetic Energy) de kinetische energie uit:

(105) KE

Dit leidt tot de vorming van ZKEi's, hulpvergelijkingen vergelijkbaar met de Zi's:

-> (106) $ZKE1 = IA(3) * U1 * U1 + MASSA * Z11 * Z11$
 -> (107) $ZKE2 = (IB(3) * U1 + IB(3) * U2) * (U1 + U2) + (Z24 * Z24 + Z25 * Z25) * MASSB$
 -> (108) $KE0 = 0$

De kinetische energie waarin de gegeneraliseerde snelheden niet voorkomen.

-> (109) $KE1 = 0$

De kinetische energie waarin de gegeneraliseerde snelheden in de eerste macht voorkomen

(110) $KE2 = .5 * (ZKE1 + ZKE2)$

De kinetische energie waarin de gegeneraliseerde snelheden in het kwadraat voorkomen.

(111) $KE = KE0 + KE1 + KE2$

is dan de totale kinetische energie.

De potentiële energie PE (met PE een syntactical form) moeten we zelf invoeren. Uit figuur 2 blijkt dat deze te schrijven is als:

$$(112) \text{ PE} = (\text{MASSA} + 2 * \text{MASSB}) * (1 - \text{COS}(\text{PHI1})) * \text{G} * \text{LA} + \text{MASSB} * \text{G} * (1 - \text{COS}(\text{PHI1} + \text{PHI2})) * \text{LB}$$

Met deze commando's zal in een van de uitvoerfiles het verloop van de totale, de potentiële en de kinetische energie als functie van de tijd worden opgeslagen.

Met het commando UNITS({},[]) vertellen we aan AUTOLEV dat de variabele {} de eenheid [] heeft. Dit kunnen we voor een aantal variabelen tegelijk doen zodat we in onze situatie de volgende commando's kunnen geven:

- (113) UNITS(T,S,U1,RAD/S,U2,RAD/S,G,M/S^2,WDA,N*M*S)
- (114) UNITS(PHI1,RAD,PHI2,RAD,WDAB,N*M*S)
- (115) UNITS(TA,N*M,TB,N*M,TAB,N*M,FGA,N,FGB,N)
- (116) UNITS(MASS,KG,INERTIA,KG*M^2,ENERGY,N*M)
- (117) UNITS(LA,M,LB,M)

Met het commando UNITS kan worden bekeken of er nog grootheden zijn die geen eenheid hebben gekregen. Het is niet verplicht om alles een eenheid te geven, maar de uitvoer wordt op deze manier voorzien van de juiste eenheden.

2.7 Gebruik van de Eulerparameters

We zullen eerst aantonen dat voor ons geval (f63) overeenkomt met (7). De as $\vec{\lambda}_k$ is gedurende de hele slingerbeweging de vector N3. Dit betekent dat de eulerparameters ε_{k1} en ε_{k2} gelijk aan nul blijven. De derde eulerparameter ε_{k3} is dan gelijk aan $\sin(\frac{\alpha}{2})$ en de vierde $\cos(\frac{\alpha}{2})$. Werken we dit uit voor (f63) met $\cos(\alpha) = 1 - 2 \sin^2(\frac{\alpha}{2}) = 2 \cos^2(\frac{\alpha}{2}) - 1$ en $2 \sin(\frac{\alpha}{2}) \cos(\frac{\alpha}{2}) = \sin(\alpha)$ dan gaat (f63) over in (f8). De formules volgens (f64) volgen door de eulerparameters te differentiëren naar de tijd. Daarbij zijn $\hat{\omega}_{k1} = \hat{\omega}_{k2} = 0$ en is $\hat{\omega}_{k3} = \text{WAN} = U1$.

In ons geval is het niet nodig Eulerparameters te gebruiken, omdat er sprake is van een vlakke beweging, opgelegd door de lijnscharnieren. De oriëntatie van het systeem berekent AUTOLEV door de hoeksnelheden U1 en U2 mee te integreren met (f60). Om het gebruik ervan

toch duidelijk te maken, geven we het volgende voorbeeld. Bewaren we de ingevoerde commando's in AUTOLEV (appendix E), dan kunnen we de regel SIMPROT(N,A) weghalen en in plaats hiervan geven we het commando:

DIRCOS(N,A,EULERP)

met als respons van AUTOLEV

Z1=1-2*(EAN2^2+EAN3^2)
 Z2=2*(EAN1*EAN2-EAN3*EAN4)
 Z3=2*(EAN3*EAN1+EAN2*EAN4)
 Z4=2*(EAN1*EAN2+EAN3*EAN4)
 Z5=1-2*(EAN3^2+EAN1^2)
 Z6=2*(EAN2*EAN3-EAN1*EAN4)
 Z7=2*(EAN3*EAN1-EAN2*EAN4)
 Z8=2*(EAN2*EAN3+EAN1*EAN4)
 Z9=1-2*(EAN1^2+EAN2^2)

DIRCOS(N,A,Z1,Z2,Z3,Z4,Z5,Z6,Z7,Z8,Z9)

Omdat we de in- en uitvoer van de rest van de commando's toch niet geven zijn de regelnummers weggelaten.

In de hulpvergelijkingen wordt de syntactical form EAN_i gebruikt, overeenkomend met de Eulerparameters ε_{ki} volgens (f61) waarbij k betrekking heeft op vectorbasis A. Door substitutie van (f62) in (f63) komen we op bovenstaande uitdrukking voor DIRCOS(N,B,EULERP).

Geven we ná het invoeren van WAN het commando:

EPDOTS(N,A)

dan antwoord AUTOLEV met:

Z16=U1*Z7
 Z17=U1*Z8
 Z18=U1*Z9
 EAN1'=0.5*(Z16*EAN4-Z17*EAN3+Z18*EAN2)
 EAN2'=0.5*(Z16*EAN3+Z17*EAN4-Z18*EAN1)
 EAN3'=0.5*(-Z16*EAN2+Z17*EAN1+Z18*EAN4)
 EAN4'=0.5*(-Z16*EAN1-Z17*EAN2-Z18*EAN3)

overeenkomend met (f64).

Laten we de rest van de file in dezelfde vorm, dan zien we twee veranderingen in de Fortran-file (de uitdraai wordt hiervan niet gegeven):

- De Eulerparameters EANi' worden omgezet naar U(i+2) met i=1,2,3,4, hetgeen betekent dat ze volgens (f60) mee worden geïntegreerd.
- Er wordt in het readstatement gevraagd naar de beginwaarden van EAN(i) met i=1,2,3,4, waarvoor we met de hoek $\theta_k = \frac{\pi}{4}$ voor de beginsituatie invullen 0,0,.3827,.9239.

De simulatieresultaten komen dan overeen met de resultaten behaald met het simuleren zonder eulerparameters. De verschillen tussen de berekeningen van de hoek Φ_1 bijvoorbeeld zijn kleiner dan 10^{-4} .

3 Simulatie

3.1 Het genereren van een FORTRAN-programma

Om tot het simuleren van het systeem te komen schrijft AUTOLEV een FORTRAN-programma dat na 'compiling' en 'linking' omgezet kan worden in een 'executable file'. We geven het commando:

```
(118) CODE(SLINGER,SUBS)
```

AUTOLEV schrijft een Fortran-programma SLINGER.FOR met drie extra subroutines DECMP2, SOLVE2 en DEQS die het gevolg zijn van het argument SUBS in het commando CODE. Deze subroutines kunnen samen een stelsel differentiaalvergelijkingen volgens (f55) oplossen en zijn gebaseerd op de methode van KUTTA-MERSON met variable stapgrootte. Ze kunnen in een aparte bibliotheek worden geplaatst zodat bij latere simulaties het argument SUBS kan worden weggelaten, hetgeen schijfruimte bespaart (zie hiervoor een handleiding in Fortran).

Gaan we uit AUTOLEV dan kunnen we met een editor de file SLINGER.FOR bekijken (appendix C). De executable file SLINGER.EXE heeft een inputfile SLINGER.IN nodig die de begincondities en constanten aangeeft volgens het readstatement uit de file SLINGER.FOR:

```
READ(11,*) LA, LB, WDA, WDAB, G  
READ(11,*) MASSA, MASSB  
READ(11,*) IA(1), IA(2), IA(3)  
READ(11,*) IB(1), IB(2), IB(3)  
READ(11,*) U(1), U(2)  
READ(11,*) PHI1, PHI2
```

3.2 Invoer van de gegevens

Voor de waarden die worden gevraagd verzinnen we zelf enige reële getallen. We kiezen hiervoor waarden die overeenkomen met getallen die bij DADS zijn gebruikt voor de uitwerking van de dubbelslinger, alleen nemen we een wat langere tijd (deze wordt overigens pas gevraagd bij het laten draaien van het programma).

Overigens, de enige vergelijking die we zullen maken betreft de resultaten van de tweede simulatie. Reeds in het begin van het verslag is gezegd dat we twee simulaties gaan doen nl.:

Vrije beweging zonder demping

We maken dus een file SLINGER.IN die precies dezelfde structuur heeft als het readstatement in (3.1). De eenheden komen overeen met de eenheden in de commando's UNITS ((113) t/m (117)).

```
1,1,0,0,9.81
2,4
0,2.66667,2.66667
0,5.33333,5.33333
0,0
.7854,.7854
```

De hoeken Φ_1 en Φ_2 zijn beide op $\frac{\pi}{4}$ gesteld. De potentiële energie is op dat moment maximaal, de kinetische energie minimaal en de totale energie moet dus gelijk blijven aan de beginwaarde van de potentiële energie.

Als we SLINGER.FOR binnen DOS hebben gecompileerd en gelinkt, geven we binnen DOS het commando SLINGER. Er wordt ons gevraagd naar de totale tijd, de grootte van het tijdsinterval voor schrijven van gegevens en de initiële waarde van de integratietijdstap (deze wordt aangepast indien de berekende fout te groot wordt). Hiervoor voeren we in 10, .1 en .05.

De resultaten van de simulatie staan in de files SLINGER.OU1 etc. (dit wordt in de Fortran-file aangeduid). Door deze files te bewerken en binnen Matlab te laden (Appendix E) kunnen we hiervan grafieken maken, die in appendix D.2 staan weergegeven. De slinger blijkt eindeloos door te blijven slingeren en de totale energie blijft constant.

Vrije beweging met demping

```
1,1,28.64789,28.64789,9.81
2,4
0,0,2.66666667
0,0,5.33333333
0,0
.7854,.7854
```

Dezelfde beginvoorwaarden alleen nu met demping in de scharnieren. De totale energie neemt nu af, de slinger valt rond zijn evenwichtspunt ($\Phi_1 = \Phi_2 = 0$) stil. De grafieken hiervan staan in appendix D.3. De vergelijking van de resultaten met die van DADS wordt in appendix D.4 gemaakt, er zijn geen principiële verschillen wat betreft getalswaarden. Er is wel een verschil wat betreft de tekens. Dit hangt

samen met de keuze van de verbindingssystemen en de definitie van de potentiële energie. In ons geval wordt deze in de eindstand voor $\Phi_1 = \Phi_2 = 0$ nul gekozen. In het voorbeeld volgens DADS wordt deze voor $\Phi_1 = \pi$ en $\Phi_2 = 0$ nul gekozen.

