

Simulatie met DADS van een systeem met gyroscopische momenten

Citation for published version (APA):

Uittenbogaart, J. R. (1989). *Simulatie met DADS van een systeem met gyroscopische momenten*. (DCT rapporten; Vol. 1989.025). Technische Universiteit Eindhoven.

Document status and date:

Gepubliceerd: 01/01/1989

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.

SIMULATIE MET DADS VAN EEN SYSTEEEM MET GYROSCOPISCHE MOMENTEN.

verslag 1^e WFW stage

J.R. Uittenbogaart

WFW 89.025

Samenvatting

Voor U ligt het verslag van mijn eerste stage, uitgevoerd binnen de vakgroep WFW. In deze stage is onderzoek gedaan naar een niet-convergerende oplossing, berekend door het programma DADS, van een simulatie van een systeem met gyroscopische momenten, dat eerder gesimuleerd is door een afstudeerder van WOC, sectie vervoerstechniek. Het model wordt opnieuw, stap voor stap opgebouwd om tenslotte tot een verbeterd model te komen.

Inhoudsopgave

Hoofdstuk 1: Inleiding	2
Hoofdstuk 2: Het programma DADS	3
Hoofdstuk 3: Het model van A. v.d. Riet	5
Hoofdstuk 4: Model vliegwiel	7
4.1 Het model	7
4.2.1 Modelleren met een Revolute–revolute joint	7
4.2.2 Modelleren met twee Revolute joints	8
Hoofdstuk 5: Model aandrijfeenheid	9
Hoofdstuk 6: Model voertuig	10
Hoofdstuk 7: Model voertuig op weg	12
Hoofdstuk 8: Conclusies	13
Literatuurlijst	14
Appendix A: Afleiding van de bewegingsvergelijkingen van het model vliegwiel	15
A1 Definitie	15
A2.1 De oriëntatie van \vec{e}^2 en \vec{e}^3 t.o.v. \vec{e}^1	16
A2.2 Bepaling van de hoeksnelheidsvector en de hoekversnellingsvector	17
A2.3 De bewegingsvergelijkingen van het vliegwiel	18
A2.4 De kwantitatieve weergave van de bewegingsvergelijkingen	20
Appendix B: Transformatie van de rotatiematrix naar Eulerparameters	21
Appendix C: Afleiding van de bewegingsvergelijkingen van het model aandrijfeenheid	23
C1 Afleiding van de bewegingsvergelijkingen	23
C2 Kwantitatieve weergave van de bewegingsvergelijkingen	25
Appendix D: De invoerfile van het model voertuig op weg	26

Hoofdstuk 1 : Inleiding

Binnen de sectie Vervoerstechniek van de vakgroep WOC is onderzoek gedaan naar de mogelijkheid om een voertuig aan te drijven met een hybride aandrijfeenheid. De aandrijfeenheid bestaat uit een motor en een vliegwiel. Bij het remmen wordt de kinetische energie van het voertuig gedeeltelijk omgezet in kinetische energie van het vliegwiel. De energie van het vliegwiel kan dan na het remmen, bijvoorbeeld bij het weer wegrijden, een gedeelte van de aandrijfenergie leveren. Een probleem bij deze constructie is dat door het vliegwiel, vooral in bochten, gyroscopische effecten optreden. In 1988 heeft André van der Riet als afstudeerwerk onderzoek gedaan naar de effecten van het vliegwiel op het weggedrag van het voertuig. Hierbij heeft hij simulaties uitgevoerd met het programma DADS (Dynamic Analysis and Design System). Bij berekeningen met een sneldraaiend vliegwiel bleek de oplossing niet te convergeren, grillig en dus waarschijnlijk fout. Dit is merkwaardig, temeer daar bij simulaties waarbij het berekende gyroscopische moment van het vliegwiel op de rest van de aandrijfeenheid als een uitwendige belasting werd gemodelleerd, de oplossing wel convergeerde. Dit vormde het probleem voor deze stage. Mijn opdracht luidde dan ook: Probeer door het stap voor stap opbouwen van het model en het vergelijken van analytische oplossingen met door DADS berekende oplossingen aan te geven waarom de simulaties van A. v.d Riet niet convergeren.

Hoofdstuk 2 : Het programma DADS

Het programmapakket DADS is een interactief programma, geschikt voor dynamische analyse van systemen van starre lichamen en continua. In deze stage is gebruik gemaakt van drie delen van het programma: de preprocessor (zie figuur 2.1), de processor DADS3D en de postprocessor.

In de preprocessor wordt het model opgebouwd. In deze stage is gebruik gemaakt van de volgende elementen:

Rigid body: definiëren van een star lichaam.

Initial condition: opgeven van beginpositie, -oriëntatie en -snelheid.

Revolute joint: creëren van een lijnscharnier.

Revolute-revolute joint: creëren van twee lijnscharnieren, waarvan de rotatie-assen of evenwijdig zijn of loodrecht op elkaar staan en die verbonden zijn door een massaloze staaf.