4 Conclusies en aanbevelingen

De formulering van de bewegingsvergelijkingen volgens Kane is efficiënt opgezet en is uitermate geschikt voor numerieke verwerking op een computer. De naamgeving van de grootheden lijkt op het eerste gezicht vrij ingewikkeld, maar is consequent en volledig. De vergelijking volgens Kane $f_r + f_r^* = 0$ is weinig meer dan het toepassen van de tweede wet van Newton en de impulsmomentvergelijking van Euler. Het zijn dan ook niet de bewegingsvergelijkingen die bijzonder zijn maar het is veel meer de *formulering* van de bewegingsvergelijkingen die bijzonder is.

De doelstellingen van de makers van het programma AUTOLEV was een programma te maken dat:

- geschikt voor gebruik op een desktopcomputer is
- geschikt is voor het genereren van een bewegingsvergelijking voor een systeem tot ongeveer 20 lichamen
- in korte tijd de gevraagde bewegingsvergelijkingen kan produceren
- een simulatie-Fortranprogramma kan schrijven
- een grote keuze van afhankelijke variabelen buiten de gegeneraliseerde snelheden heeft
- routinematig de bewegingsvergelijkingen kan opzetten (zie appendix F)
- vergelijkingen kort kan houden door het automatisch instellen van hulpvergelijkingen
- gegeven snelheden van lichamen kan verwerken
- de gebruiker laat kiezen welke namen er aan variabelen en gegeneraliseerde coördinaten worden gegeven
- uitdrukkingen kan geven voor interne momenten en krachten die niet in de bewegingsvergelijkingen voorkomen
- gebruikersvriendelijk is, zodat dynamici na een paar uur instructie met behulp van de online helpmogelijkheid kunnen beginnen

Wat betreft het laatste punt hebben we nog wat opmerkingen, namelijk:

- de editor binnen autolev heeft heel weinig mogelijkheden (geen geheugen of wismogelijkheid)

- fouten die wel aan de 'AUTOLEV-taal' voldoen zijn soms moeilijk op te sporen
- er is nog weinig echte zekerheid over de goede invoer
- er zijn weinig extra mogelijkheden in de pre- en postprocessing binnen AUTOLEV
- AUTOLEV en ook de oorspronkelijke theorie werkt niet met matrixrepresentaties, een vorm van wiskundige notatie die afwijkt van die van ons
- door de systematiek in de naamgeving van grootheden die wel aan naamgeving gebonden zijn, is het verband met de theorie duidelijk zichtbaar
- het is vaak mogelijk een link te leggen met colleges op deze TU, zoals het college Dynamisch gedrag van constructies en het college Numerieke analyse, methoden en programmatuur

Hoewel de implementatie van de twee laatst besproken doelstellingen voor verbetering vatbaar is, vinden wij AUTOLEV een veelbelovend programma, vooral omdat het de theorie zo duidelijk ondersteunt. Ook de efficiëntie van het opstellen van de bewegingsvergelijkingen draagt hier toe bij. Om AUTOLEV zinvol te kunnen gebruiken is kennis van de theorie onontbeerlijk. De vele mogelijkheden die AUTOLEV bij goede kennis van de theorie heeft, staan in schril contrast met de soberheid van de pre- en postprocessing. Daar tegenover staat dat de resultaten bij goede kennis van de theorie ook beter zullen worden ingeschat.

We hebben nog de volgende vragen staan:

- Zijn er constructies, belastingen of situaties te verzinnen waarbij AUTOLEV in de fout gaat?
- Het is mogelijk om uitdrukkingen voor interne krachten en momenten te genereren, hoe gaat dit in zijn werk?
- Hoe kunnen we niet-holonome systemen modelleren, en wat is de achterliggende theorie daarbij?
- Hoe kunnen we efficiënt het systeem controleren op foute invoer?
- Is het mogelijk de foute invoer systematisch te localiseren?
- Hoe kunnen we interne verbindingskrachten laten uitvoeren? Is hier een algemene strategie voor?
- Kan de postprocessing worden uitgebreid met een eenvoudig (in bijvoorbeeld Pascal geschreven) animatieprogramma?

A Literatuurverwijzing

Gebruikte literatuur:

1. R.L. Huston
Computer methods in flexible multibody dynamics
International Journal For Numerical Methods In Engineering,
vol. 32, blz 1657-1668 (1991)
2. T.R. Kane & D.A. Levinson.
Dynamics: Theory and Applications
McGrawhill 1985
3. Schaechter & Levinson
AUTOLEV manual version 2.0, 1991
4. A. Sauren
Diktaat Multibody Dynamica
versie september 1990 nr. 4659

B Uitwerking translatiesnelheid

B.1 2^e term in het rechterlid van (f39)

$$(a1) \vec{\omega}^i * (\vec{b}^{i,j} + \vec{c}^j) = -(\vec{b}^{i,j} + \vec{c}^j) * \vec{\omega}^i$$

Noteer verkort:

$$(a2) \vec{g}^j = -(\vec{b}^{i,j} + \vec{c}^j)$$

Substitutie van (a2) in (a1) geeft:

$$(a3) \vec{\omega}^i * (\vec{b}^{i,j} + \vec{c}^j) \stackrel{(a1)}{=} \vec{g}^j * \vec{\omega}^i$$

Dit is te schrijven als:

$$(a4) \vec{\omega}^i * (\vec{b}^{i,j} + \vec{c}^j) = \mathbf{G}^j \cdot \vec{\omega}^i$$

met:

$$(a5) \mathbf{G}^j \cdot \vec{u}^i = \vec{g}^j * \vec{u} \quad \forall \vec{u}; (\mathbf{G}^j)^c = -(\mathbf{G}^j)$$

Voor de componenten van (a4) ten opzichte van de basis \vec{e}^0 geldt:

$$(a6) \vec{e}^0 \cdot \{ \vec{\omega}^i * (\vec{b}^{i,j} + \vec{c}^j) \} = {}^0\underline{G}^j {}^0\omega^i$$

met:

$$(a7) \vec{b}^{i,j} + \vec{c}^j = (\vec{e}^i)^T ({}^i\underline{b}^{i,j} + {}^i\underline{c}^j) = (\vec{e}^0)^T \underbrace{\mathbf{R}^i ({}^i\underline{b}^{i,j} + {}^i\underline{c}^j)}_{{}^0\underline{b}^{i,j} + {}^0\underline{c}^j}$$

$$(a8) {}^0\underline{G}^j = \begin{bmatrix} 0 & -{}^0g_3^j & {}^0g_2^j \\ {}^0g_3^j & 0 & -{}^0g_1^j \\ -{}^0g_2^j & {}^0g_1^j & 0 \end{bmatrix}$$

$$(a9) -{}^0g_3^j = {}^0\underline{b}^{i,j} + {}^0\underline{c}^j$$

$$(a10) {}^0g_1^j = -\mathbf{R}^i ({}^i\underline{b}^{i,j} + {}^i\underline{c}^j)$$

Met ${}^0\omega^i \stackrel{(f32)}{=} \underline{Z}^i \underline{y}$ resulteert in:

$$(a11) \vec{e}^0 \cdot \{ \vec{\omega}^i * (\vec{b}^{i,j} + \vec{c}^j) \} = {}^0\underline{G}^j \underline{Z}^i \underline{y}$$

met ${}^0\underline{G}^j$ volgens (a8) & (a10) en \underline{Z}^i volgens (f31).

B.2 3^e term in het rechterlid van (f39)

$$(a12) (\bar{\epsilon}^i)^T \dot{\zeta}^j \stackrel{(f8)}{=} \bar{\epsilon}^0)^T \underline{R}^i \dot{\zeta}^j$$

dus:

$$(a13) \bar{\epsilon}^0 \cdot \{ (\bar{\epsilon}^i)^T \dot{\zeta}^j \} = \underline{R}^i \dot{\zeta}^j$$

Uit (f23a) is eenvoudig af te leiden dat geldt:

$$(a14) \dot{\zeta}^j = [\underbrace{0}_{3(n+j-1)} \quad \underbrace{I}_3 \quad \underbrace{0}_{3(n-j)}] \underline{y} = \underline{L}^j \underline{y}$$

Substitutie van (a14) in (a13) geeft:

$$(a15) \bar{\epsilon}^0 \cdot \{ (\bar{\epsilon}^i)^T \dot{\zeta}^j \} = \underline{L}^j \underline{y}$$

$$\text{met } \underline{L}^j = [\underbrace{0}_{3(n+j-1)} \quad \underbrace{R^i}_3 \quad \underbrace{0}_{3(n-j)}]$$

$$\text{Opmerking: } \underline{L}^0 = [\underline{R}^0 \quad \underline{0}] = [\underbrace{I}_3 \quad \underbrace{0}_{6n-3}]$$

B.3 4^e term in het rechterlid van (f39)

$$(a16) \bar{\omega}^j * \bar{b}^{j,j} = -\bar{b}^{j,j} * \bar{\omega}^j = -\underline{K}^j \cdot \bar{\omega}^j$$

met:

$$(a17) \underline{K}^j \cdot \bar{u} = \bar{b}^{j,j} * \bar{u} \quad \forall \bar{u}; \quad (\underline{K}^j)^c = -\underline{K}^j$$

$$(a18) {}^0\underline{K}^j = \begin{bmatrix} 0 & -{}^0b_3^{j,j} & {}^0b_2^{j,j} \\ {}^0b_3^{j,j} & 0 & -{}^0b_1^{j,j} \\ -{}^0b_2^{j,j} & {}^0b_1^{j,j} & 0 \end{bmatrix}$$

Voor de vector $\bar{b}^{j,j}$ kan worden geschreven:

$$(a19) \bar{b}^{j,j} \stackrel{(f8)}{=} (\bar{\epsilon}^j)^T \dot{\zeta}^j \underline{b}^{j,j} = (\bar{\epsilon}^0)^T \underline{R}^j \dot{\zeta}^j \underline{b}^{j,j} \Rightarrow {}^0b^{j,j} = \underline{R}^j \dot{\zeta}^j \underline{b}^{j,j}$$

Substitutie van (a18), welke voor- en navermenigvuldigd wordt met de basis \vec{e}^0 , en (a19) in (a16) levert:

$$(a20) \vec{e}^0 \cdot (\vec{\omega}^j * \vec{b}^{j,j}) = -{}^0\underline{K}^j {}^0\omega^j$$

Met (f32) ${}^0\omega^j = \underline{Z}^j \underline{y}$ levert dit:

$$(a21) \vec{e}^0 \cdot (\vec{\omega}^j * \vec{b}^{j,j}) = -{}^0\underline{K}^j \underline{Z}^j \underline{y}$$

met ${}^0\underline{K}^j$ volgens (a18) & (a19) en \underline{Z}^j volgens (f31).