Driver: voorschrijven van de beweging van een verbindingselement.

RSDA: creëren van een rotatie veer-demper element.

TSDA: creëren van een translatie veer-demper element.

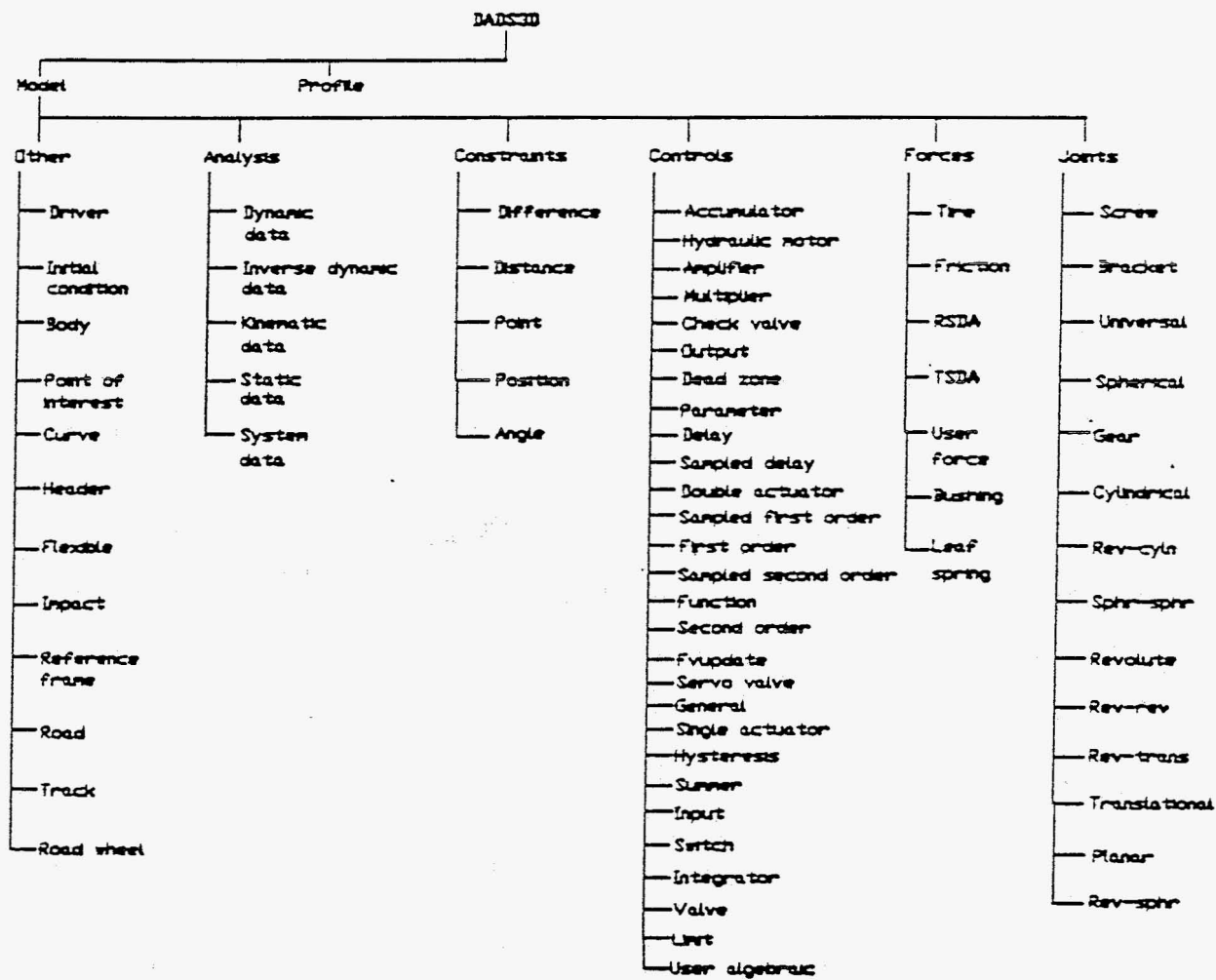
Bushing: creëren van een element dat een kracht langs of een moment om zijn eigen drie hoofdassen levert. De eigenschappen worden weergegeven als combinatie van vier veer- en vier dempingscoëfficiënten.

Verder worden hier gegevens die voor het rekenproces van belang zijn, zoals het integratie-interval, de maximaal toelaatbare integratiefout en het type analyse opgegeven. De preprocessor levert de voor het rekengedeelte benodigde file.

De processor DADS3D (het rekengedeelte) assembleert het model, bepaalt de oplossing van het probleem over het opgegeven interval en levert twee files: een file voor de postprocessor en een informatiefile die gegevens bevat omtrent de assemblage, het aantal vrijheidsgraden, de beginsnelheden en de verbruikte rekestijd.

Met de postprocessor kunnen plaatjes getekend worden van de door het rekengedeelte bepaalde oplossing.