Substitutie van (a11), (a15) en (a21) in (f43) geeft voor de matrixrepresentatie van (f43) ten opzichte van de basis \vec{e}^0 :

$$(a22) {}^0v^j = {}^0v^i + {}^0\underline{G}^j \underline{Z}^i \underline{y} + \underline{L}^j \underline{y} + {}^0\underline{K}^j \underline{Z}^j \underline{y}$$

of:

$$(a23) {}^0v^j = {}^0v^i + ({}^0\underline{G}^j \underline{Z}^i + \underline{L}^j + {}^0\underline{K}^j \underline{Z}^j) \underline{y}$$

Vervang in (a23) de indices i en j weer door k-1 respectievelijk k:

$$(a24) {}^0v^k = {}^0v^{k-1} + ({}^0\underline{G}^k \underline{Z}^{k-1} + \underline{L}^k + {}^0\underline{K}^k \underline{Z}^k) \underline{y} \Rightarrow$$

$$(a25) k=1 : {}^0v^1 = \underbrace{{}^0v^0}_{=0} + ({}^0\underline{G}^1 \underline{Z}^0 + \underline{L}^1 + {}^0\underline{K}^1 \underline{Z}^1) \underline{y}$$

$$(a26) k=2 : {}^0v^2 = {}^0v^1 + ({}^0\underline{G}^2 \underline{Z}^1 + \underline{L}^2 + {}^0\underline{K}^2 \underline{Z}^2) \underline{y}$$

$$= \sum_{i=1}^2 ({}^0\underline{G}^i \underline{Z}^{i-1} + \underline{L}^i + {}^0\underline{K}^i \underline{Z}^i) \underline{y}$$

$$dus : (a27) \quad {}^0v^j = \sum_{i=1}^k ({}^0\underline{G}^i \underline{Z}^{i-1} + \underline{L}^i + {}^0\underline{K}^i \underline{Z}^i) \underline{y}$$

zodat:

$$(a28) {}^0v^k = \underline{Y}^k \underline{y}; \quad \underline{Y}^k = \sum_{i=1}^k ({}^0\underline{G}^i \underline{Z}^{i-1} + \underline{L}^i + {}^0\underline{K}^i \underline{Z}^i)$$

met:

$$(a28a) {}^0\underline{G}^i \stackrel{(a7)}{=} \begin{bmatrix} 0 & -{}^0g_3^i & {}^0g_2^i \\ {}^0g_3^i & 0 & -{}^0g_1^i \\ -{}^0g_2^i & {}^0g_1^i & 0 \end{bmatrix}$$

$$(a28b) \quad {}^0\underline{g}^i \stackrel{(a10)}{=} -\underline{R}^{i-1} ({}^{i-1}\underline{b}^{i-1,i} + {}^{i-1}\underline{c}^i)$$

$$(a28c) \quad \underline{Z}^i \stackrel{(f31)}{=} [\underbrace{\underline{R}^0 \quad \underline{R}^1 \quad \dots \quad \underline{R}^{i-1}}_{3i} \quad \vdots \quad \underbrace{\underline{0}}_{3(2n-i)}]$$

$$(a28d) \quad \underline{L}^i \stackrel{(a15)}{=} [\underbrace{\underline{0}}_{3(n+i-1)} \quad \vdots \quad \underbrace{\underline{R}^{i-1}}_3 \quad \vdots \quad \underbrace{\underline{0}}_{3(n-i)}]$$

$$(a28e) \quad {}^0\underline{K}^i \stackrel{(a18)}{=} \begin{bmatrix} 0 & -{}^0b_3^{i,i} & {}^0b_2^{i,i} \\ {}^0b_3^{i,i} & 0 & -{}^0b_1^{i,i} \\ -{}^0b_2^{i,i} & {}^0b_1^{i,i} & 0 \end{bmatrix}$$

$$(a28f) \quad {}^0\underline{b}^{i,i} \stackrel{(a19)}{=} \underline{R}^i \underline{b}^{i,i}$$

C Uitraai slinger.for

```

C THE NAME OF THIS PROGRAM IS SLINGER.FOR
C CREATED BY AUTOLEV ON 08-11-1992 AT 21:38:08
C
  IMPLICIT DOUBLE PRECISION (A-Z)
  INTEGER JLOOP,NSTEPS,NCUTS,NEQS,ILOOP,COUNTER,NPSTEP
  LOGICAL STPSZ
  EXTERNAL EQNS
  CHARACTER MSG(75)
  DIMENSION U(4)
  COMMON/CZEES/Z(29)
  COMMON/CPAR/IA(3),IB(3),MASSA,MASSB,PI,DEGTORAD,RADTODEG,LA,LB,G,W
& DA,WDAB
  COMMON/CONT/TA,TAB
  COMMON/CNRG/KEO,KE1,KE2,KE,PE,NRG
  COMMON/DFQLST/T,STEP,RELERR,ABSERR,NCUTS,NEQS,STPSZ
C
  OPEN(UNIT=11,FILE='SLINGER.IN',STATUS='UNKNOWN')
  OPEN(UNIT=12,FILE='SLINGER.OU1',STATUS='UNKNOWN')
  OPEN(UNIT=20,FILE='SLINGER.NRG',STATUS='UNKNOWN')
  OPEN(UNIT=21,FILE='SLINGER.H',STATUS='UNKNOWN')
  OPEN(UNIT=31,FILE='SLINGER.CD1',STATUS='UNKNOWN')
  PI = 4.000*DATAN(1.000)
  DEGTORAD = PI/180.000
  RADTODEG = 1.000/DEGTORAD
  WRITE(*,6001)
C
*****
*
* NOTE REGARDING INPUT AND OUTPUT DATA FILES
*
* The user must supply an input data file to this program. The
* file must be named FILENAME.IN, where FILENAME is obtained from
* the first line of this program. The data must be arranged in
* accordance with the READ statements that immediately follow this
* NOTE.
*
* The output from the program is sent to data files whose names
* appear on the screen at the completion of each run. The first
* column in each output data file contains the time T,
* running from zero to TMAX in increments of PSTEP. TMAX, PSTEP,
* and STEP are input from the terminal by the user at runtime, STEP
* being the initial integration stepsize, a number that should be
* chosen to be less than or equal to PSTEP. The terminal also
* prompts the user for a message identifying the run. This message
*

```



```

* is printed on each of the output files. Output files ending *
* in .OUn contain time-histories of generalized speeds and *
* generalized coordinates; files ending in .NRG contain kinetic *
* energy, potential energy, and total energy time-histories; files *
* ending in .H contain angular momentum time-histories; files ending *
* in .COn contain time-histories of quantities appearing as *
* arguments in CONTROLS commands; files ending in .SPn contain *
* time-histories of SPECIFIED variables; and files ending in .AUn *
* contain time-histories of force and/or torque measure numbers *
* corresponding to AUXILIARY generalized speeds. *
* *
*****

```

C

```

READ(11,*) LA, LB, G, WDA, WDAB
READ(11,*) MASSA, MASSB
READ(11,*) IA(1), IA(2), IA(3)
READ(11,*) IB(1), IB(2), IB(3)
READ(11,*) U(1), U(2)
READ(11,*) FI1, FI2

```

C

C

```

WRITE(*, 6002)
READ(*, 6003) (MSG(ILOOP), ILOOP = 1, 75)
WRITE(*, 6009)
READ(*, *) TMAX, PSTEP, STEPO
NPSTEP = IDINT((PSTEP-1.D-8)/STEPO + 1)
STEP = PSTEP/NPSTEP

```

C

```

WRITE(*, 6012)
WRITE(*, 6010) (MSG(ILOOP), ILOOP = 1, 75)
WRITE(12, 6101)
WRITE(12, 6010) (MSG(ILOOP), ILOOP = 1, 75)
WRITE(20, 6150)
WRITE(20, 6010) (MSG(ILOOP), ILOOP = 1, 75)
WRITE(21, 6151)
WRITE(21, 6010) (MSG(ILOOP), ILOOP = 1, 75)
WRITE(31, 6201)
WRITE(31, 6010) (MSG(ILOOP), ILOOP = 1, 75)
WRITE(*, 6011)
WRITE(12, 6011)

```

C

```

WRITE(*, 6500) LA, LB, G, WDA, WDAB
WRITE(12, 6500) LA, LB, G, WDA, WDAB
WRITE(*, 6512) IA(1), IA(2), IA(3)
WRITE(12, 6512) IA(1), IA(2), IA(3)
WRITE(*, 6513) IB(1), IB(2), IB(3)
WRITE(12, 6513) IB(1), IB(2), IB(3)

```

```
WRITE(* ,6600) MASSA,MASSE
WRITE(12,6600) MASSA,MASSE
WRITE(* ,6601) U(1),U(2)
WRITE(12,6601) U(1),U(2)
WRITE(* ,6602) FI1,FI2
WRITE(12,6602) FI1,FI2
WRITE(* ,6006) TMAX,PSTEP,STEP,STEP0
WRITE(12,6006) TMAX,PSTEP,STEP,STEP0
C
U(3) = FI1
U(4) = FI2
C
WRITE(* ,6008)
WRITE(20,6008)
WRITE(21,6007)
WRITE(12,6701)
WRITE(31,6751)
C
NEQS   = 4
NCUTS  = 20
T       = 0.0
RELERR = 1.0D-8
ABSERR = 1.0D-8
STPSZ  = .FALSE.
NSTEPS = IDINT(TMAX/STEP+0.1)+1
C
COUNTER = 0
C
C
C
DO 1000 JLOOP = 1 , NSTEPS
CALL ZEES(T,U)
C
IF (COUNTER.EQ.NPSTEP.OR.COUNTER.EQ.0) THEN
CALL ANGMOM(T,U,HN1,HN2,HN3,HN)
WRITE(21,6005) T,HN1,HN2,HN3,HN
CALL ENERGY(T,U)
WRITE(* ,6005) T,KE,PE,NRG
WRITE(20,6005) T,KE,PE,NRG
WRITE(12,6005) T,U(1),U(2),U(3),U(4)
CALL CNTRL(T,U)
WRITE(31,6005) T,TA,TAB
COUNTER = 0
ENDIF
C
COUNTER = COUNTER + 1
IF (JLOOP.EQ.NSTEPS) GO TO 1000
CALL DEQS(EQNS,U,*99)
```

```

1000 CONTINUE
C
      WRITE(*,6999)
C
      STOP
99 WRITE(*,6004)
C
6001 FORMAT(/1X,'SYSTEM PARAMETERS AND INITIAL CONDITIONS'/
&      2X,'ARE NOW BEING READ FROM THE INPUT FILE'//)
6002 FORMAT(1X,'INPUT A DESCRIPTION OF THIS RUN'//)
6003 FORMAT(75A1)
6004 FORMAT(1X,'STEPSIZE HALVED TOO MANY TIMES'//)
6005 FORMAT(6(1X,1PE12.5))
6006 FORMAT(11X,'TMAX = ',1PE12.5,' S'/10X,'PSTEP = ',1PE12.5,' S'/11X,
&'STEP = ',1PE12.5,' S (USER INPUT VALUE = ',1PE12.5,' S)'//)
6007 FORMAT(/1X,'SIMULATION RESULTS'//7X,'T',11X,'HN1',10X,'HN2',10X,'
&HN3',10X,'HN',/6X,'(S)',8X,'(N*M*S)',6X,'(N*M*S)',6X,'(N*M*S)',6X,
&'(N*M*S)',/)
6008 FORMAT(/1X,'SIMULATION RESULTS'//7X,'T',11X,'KE',11X,'PE',9X,'KE
&+ PE',/6X,'(S)',9X,'(N*M)',8X,'(N*M)',8X,'(N*M)',/)
6009 FORMAT(/1X,'INPUT TMAX, PSTEP, STEP '/
&1X,'====='/
&1X,'| TMAX : FINAL TIME |'/
&1X,'| PSTEP: TIME INTERVAL FOR PRINTING |'/
&1X,'| STEP : MAXIMUM INTEGRATION TIME STEP |'/
&1X,'=====')
6010 FORMAT(1X,'*** ',75A1)
6011 FORMAT(/1X,'SYSTEM PARAMETERS'//)
6012 FORMAT(1X,'OUTPUT FROM PROGRAM SLINGER.FOR'//)
6101 FORMAT(1X,'FILE: SLINGER.OU1 (OUTPUT FROM PROGRAM SLINGER.FOR)'/
&/)
6150 FORMAT(1X,'FILE: SLINGER.NRG (OUTPUT FROM PROGRAM SLINGER.FOR)'/
&/)
6151 FORMAT(1X,'FILE: SLINGER.H (OUTPUT FROM PROGRAM SLINGER.FOR)'/)
6201 FORMAT(1X,'FILE: SLINGER.CO1 (OUTPUT FROM PROGRAM SLINGER.FOR)'/
&/)
6500 FORMAT(13X,'LA = ',1PE12.5,' M'/13X,'LB = ',1PE12.5,' M'/14X,'G =
&',1PE12.5,' M/S^2'/12X,'WDA = ',1PE12.5,' N*M/S'/11X,'WDAB = ',1PE
&12.5,' NM'//)
6512 FORMAT(10X,'IA(1) = ',1PE12.5,' KG*M^2'/10X,'IA(2) = ',1PE12.5,' K
&G*M^2'/10X,'IA(3) = ',1PE12.5,' KG*M^2'//)
6513 FORMAT(10X,'IB(1) = ',1PE12.5,' KG*M^2'/10X,'IB(2) = ',1PE12.5,' K
&G*M^2'/10X,'IB(3) = ',1PE12.5,' KG*M^2'//)
6600 FORMAT(10X,'MASSA = ',1PE12.5,' KG'/10X,'MASSB = ',1PE12.5,' KG'//
&)
6601 FORMAT(/1X,'INITIAL CONDITIONS'//10X,'U1(0) = ',1PE12.5,' RAD/S'/1
&0X,'U2(0) = ',1PE12.5,' RAD/S'//)