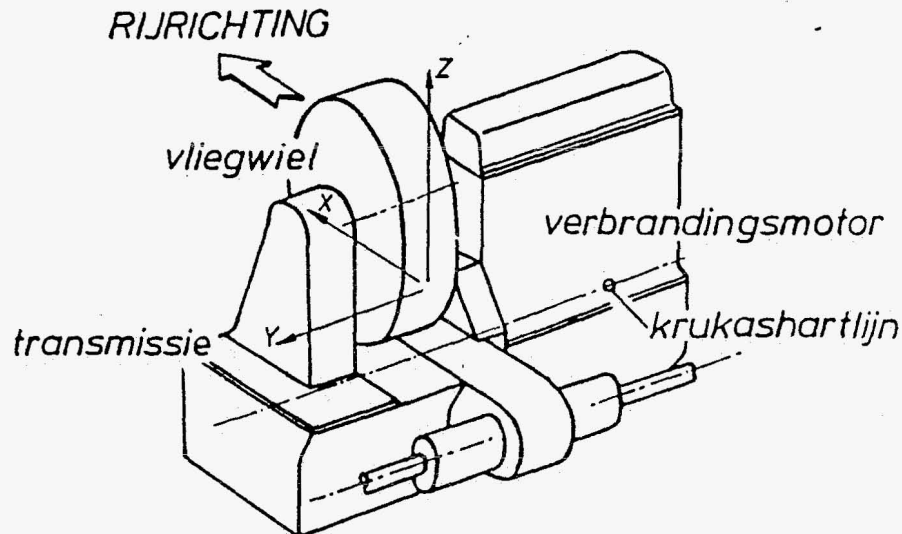
Een nadeel van DADS is dat de ingevoerde constructie niet weergegeven kan worden zoals bij veel andere programma's die gebaseerd zijn op de EEM.



figuur 2.1 : De structuur van de preprocessor

Hoofdstuk 3 : Het model van A. v.d. Riet

André van der Riet heeft als afstudeeropdracht een simulatie gemaakt van een Volvo 480 die van een hybride aandrijfeenheid voorzien is. De aandrijfeenheid bestaat uit een verbrandingsmotor, een vliegwiel en een transmissie (figuur 3.1)

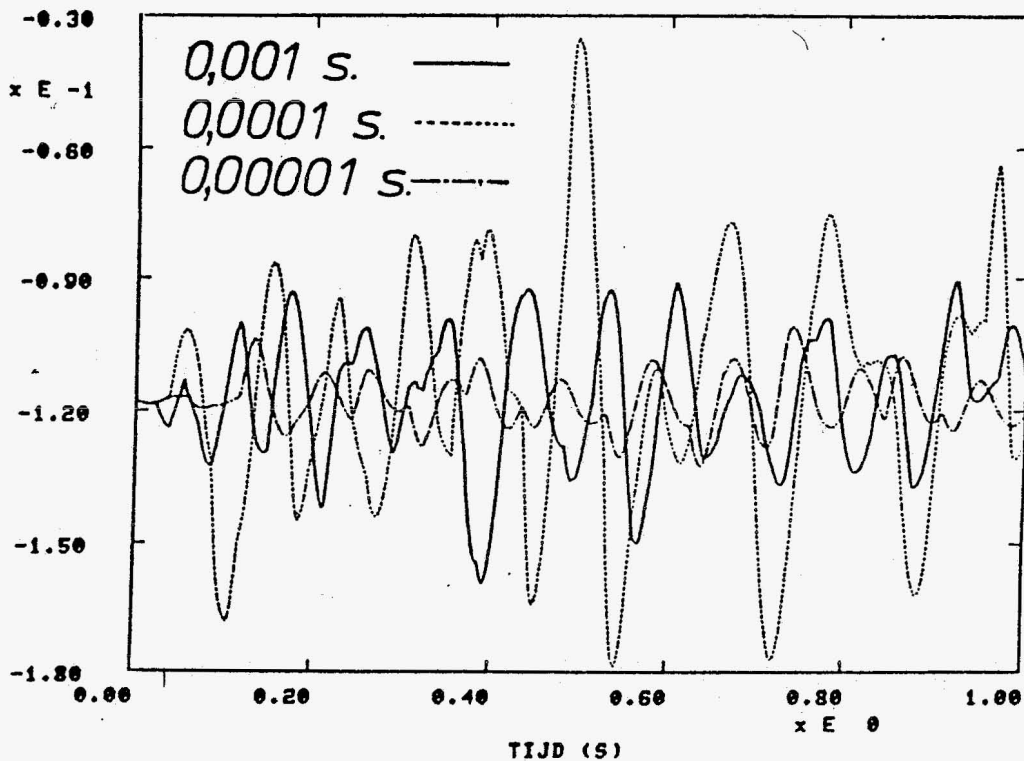


figuur 3.1 : De hybride aandrijfeenheid

De aandrijfeenheid is in het motorcompartiment opgehangen met twee conventionele en één vloeistofgevuld ophangrubber. Om de gyroscopische effecten te kunnen bestuderen is hij uit gegaan van de situatie dat het vliegwiel roteert t.o.v. de motor met 19000 omw/min ($=1990 \text{ rad/sec}$) en dat het voertuig een bocht ($r=6\text{m}$) doorloopt met een snelheid van 4.5 m/s ($\Rightarrow \omega=0.75 \text{ rad/s}$). Dit is als volgt gemodelleerd:

Een hulplichaam, te beschouwen als een t.o.v. het wegdek bewegend lichaam, doorloopt een cirkel. Aan dit hulplichaam zit via een Revolute joint het voertuig verbonden. De Revolute joint kan gezien worden als de rol-as van het voertuig. Om de rolstijfheid en -demping te karakteriseren is in de Revolute joint een RSDA element met veer- en dempingscoëfficiënt geplaatst. De motor is via drie Bushing elementen verbonden aan het voertuig. De trek- en afschuivingseigenschappen van de twee conventionele ophangrubbers zijn als veer- en dempingscoëfficiënten gemodelleerd. Het vloeistofgevlude ophangrubber is deels als Bushing en deels als USER-Defined Subroutine gemodelleerd. Met deze Subroutine kunnen zaken die niet met de standaard elementen gedefinieerd kunnen worden (b.v. niet-lineaire veren, van positie afhankelijke krachten) toch aan de simulatie toegevoegd worden door ze in Fortran te programmeren en aan de simulatie te koppelen. Het vliegwiel is aan de motor verbonden met een Revolute joint. De beweging van het hulplichaam t.o.v. de vaste wereld en van het vliegwiel t.o.v. de motor zijn met Driver elementen voorgeschreven.

De simulaties met een draaiend vliegwiel zijn mislukt. De oplossing bleek grillig en niet te convergeren. Zowel verkleinen van de maximaal toelaatbare integratiestap als verkleinen van de hoeksnelheid van het vliegwiel leverden geen convergerende oplossing. Zie bijvoorbeeld de rotatie om de dwarsas van de motor (figuur 3.2).



figuur 3.2 : De hoekverdraaiing van de motor om de dwarsas

Wanneer het gyroscopische moment van het vliegwiel als een uitwendige belasting op de motor (m.b.v. een USER-Defined Subroutine), verliepen de simulaties wel goed. Met de tot mijn beschikking staande gegevens (het afstudeerverslag van A. v.d.Riet) bleek het niet mogelijk om de door hem geconstateerde problemen te reproduceren. Toen is besloten om zelf een model te maken waarbij zo veel mogelijk met de wél bruikbare gegevens (massa's, veerconstanten enz.) gewerkt zou worden. Het model is in vier stappen opgebouwd:

- 1 : Model vliegwiel: roterend vliegwiel doorloopt een bocht (hoofdstuk 3).
- 2 : Model aandrijfeenheid: aandrijfeenheid met roterend vliegwiel doorloopt een bocht (hoofdstuk 5).
- 3 : Model voertuig: voertuig, met daarin opgehangen de aandrijfeenheid, doorloopt een bocht (hoofdstuk 6).
- 4 : Model voertuig op weg: als model voertuig, voertuig kan nu roteren om zijn langsas (hoofdstuk 7).

Hoofdstuk 4 : Model vliegwiel

4.1 Het model

Het roterende vliegwiel dat een bocht doorloopt is gemodelleerd als een massaloze staaf die roteert met hoeksnelheid ω_z om O^0 , met daaraan een vliegwiel dat roteert om de staaf in O met hoeksnelheid ω_v , zoals in figuur 4.1



figuur 4.1 : Het model vliegwiel

De bewegingsvergelijkingen voor dit model (afleiding zie Appendix A) zijn:

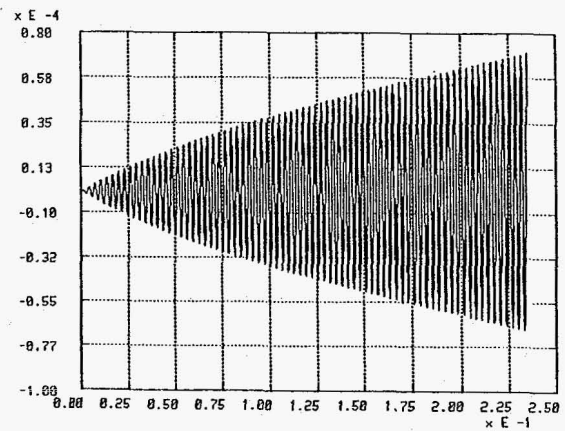
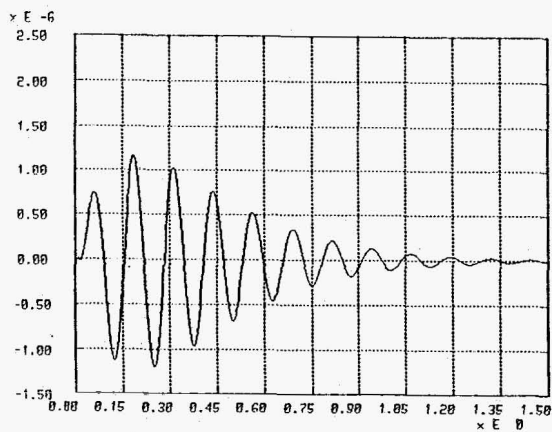
$$\vec{M} = \vec{e}^{2T} \begin{bmatrix} 543.27 \\ 0 \\ 0 \end{bmatrix} \quad \text{en} \quad \vec{F} = \vec{e}^{2T} \begin{bmatrix} 0 \\ 64.125 \\ 0 \end{bmatrix}$$