```

```

6602 FORMAT(9X,'FI1(0) = ',1PE12.5,' RAD'/9X,'FI2(0) = ',1PE12.5,' RAD'
&/)
6701 FORMAT(/1X,'SIMULATION RESULTS'//7X,'T',11X,'U1',11X,'U2',11X,'FI
&1',10X,'FI2',/6X,'(S)',8X,'(RAD/S)',6X,'(RAD/S)',7X,'(RAD)',8X,'(R
&AD)',/)
6751 FORMAT(/1X,'SIMULATION RESULTS'//7X,'T',11X,'TA',11X,'TAB',/6X,'(
&S)',9X,'(N*M)',8X,'(N*M)',/)
6999 FORMAT(/1X,'OUTPUT IS ON FILES: ', 'SLINGER.OU1'/22X,'SLINGER.H'/
&22X,'SLINGER.NRG'/22X,'SLINGER.CO1'/22X,/)
END

```

C
C
C

```

SUBROUTINE EQNS(T,U,UDOT)
IMPLICIT DOUBLE PRECISION (A-Z)
DIMENSION U(4),UDOT(4),COEF(2,2),RHS(2)
COMMON/CZEES/Z(29)
COMMON/CPAR/IA(3),IB(3),MASSA,MASSB,PI,DEGTORAD,RADTODEG,LA,LB,G,W
& DA,WDAB
COMMON/CONT/TA,TAB

```

C
C
C

```
CALL ZEES(T,U)
```

```
FI1 = U(3)
FI2 = U(4)
```

C
C
C

```
CALL CNTRL(T,U)
```

```
COEF(1,1) = Z(22)
COEF(1,2) = Z(23)
COEF(2,1) = Z(23)
COEF(2,2) = Z(25)
```

C
C
C

```
RHS(1) = G*MASSA*Z(1)+G*MASSB*Z(5)-TA+Z(24)
RHS(2) = G*MASSB*Z(6)-TAB+Z(26)
```

C
C
C

```
CALL UNCUPL(2,COEF,RHS,UDOT)
```

C
C
C

```
U3 IS DEFINED TO BE FI1
UDOT(3) = U(1)
```

C
C
C

```
U4 IS DEFINED TO BE FI2
UDOT(4) = U(2)
```

```

C
  RETURN
  END
C
C
C
  SUBROUTINE UNCUPL(NDIM,COEF,RHS,UDOT)
  IMPLICIT DOUBLE PRECISION (A-Z)
  INTEGER NDIM,IPS(50)
  DIMENSION COEF(NDIM,NDIM),RHS(NDIM),UDOT(NDIM)
C
  CALL DECMP2(NDIM,COEF,NDIM,COEF,IPS,*901,*902)
  CALL SOLVE2(NDIM,COEF,NDIM,RHS,UDOT,IPS)
C
  RETURN
C
  901 WRITE(*,601)
  STOP
  902 WRITE(*,602)
  STOP
  601 FORMAT(/1X,'ALL ELEMENTS IN A ROW OF COEF ARE ZEROS'/)
  602 FORMAT(/1X,'A PIVOT ELEMENT ENCOUNTERED IN THE DECOMPOSITION',
C      ' OF COEF IS ZERO')
  END
C
C
C
  SUBROUTINE ZEES(T,U)
  IMPLICIT DOUBLE PRECISION (A-Z)
  DIMENSION U(4)
  COMMON/CZEES/Z(29)
  COMMON/CPAR/IA(3),IB(3),MASSA,MASSB,PI,DEGTORAD,RADTODEG,LA,LB,G,W
  & DA,WDAB
C
  FI1 = U(3)
  FI2 = U(4)
C
C
  S1 = DSIN(FI1)
  C1 = DCOS(FI1)
  S2 = DSIN(FI1+FI2)
  C2 = DCOS(FI1+FI2)
  S3 = DSIN(FI2)
  C3 = DCOS(FI2)
C
  Z(1) = LA*S1
  Z(2) = C1*LA

```

```

Z(3) = LA*S1+Z(1)
Z(4) = C1*LA+Z(2)
Z(5) = LB*S2+Z(3)
Z(6) = LB*S2
Z(7) = C2*LB+Z(4)
Z(8) = C2*LB
Z(9) = C1*LA*U(1)
Z(10) = LA*S1*U(1)
Z(11) = U(1)*Z(9)
Z(12) = U(1)*Z(10)
Z(13) = C1*LA*U(1)+Z(9)
Z(14) = LA*S1*U(1)+Z(10)
Z(15) = (U(1)+U(2))*C2*LB+Z(13)
Z(16) = (U(1)+U(2))*C2*LB
Z(17) = (U(1)+U(2))*LB*S2+Z(14)
Z(18) = (U(1)+U(2))*LB*S2
Z(19) = U(1)*Z(15)+U(2)*Z(16)
Z(20) = U(1)*Z(17)+U(2)*Z(18)
Z(21) = U(1)+U(2)
Z(22) = (-Z(1)*Z(1)-Z(2)*Z(2))*MASSA+(-Z(5)*Z(5)-Z(7)*Z(7))*MASSB-
& IA(3)-IB(3)
Z(23) = (-Z(5)*Z(6)-Z(7)*Z(8))*MASSB-IB(3)
Z(24) = (Z(1)*Z(11)-Z(12)*Z(2))*MASSA+(Z(19)*Z(5)-Z(20)*Z(7))*MASS
& B
Z(25) = (-Z(6)*Z(6)-Z(8)*Z(8))*MASSB-IB(3)
Z(26) = (Z(19)*Z(6)-Z(20)*Z(8))*MASSB
Z(27) = MASSA+MASSB
Z(28) = (C3*LA+LB)*MASSB/Z(27)
Z(29) = LA*MASSB*S3/Z(27)

```

C

```

RETURN
END

```

C

C

C

```

SUBROUTINE CNTRL(T,U)
IMPLICIT DOUBLE PRECISION (A-Z)
DIMENSION U(4)
COMMON/CZEES/Z(29)
COMMON/CPAR/IA(3),IB(3),MASSA,MASSB,PI,DEGTORAD,RADTODEG,LA,LB,G,W
& DA,WDAB
COMMON/CONT/TA,TAB
COMMON/CNRG/KE0,KE1,KE2,KE,PE,NRG

```

C

```

FI1 = U(3)
FI2 = U(4)

```

C

```

TA = -WDA*U(1)
TAB = -WDAB*U(2)
C
RETURN
END
C
C
C
SUBROUTINE ANGMOM(T,U,HN1,HN2,HN3,HN)
IMPLICIT DOUBLE PRECISION (A-Z)
DIMENSION U(4)
COMMON/CZEES/Z(29)
COMMON/CPAR/IA(3),IB(3),MASSA,MASSB,PI,DEGTORAD,RADTODEG,LA,LB,G,W
& DA,WDAB
C
FI1 = U(3)
FI2 = U(4)
C
C
S1 = DSIN(FI1)
C1 = DCOS(FI1)
S2 = DSIN(FI2)
C2 = DCOS(FI2)
C
ZH1 = -C2*Z(28)-S2*Z(29)
ZH2 = C2*Z(29)-S2*Z(28)
ZH3 = -C1*U(1)*Z(1)+S1*U(1)*Z(2)
ZH4 = C1*U(1)*Z(2)+S1*U(1)*Z(1)
ZH5 = ZH1*ZH4-ZH2*ZH3
ZH6 = (LB-Z(28))*C2+LA-S2*Z(29)
ZH7 = (LB-Z(28))*S2+C2*Z(29)
ZH8 = (-U(1)*Z(5)-U(2)*Z(6))*C1+(U(1)*Z(7)+U(2)*Z(8))*S1
ZH9 = -(-U(1)*Z(5)-U(2)*Z(6))*S1+(U(1)*Z(7)+U(2)*Z(8))*C1
ZH10 = ZH6*ZH9-ZH7*ZH8
HN1 = 0.0
HN2 = 0.0
HN3 = IA(3)*U(1)+IB(3)*Z(21)+MASSA*ZH5+MASSB*ZH10
HN = DSQRT(HN1*HN1 + HN2*HN2 + HN3*HN3)
C
RETURN
END
C
C
C
SUBROUTINE ENERGY(T,U)
IMPLICIT DOUBLE PRECISION (A-Z)
DIMENSION U(4)

```

```

COMMON/CZEES/Z(29)
COMMON/CPAR/IA(3),IB(3),MASSA,MASSB,PI,DEGTORAD,RADTODEG,LA,LB,G,W
& DA,WDAB
COMMON/CNRG/KEO,KE1,KE2,KE,PE,NRG

```

C

```

FI1 = U(3)
FI2 = U(4)

```

C

C

```

S1 = DSIN(FI1)
C1 = DCOS(FI1)
S2 = DSIN(FI1+FI2)
C2 = DCOS(FI1+FI2)

```

C

```

ZKE1 = -U(1)*Z(1)
ZKE2 = U(1)*Z(2)
ZKE3 = (ZKE1*ZKE1+ZKE2*ZKE2)*MASSA+IA(3)*U(1)*U(1)
ZKE4 = -U(1)*Z(5)-U(2)*Z(6)
ZKE5 = U(1)*Z(7)+U(2)*Z(8)
ZKE6 = (IB(3)*U(1)+IB(3)*U(2))*(U(1)+U(2))+(ZKE4*ZKE4+ZKE5*ZKE5)*M

```

& ASSB

```

KE0 = 0
KE1 = 0
KE2 = .5*(ZKE3+ZKE6)
KE = KE0+KE1+KE2
PE = (MASSA+2*MASSB)*(1-C1)*G*LA+MASSB*G*(1-C2)*LB
NRG = KE + PE