Dit model is met DADS op twee manieren gemodelleerd:

- 1) met een Revolute-revolute joint en begincondities
- 2) met twee Revolute joints en bijbehorende drivers

4.2.1 Modelleren met een Revolute-revolute joint

Voor dit model zijn de rotatie-assen van de Revolute-revolute joint loodrecht op elkaar gekozen. Aangezien het voorschrijven van de relatieve hoeken als functie van de tijd met Drivers bij dit element onmogelijk is, wordt het model losgelaten met begin-rotatiesnelheden voor het vliegwiel (afleiding in Appendix B). De resultaten zijn voor verschillende waarden van ω_v .



figuur 4.2 A en B : De verplaatsing in z-richting voor ω_v is 50 resp. 1990

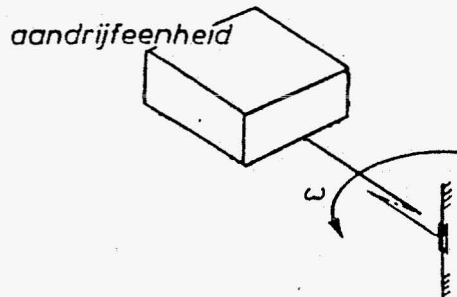
Ondanks dat beweging in z-richting voor het vliegwiel onderdrukt is, is er een sinusvormige beweging met toenemende amplitude in deze richting. De frequentie van deze beweging is gelijk aan de hoekfrequentie van het vliegwiel. De oorzaak van deze beweging is waarschijnlijk een fout die zich in de numerieke integratie voortplant.

4.2.2 Modelleren met twee Revolute joints

In dit model is de Revolute-revolute joint van 4.2.1 vervangen door twee Revolute joints, verbonden door een staaf. De staaf heeft een t.o.v. het vliegwiel verwaarloosbare massa (0.05 %). Er mag derhalve aangenomen worden dat deze staaf geen invloed heeft op het dynamische gedrag van het systeem. De hoeksnelheden ω_z en ω_v worden met behulp van Drivers voorgeschreven. Voor dit model doen we een inverse-dynamische analyse, omdat alle vrijheidsgraden òf onderdrukt òf voorgeschreven zijn. De numerieke oplossing voldoet nu wel aan de analytische oplossing.

Hoofdstuk 5 : Model aandrijfeenheid

De roterende aandrijfeenheid is gemodelleerd als een motor die met hoeksnelheid ω_z roteert om O^0 , met daaraan een vliegwiel dat met hoeksnelheid ω_v roteert om O .



figuur 5.1 : Het model aandrijfeenheid

De analytische oplossing is bepaald in Appendix C. Wanneer we de numerieke en de analytische resultaten met elkaar vergelijken krijgen we de volgende tabel

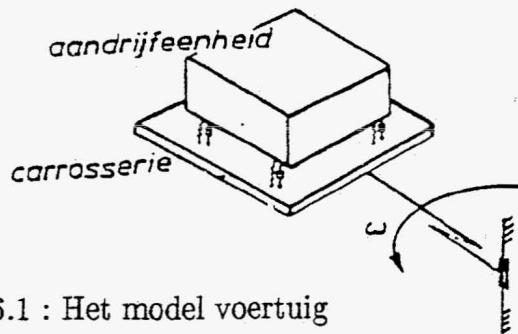
	Model	DADS		Model	DADS
2M_x	543.27	543.27002	1M_x	553.85	553.85
2M_y	0	<1E-10	1M_y	-4.046E-3	-0.28 < 1M_y < 0.17
2M_z	0	<1E-10	1M_z	0	<1E-15

	Model	DADS		Model	DADS
2F_x	0.81225	0.81225	1F_x	0.81225	0.81225
2F_y	-65.3327	-65.3327	1F_y	-844.283	-844.2827
2F_z	0	<1E-10	1F_z	0	<1E-10

De numerieke resultaten voldoen weer aan de analytische oplossing.

Hoofdstuk 6 : Model voertuig

Het voertuig is gemodelleerd als een frame met daarin opgehangen de aandrijfeenheid. Het frame doorloopt een bocht met hoeksnelheid ω_z en het vliegwiel roteert weer t.o.v. de rest van de aandrijfeenheid met hoeksnelheid ω_v .

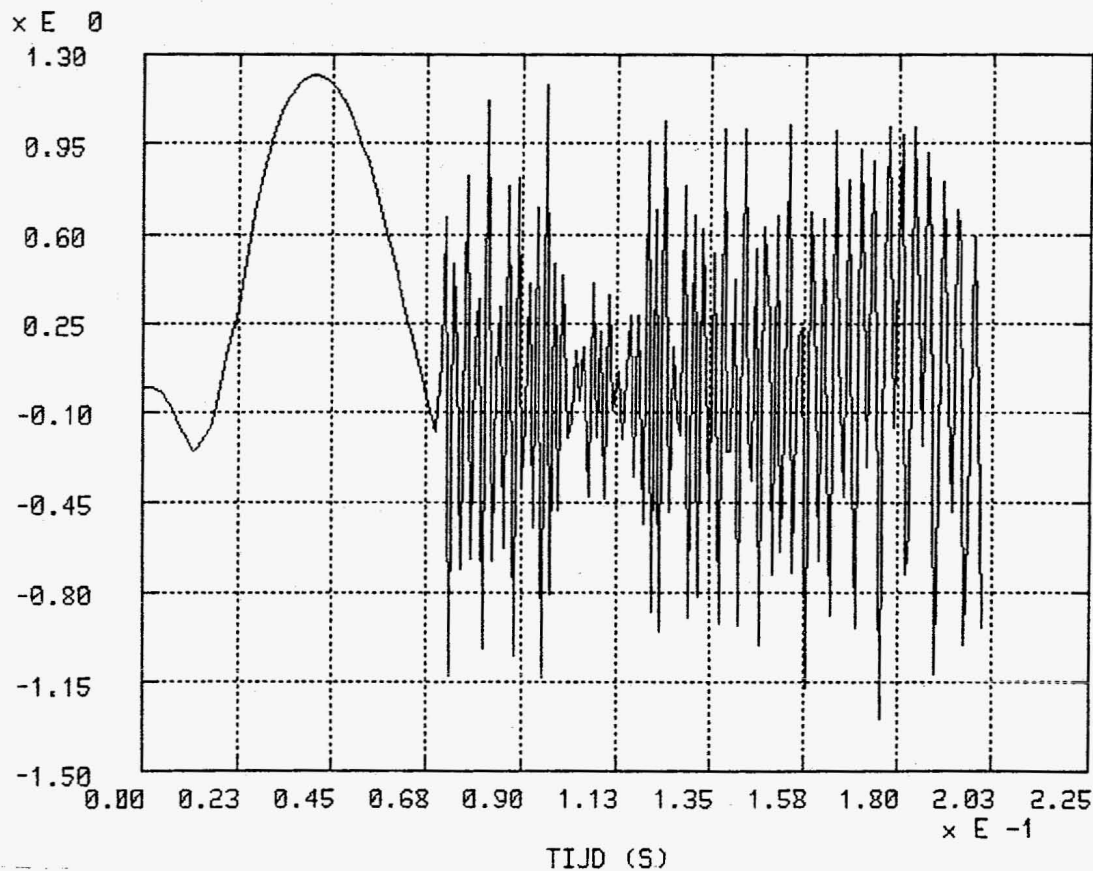


figuur 6.1 : Het model voertuig

De analytische oplossing is voor dit model met de hand zeer moeilijk te bepalen, omdat de motor zes vrijheidsgraden heeft.

De drie ophangrubbers waren door A. v.d. Riet gedeeltelijk gemodelleerd met Bushing elementen. De niet-lineaire dempingseigenschappen waren in een USER-Defined Subroutine gemodelleerd. Ter vereenvoudiging zijn de dempingscoëfficiënten van de vloeistofgevulde ophangrubbers constant genomen en ook in de Bushings gedeclareerd.