```

C

```

RETURN
END

```

C

C

C

```

SUBROUTINE DECMP2(N,A, IDIM,LU, IPS,*,*)
IMPLICIT DOUBLE PRECISION (A-Z)
INTEGER N, IDIM, IPS(N), I, J, K, IP, KP, KP1, NM1, IDXPIV
DIMENSION A(IDIM,N), LU(IDIM,N), SCALES(100)
ZERO=0.0D0
DO 5 I=1,N
  IPS(I)=I
  ROWNRM=0.0D0
DO 2 J=1,N
  LU(I,J)=A(I,J)
  ROWNRM=DMAX1(ROWNRM,DABS(LU(I,J)))
2 CONTINUE
IF(ROWNRM.EQ.ZERO) RETURN 1
SCALES(I)=1.0/ROWNRM

```



```

5  CONTINUE
   NM1=N-1
   DO 17 K=1,NM1
   BIG=0.0D0
   DO 11 I=K,N
   IP=IPS(I)
   SIZE=DABS(LU(IP,K))*SCALES(IP)
   IF(SIZE.LE.BIG) GO TO 11
   BIG=SIZE
   IDXPIV=I
11  CONTINUE
   IF(BIG.EQ.ZERO) RETURN 2
   IF(IDXPIV.EQ.K) GO TO 15
   J=IPS(K)
   IPS(K)=IPS(IDXPIV)
   IPS(IDXPIV)=J
15  KP=IPS(K)
   PIVOT=LU(KP,K)
   KP1=K+1
   DO 16 I=KP1,N
   IP=IPS(I)
   EM=LU(IP,K)/PIVOT
   LU(IP,K)=EM
   DO 16 J=KP1,N
   LU(IP,J)=LU(IP,J)-EM*LU(KP,J)
16  CONTINUE
17  CONTINUE
   IF(LU(IPS(N),N).EQ.ZERO) RETURN 2
   RETURN
   END

```

C
C
C

```

SUBROUTINE SOLVE2(N,LU,IDIM,B,X,IPS)
IMPLICIT DOUBLE PRECISION (A-Z)
INTEGER I,J,IP,IP1,IM1,NP1,IBACK,N,IDIM,IPS(N)
DIMENSION LU(IDIM,N),B(N),X(N)
NP1=N+1
X(1)=B(IPS(1))
DO 2 I=2,N
IP=IPS(I)
IM1=I-1
SUM=0.0D0
DO 1 J=1,IM1
SUM=SUM+LU(IP,J)*X(J)
1  CONTINUE
X(I)=B(IP)-SUM

```

```

2  CONTINUE
   X(N)=X(N)/LU(IPS(N),N)
   DO 4 IBACK=2,N
     I=NP1-IBACK
     IP=IPS(I)
     IP1=I+1
     SUM=0.0DO
     DO 3 J=IP1,N
       SUM=SUM+LU(IP,J)*X(J)
3  CONTINUE
4  X(I)=(X(I)-SUM)/LU(IP,I)
   RETURN
   END

C
C
SUBROUTINE DEQS(F,Y,*)
IMPLICIT DOUBLE PRECISION (A-Z)
INTEGER I,NCUTS,NEQ
LOGICAL DBL,STPSZ
EXTERNAL F
COMMON/DFQLST/T,STEP,REL,ABS,NCUTS,NEQ,STPSZ
DIMENSION F0(200),F1(200),F2(200),Y1(200),Y2(200),Y(NEQ)
DATA HC/0.0DO/
C *** CHECK FOR INITIAL ENTRY AND ADJUST HC, IF NECESSARY.
   IF(NEQ.NE.0) GO TO 10
   HC=STEP
   RETURN
10  IF(STEP.EQ.0.0DO) RETURN 1
C *** CHANGE DIRECTION, IF REQUIRED.
   IF(HC*STEP) 20,30,40
20  HC=-HC
   GO TO 40
30  HC=STEP
C *** SET LOCAL VARIABLES
40  EPSL=REL
   FINAL=T+STEP
   H=HC
   TT=T+H
   T=FINAL
   H2=H/2.0DO
   H3=H/3.0DO
   H6=H/6.0DO
   H8=H/8.0DO
C *** MAIN KUTTA-MERSON STEP
50  IF((H.GT.0.0DO.AND.TT.GT.FINAL).OR.
      C (H.LT.0.0DO.AND.TT.LT.FINAL)) GO TO 190
60  CALL F(TT-H,Y,F0)

```

```
DO 70 I=1,NEQ
70  Y1(I)=F0(I)*H3+Y(I)
   CALL F(TT-2.0*H3,Y1,F1)
DO 80 I=1,NEQ
80  Y1(I)=(F0(I)+F1(I))*H6+Y(I)
   CALL F(TT-2.0*H3,Y1,F1)
DO 90 I=1,NEQ
90  Y1(I)=(F1(I)*3.0+F0(I))*H8+Y(I)
   CALL F(TT-H2,Y1,F2)
DO 100 I=1,NEQ
100 Y1(I)=(F2(I)*4.0-F1(I)*3.0+F0(I))*H2+Y(I)
   CALL F(TT,Y1,F1)
DO 110 I=1,NEQ
110 Y2(I)=(F2(I)*4.0+F1(I)+F0(I))*H6+Y(I)
C *** DOES THE STEPSIZE H NEED TO BE CHANGED?
   IF(EPSL.LE.0.0D0) GO TO 170
   DBL=.TRUE.
   DO 160 I=1,NEQ
   ERR=DABS(Y1(I)-Y2(I))*0.2
   TEST=DABS(Y1(I))*EPSL
   IF(ERR.LT.TEST.OR.ERR.LT.ABS) GO TO 150
C *** HALVE THE STEPSIZE
   H=H2
   TT=TT-H2
   IF(.NOT.STPSZ) GO TO 120
   TEMP=TT-H2
   WRITE(*,200) H,TEMP
C *** HAS THE STEPSIZE BEEN HALVED TOO MANY TIMES?
120  NCUTS=NCUTS-1
   IF(NCUTS.GE.0) GO TO 130
   T=TT-H2
   WRITE(*,210) T
   RETURN 1
C *** IF STEPSIZE IS TOO SMALL RELATIVE TO TT TAKE RETURN 1
130  IF(TT+H.NE.TT) GO TO 140
   T=TT
   RETURN 1
140  H2=H/2.0D0
   H3=H/3.0D0
   H6=H/6.0D0
   H8=H/8.0D0
   GO TO 60
150  IF(DBL.AND.64.0D0*ERR.GT.TEST
C      .AND.64.0D0*ERR.GT.ABS) DBL=.FALSE.
160  CONTINUE
C *** DOUBLE THE STEPSIZE, MAYBE.
   IF(.NOT.DBL.OR.DABS(2.0D0*H).GT.DABS(STEP).OR.
```

```
C   DABS(TT+2.0D0*H).GT.DABS(FINAL).AND.
C   DABS(TT-FINAL).GT.DABS(FINAL)*1.0D-7) GO TO 170
H2=H
H=H+H
IF(STPSZ) WRITE(*,200) H,TT
H3=H/3.0D0
H6=H/6.0D0
H8=H/8.0D0
NCUTS=NCUTS+1
170  DO 180 I=1,NEQ
180  Y(I)=Y2(I)
      TT=TT+H
      GO TO 50
190  IF(EPSL.LT.0.0D0) RETURN
C *** NOW BE SURE TO HAVE T=FINAL.
      HC=H
      H=FINAL-(TT-H)
      IF(DABS(H).LE.DABS(FINAL)*1.0D-7) RETURN
      TT=FINAL
      EPSL=-1.0D0
      H2=H/2.0D0
      H3=H/3.0D0
      H6=H/6.0D0
      H8=H/8.0D0
      GO TO 60
200  FORMAT(1X,'THE STEPSIZE IS NOW ',1PD12.4,' AT T = ',1PD12.4)
210  FORMAT(1X,'THE STEPSIZE HAS BEEN HALVED TOO MANY TIMES; ',
C      'T = ',1PD12.4)
      END
```

D Resultaten Simulaties

D.1 Voorbeelden output-files

FILE: SLINGER.OU1 (OUTPUT FROM PROGRAM SLINGER.FOR)

*** Vrije beweging van de slinger zonder demping

SYSTEM PARAMETERS

LA = 1.00000E+00 M
LB = 1.00000E+00 M
G = 9.81000E+00 M/S²
WDA = 0.00000E-01 N*M/S
WDAB = 0.00000E-01 NM

IA(1) = 0.00000E-01 KG*M²
IA(2) = 2.66667E+00 KG*M²
IA(3) = 2.66667E+00 KG*M²

IB(1) = 0.00000E-01 KG*M²
IB(2) = 5.33333E+00 KG*M²
IB(3) = 5.33333E+00 KG*M²

MASSA = 2.00000E+00 KG
MASSB = 4.00000E+00 KG

INITIAL CONDITIONS

U1(0) = 0.00000E-01 RAD/S
U2(0) = 0.00000E-01 RAD/S

FI1(0) = 7.85400E-01 RAD
FI2(0) = 7.85400E-01 RAD

TMAX = 1.00000E+01 S
PSTEP = 1.00000E-01 S
STEP = 1.00000E-01 S (USER INPUT VALUE = 1.00000E-01 S)

SIMULATION RESULTS

T (S)	U1 (RAD/S)	U2 (RAD/S)	FI1 (RAD)	FI2 (RAD)
0.00000E-01	0.00000E-01	0.00000E-01	7.85400E-01	7.85400E-01
1.00000E-01	-2.61460E-01	-1.89390E-03	7.72252E-01	7.85460E-01
2.00000E-01	-5.04946E-01	-4.02686E-02	7.33707E-01	7.83802E-01
:	:	:	:	:
:	:	:	:	:
9.90000E+00	-5.75472E-01	7.49501E-01	-8.00972E-01	-6.67137E-01
1.00000E+01	-3.03698E-01	7.17326E-01	-8.45089E-01	-5.93321E-01

FILE: SLINGER.CO1 (OUTPUT FROM PROGRAM SLINGER.FOR)

*** Vrije beweging van de slinger zonder demping

SIMULATION RESULTS

T (S)	TA (N*M)	TAB (N*M)
0.00000E-01	0.00000E-01	0.00000E-01
1.00000E-01	0.00000E-01	0.00000E-01
2.00000E-01	0.00000E-01	0.00000E-01
:	:	:
:	:	:
9.90000E+00	0.00000E-01	0.00000E-01
1.00000E+01	0.00000E-01	0.00000E-01

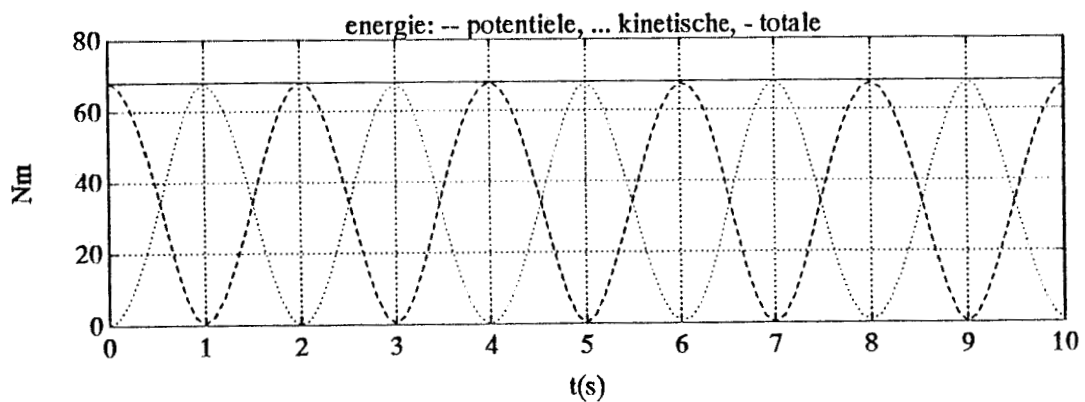
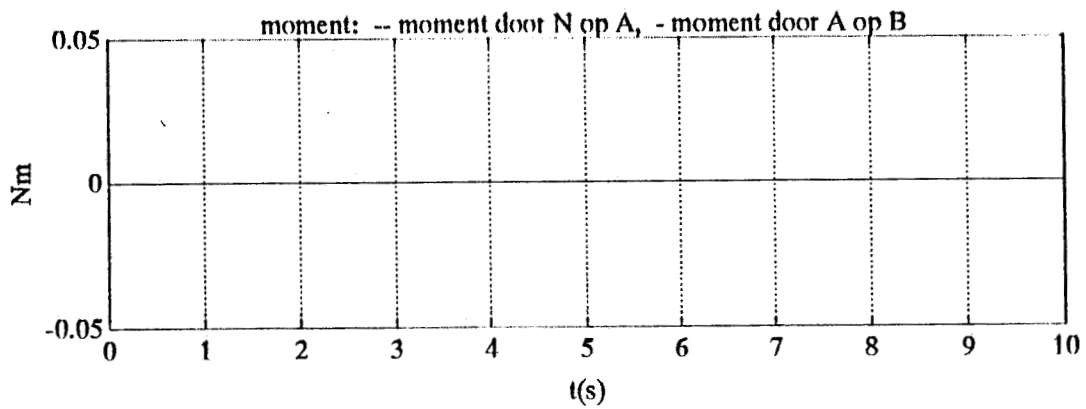
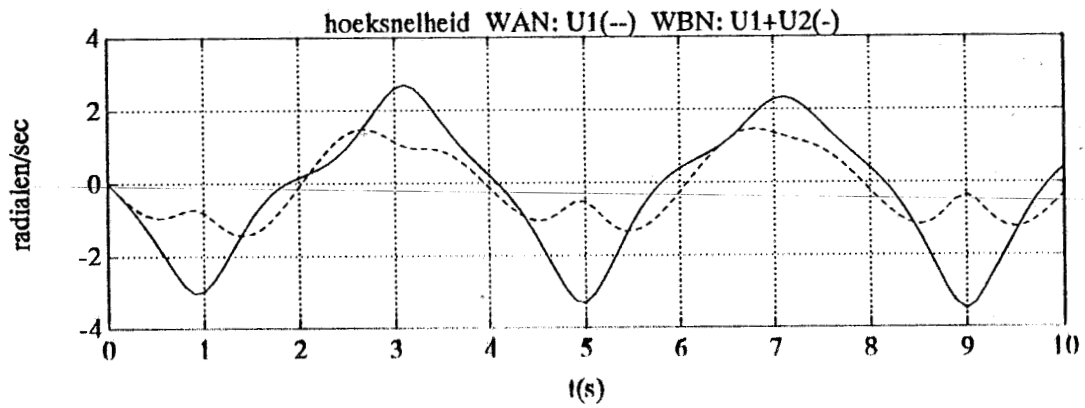
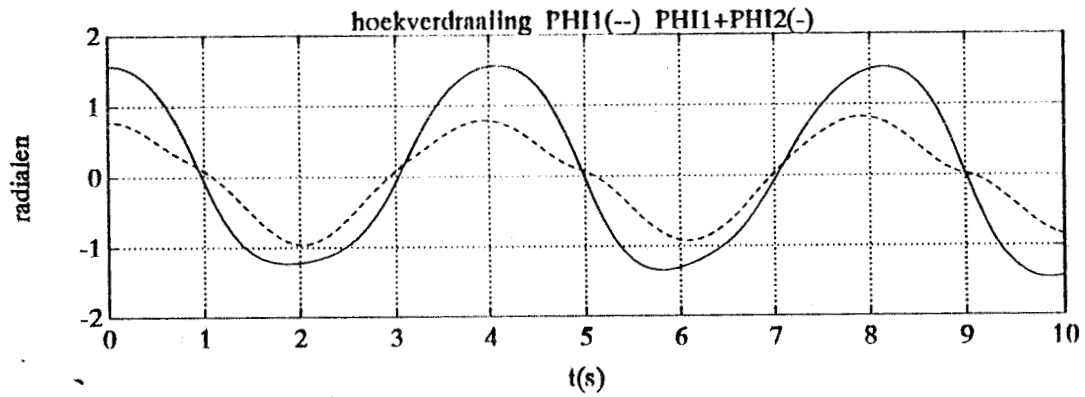
FILE: SLINGER.NRG (OUTPUT FROM PROGRAM SLINGER.FOR)

*** Vrije beweging van de slinger zonder demping

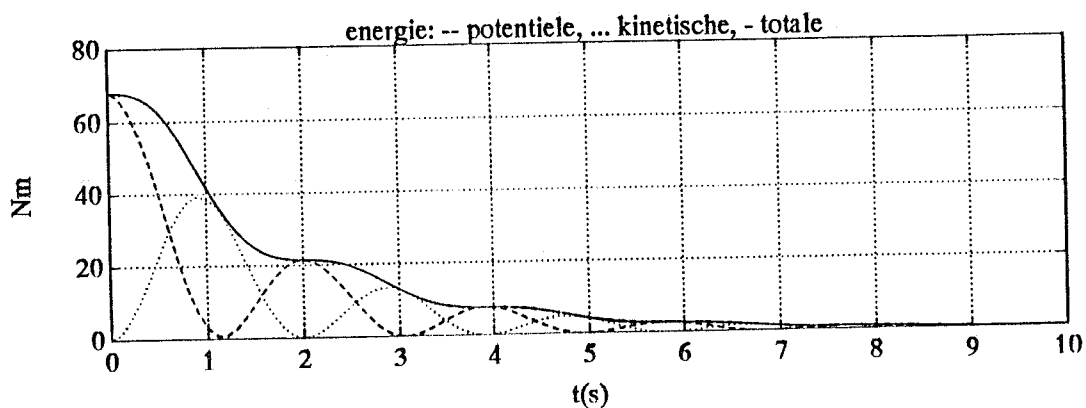
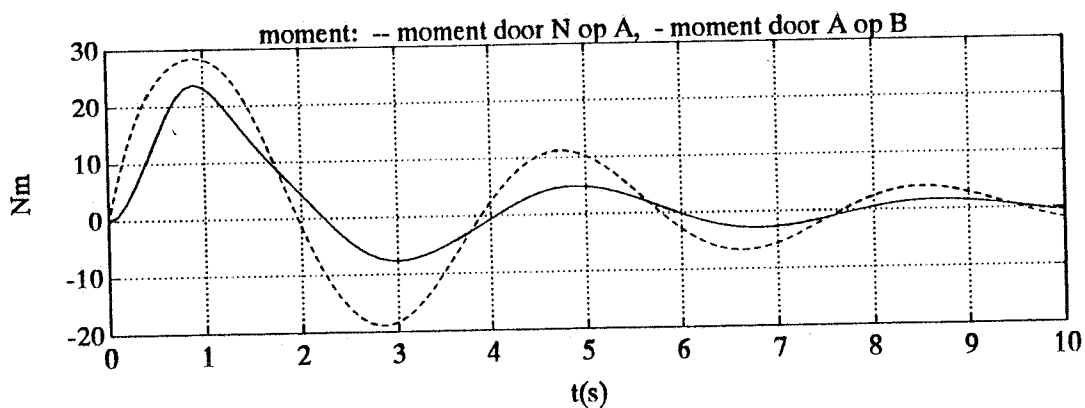
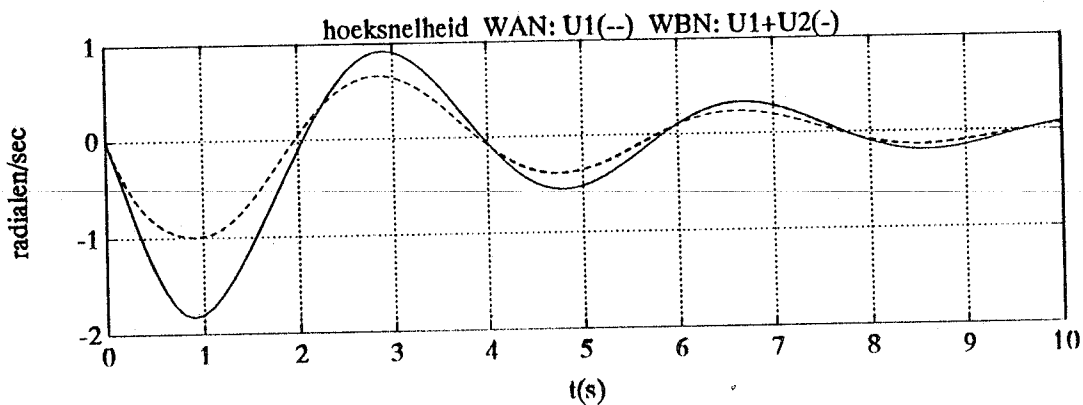
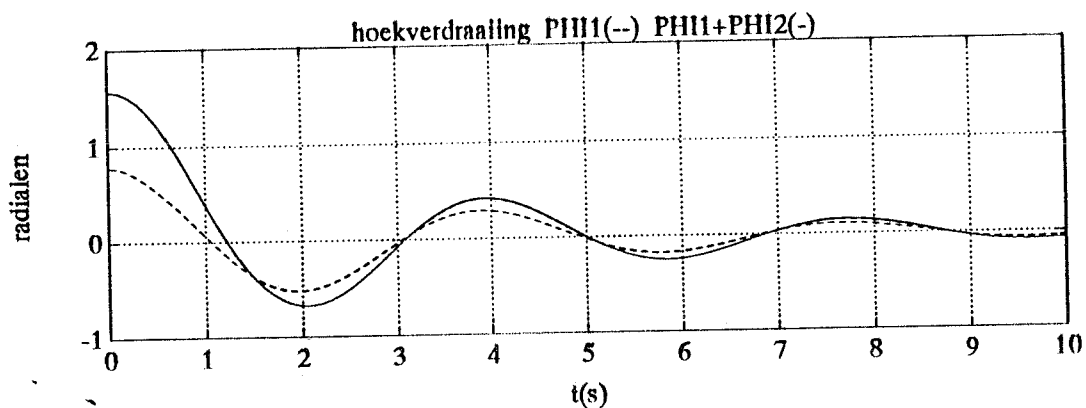
SIMULATION RESULTS

T (S)	KE (N*M)	PE (N*M)	KE + PE (N*M)
0.00000E-01	0.00000E-01	6.79731E+01	6.79731E+01
1.00000E-01	1.41954E+00	6.65536E+01	6.79731E+01
2.00000E-01	5.58173E+00	6.23914E+01	6.79731E+01
:	:	:	:
:	:	:	:
9.90000E+00	2.93399E+00	6.50391E+01	6.79731E+01
1.00000E+01	9.18292E-01	6.70548E+01	6.79731E+01