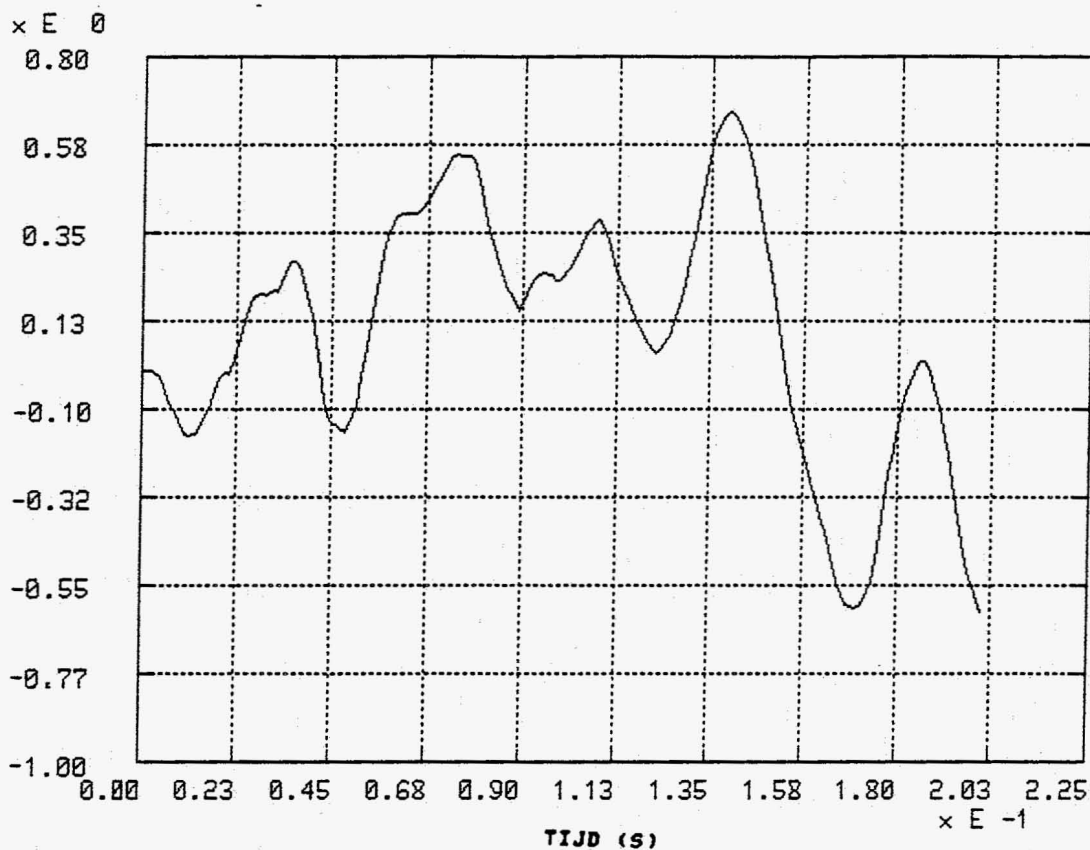
De simulaties leverden dit resultaat:



figuur 6.2 : De hoekverdraaiing van de motor om de dwarsas

De oplossing is fout vanaf $t=0.07$ s. Verkleinen van de maximale integratiestap zorgt voor een verschuiving van dit tijdstip, maar lost de problemen niet op. Opvallend is dat op het tijdstip dat de oplossing numeriek gaat oscilleren, de totale energie in het systeem met een factor 100 groter wordt en dat de locale hoeksnelheid van het vliegwiel van teken verandert.

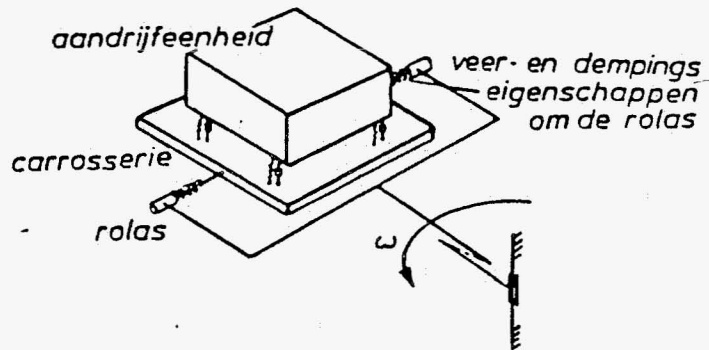
Vervolgens zijn de drie Bushings vervangen door drie maal drie overeenkomstige TSDA's. De simulaties vertoonden nu geen problemen (figuur 6.3).