D.2 Slinger zonder demping



D.3 Slinger met demping



D.4 Resultaten DADS versus AUTOLEV

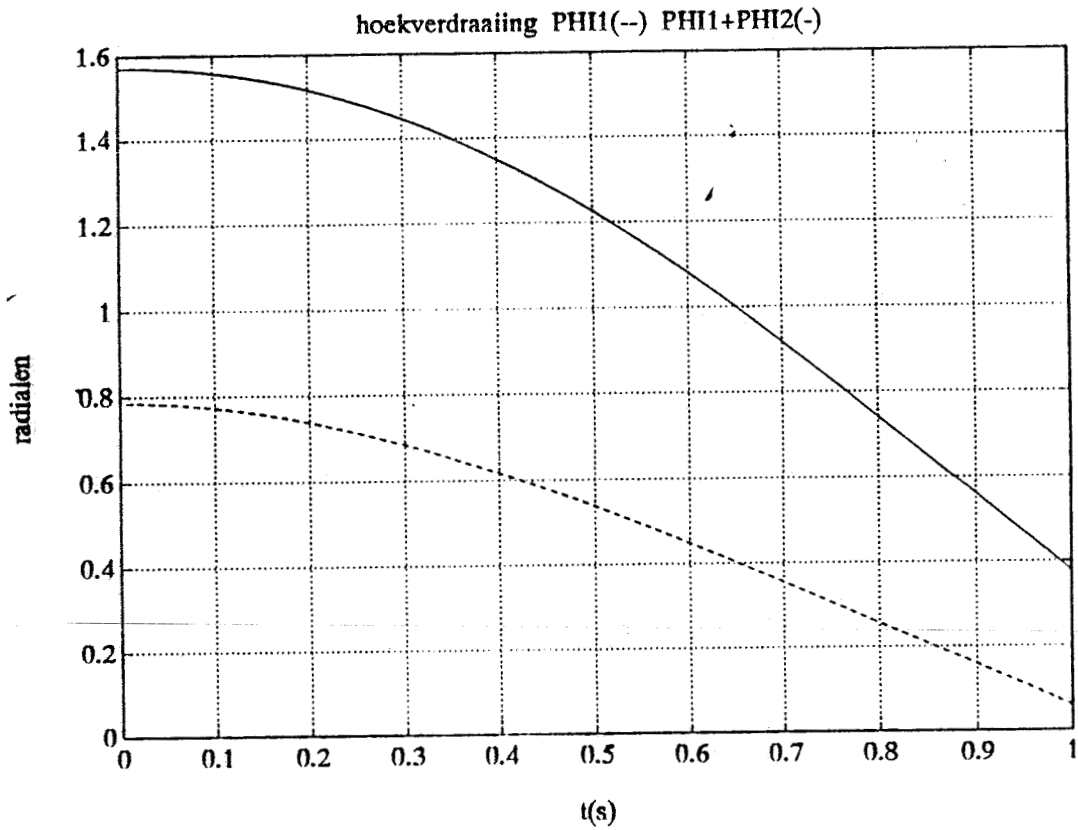
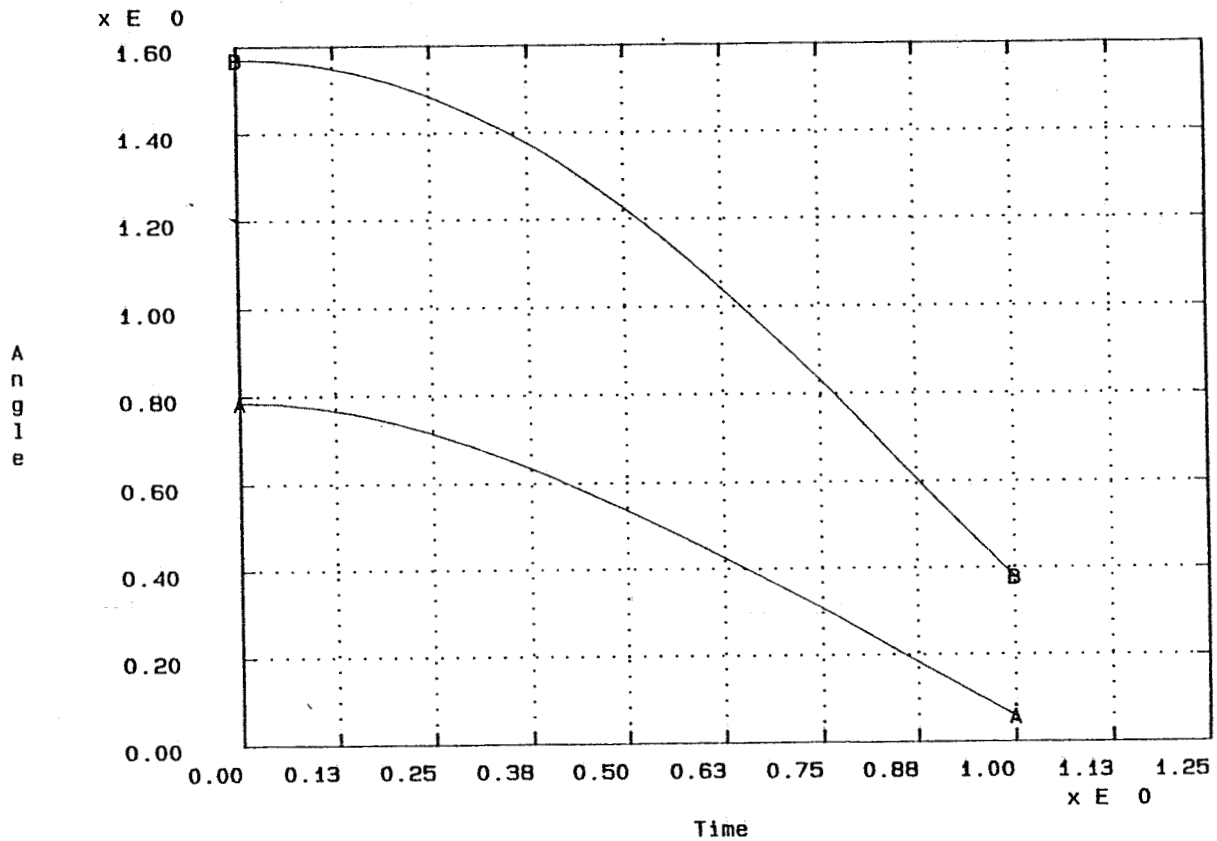


Figure 2. Angle vs. Time: A=Pend1, B=Pend2.



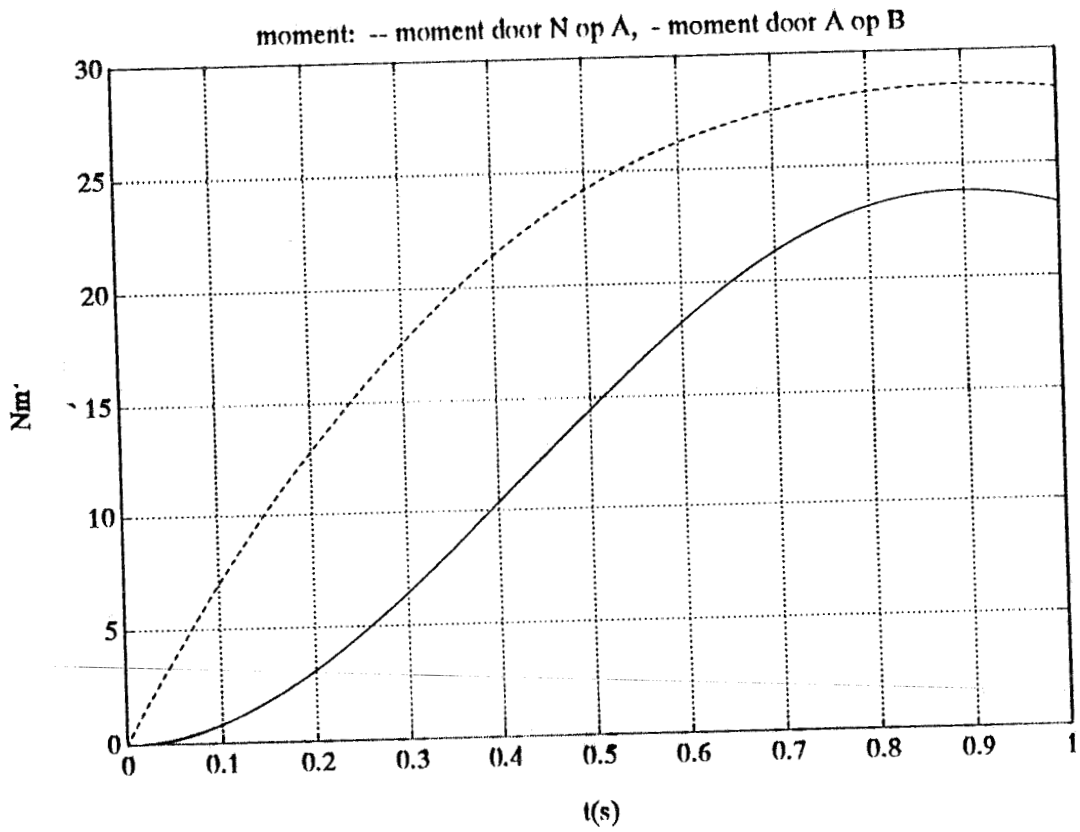
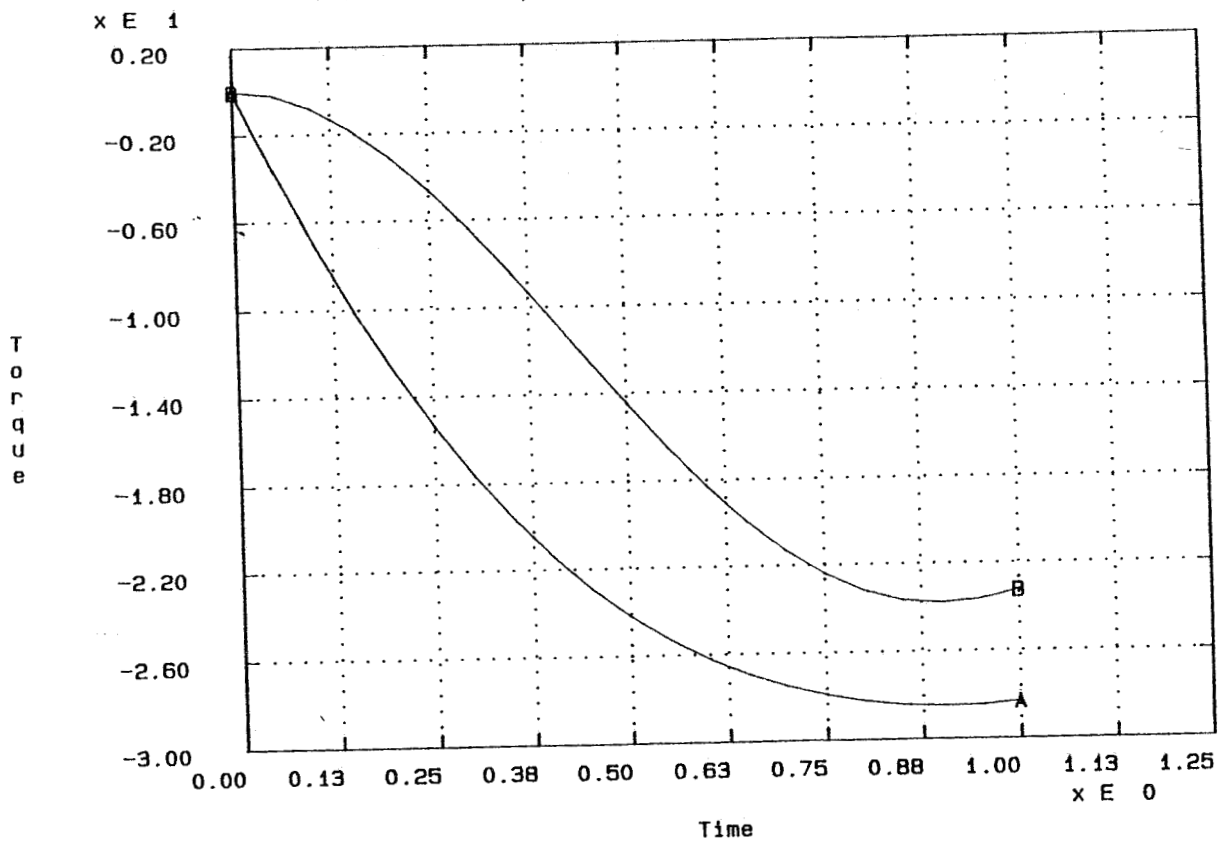


Figure 5. Torques: A=Rev1, B=Rev2



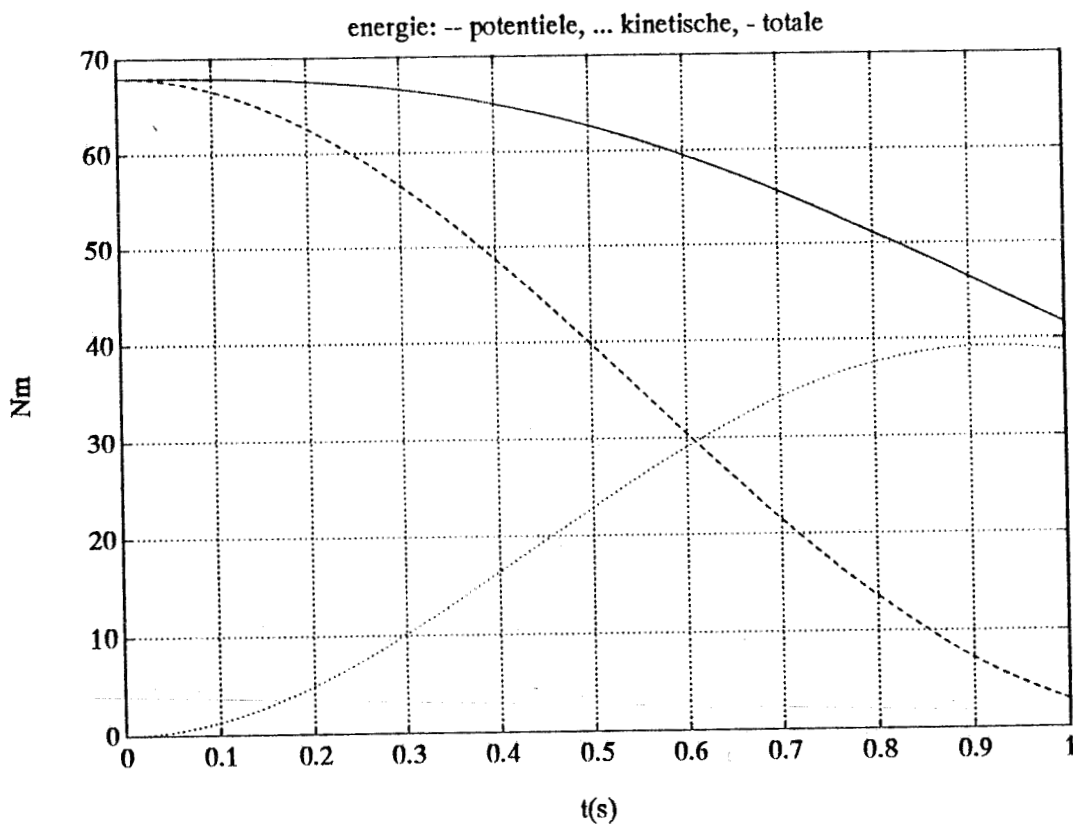
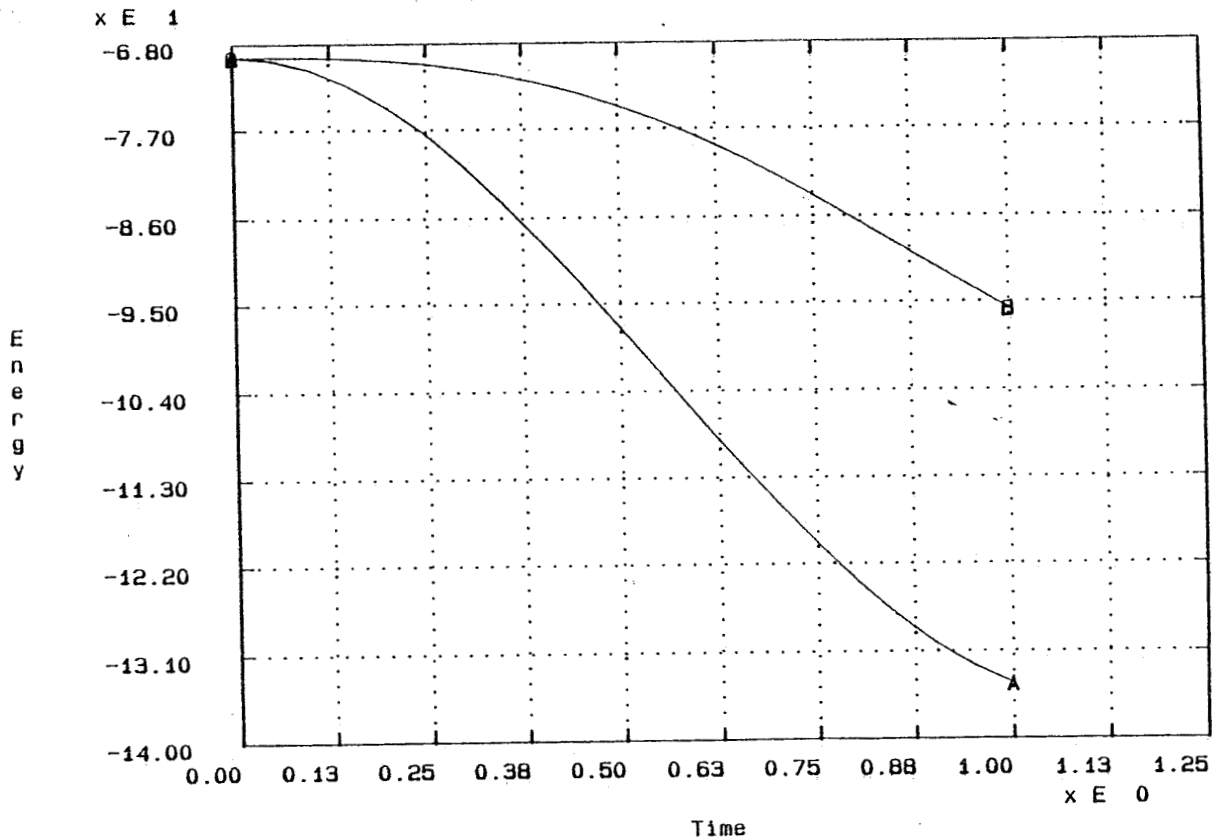


Figure 3. Energy vs. Time: A=potential, B=Total



E Gebruik van AUTOLEV onder DOS

E.1 Installatie

Om AUTOLEV te installeren op harddisk C: maakt men op de harde schijf een directory C:\ AUTOLEV. Van hieruit geeft men het commando A:INSTALL (indien A: de naam is van de diskdrive met de AUTOLEV-diskettes). Vervolgens moet men in het pad C:\ AUTOLEV; toevoegen. Vanuit een werkdirectory (bijv C:\ AUTOLEV\ WORK, zelf aan te maken) kan men dan het DOS-commando AUTOLEV geven.

E.2 Werken binnen AUTOLEV

Er volgen nu enige veel gebruikte commando's. Indien er vragen zijn over het gebruik van een commando kan men HELP COMMANDO typen. Het commando WHAT geeft aan welke commando's beschikbaar zijn.

CLEAR wist de geprogrammeerde omgeving uit, met QUIT verlaat men AUTOLEV. Met het commando LIST kan men de invoer met antwoorden bekijken (zie HELP LIST voor de opties). Er zijn twee mogelijkheden om binnen AUTOLEV ingevoerde gegevens te bewaren en op te vragen. Het commando RECORD(file) bewaart de zelf ingevoerde regels in de file 'file'. Deze kan met een ASCII-editor (bijv. de Norton Editor) buiten AUTOLEV worden bewerkt. Binnen AUTOLEV wordt met het commando EXEC(file) de file 'afgespeeld'. Het commando SAVE(file) bewaart de invoer inclusief de antwoorden van AUTOLEV in twee files namelijk 'file.wrk' en 'file.sav'. Met het commando LOAD(file) wordt de huidige werkruimte gewist en de bewaarde omgeving opgeroepen. De bewaarde files bevinden zich in de directory van waaruit men gestart is, dus in ons geval C:\ AUTOLEV\ WORK

E.3 Compileren Fortranprogramma

Naast de installatiediskettes moet er ook een schijfje met een Fortrancompiler worden gekopieerd. Zetten we deze in een directory C:\ FOR en passen we de batchfile PROFOR.BAT daarop aan, dan kunnen we met het geven van PROFOR PROGRAMMA de file PROGRAMMA.FOR compileren en linken. In de file PROGRAMMA.FOR staat na de opmerking over in- en uitvoerfiles het readstatement. In dezelfde vorm moet er een invoerfile worden gemaakt met de naam PROGRAMMA.IN waarin de waarde van de gevraagde grootheden in staat vermeld. Denk eraan een enter te geven na de laatste waarde, anders kan de file niet worden gelezen.

E.4 Grafische verwerking van resultaten met Matlab

In dit verslag is voor de grafische verwerking gebruik gemaakt van Matlab. De uitvoerfiles bevatten kolommen met getallen voorafgegaan door commentaar en specificatie van de kolommen. Met de Norton Editor kan dit worden weggehaald. De file moet dan een naam krijgen zónder extensie en kan binnen Matlab worden ingelezen met het commando: 'load naam.' (denk om de punt). De eerste kolom bevat altijd de tijd en met het commando 'plot(naam(:,1),naam(:,2))' worden de waarden in de tweede kolom uitgezet tegen de tijd.

We hebben geëxperimenteerd met het programma 3D-STUDIO (uitleenbaar bij het rekencentrum) waar men animaties mee kan maken. Dit programma heeft de mogelijkheid ASCII-files in te lezen die de verdraaiingen beschrijven. Het Fortranprogramma zou kunnen worden aangepast zodat de gegevens meteen in de goede vorm worden geschreven, hier zou dan wel veel (vrije) tijd in moeten worden gestoken. Dit lijkt ons weinig zinvol omdat dit per systeem opnieuw moet worden bekeken.

F Schematische aanpak modellering