figuur 6.3 : De hoekverdraaiing van de motor om de dwarsas

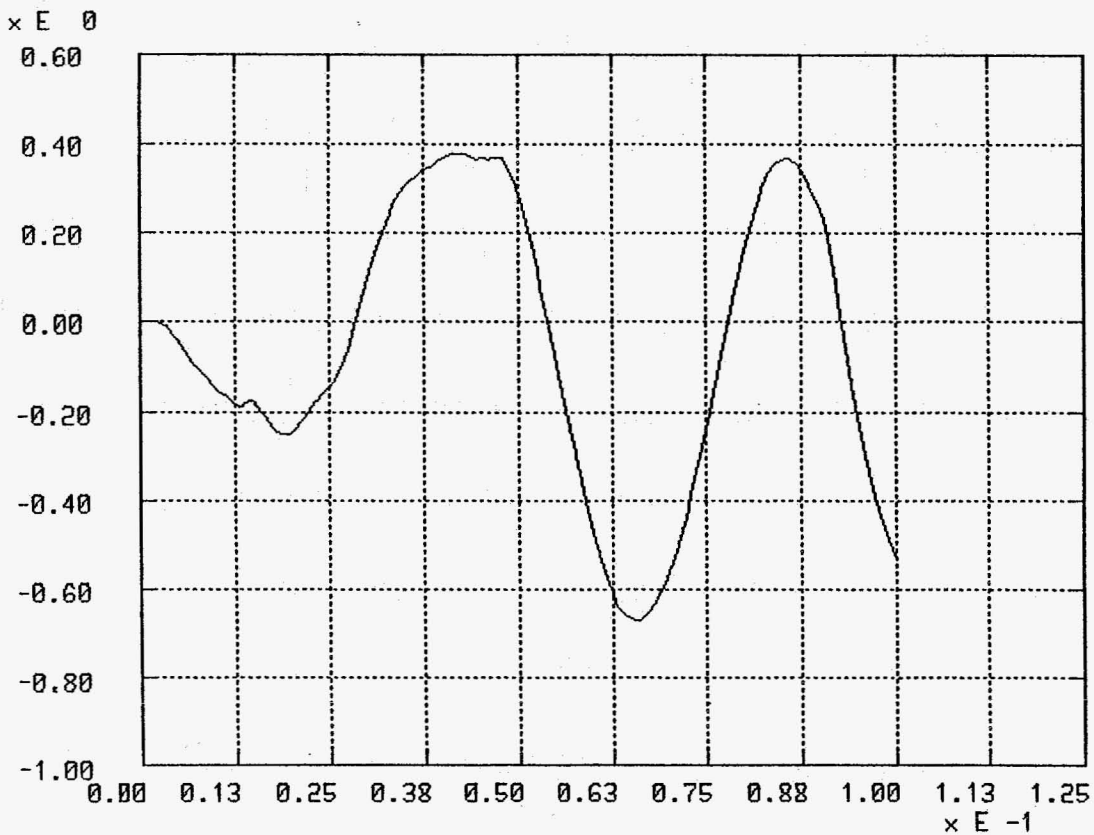
Hoofdstuk 7 : Model voertuig op weg

Het model is identiek aan het model van hoofdstuk 6, met de uitbreiding dat het frame kan roteren t.o.v. de weg (figuur 7.1). Deze zogenaamde rolei-eigenschappen zijn gemodelleerd met een TSDA waarin de rolstijfheid en -dempingscoëfficiënt zijn gedeclareerd.



figuur 7.1 : Het model voertuig op weg

De invoerfile van dit model is gegeven in Appendix D. De simulaties van DADS vertoonden geen problemen.



figuur 7.2 : De hoekverdraaiing van de motor om de dwarsas

Hoofdstuk 8 : Conclusies

De door André van der Riet geconstateerde divergentie van de oplossing van de simulatie is door mij niet aangetroffen. De oorzaak hiervan is dat ik met de beschikbare invoergegevens het door hem doorgerekende model niet kon reproduceren. Bij de simulaties van verwante problemen die door mij zijn uitgevoerd, zijn toch enige andere problemen naar voren gekomen. Allereerst vertoonde het Revolute-revolute element bij grote beginrotatiesnelheden numerieke oscillaties in een onderdrukte bewegingsrichting. Verder is gebleken dat het modelleren van ophangrubbers, het (bij hoge toerentallen) beter is drie veer-demper elementen in onderling loodrechte richtingen dan Bushing elementen te gebruiken.

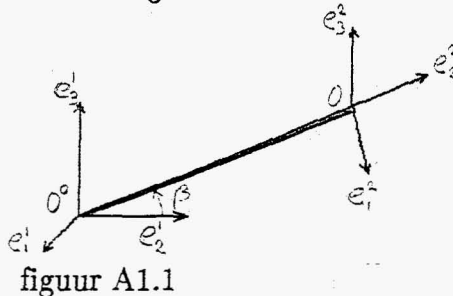
Literatuurlijst

- [1] Sauren, A.A.H.J.
Multibody Dynamica
collegedictaat 4659, TU Eindhoven, 1988
- [2] Wittenburg, J.
Dynamics of systems of rigid bodies
B.G. Teubner, Stuttgart, 1977
- [3] Van der Riet, A.
Simulatieberekeningen aan de ophanging van de hybride aandrijfteenheid
afstudeerverslag, TU Eindhoven, 1988
- [4] Haug, E.J.
Computer aided kinematics and dynamics of mechanical systems
The University of Iowa, 1985
- [5] DADS User's Manual
revision 5.0, 1988

Appendix A : Afleiding van de bewegingsvergelijkingen van het model vliegwiel

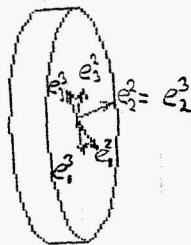
A.1 Definitie

We voeren een inertiaalbasis \vec{e}^1 (orthogonaal, rechtsdraaiend) en een oorsprong O^0 in. In het vlak opgespannen door \vec{e}_1^1 en \vec{e}_2^1 roteert een starre, massaloze as O^0O met lengte r om O^0 . De vectorbasis \vec{e}^2 zit vast aan de as in O zodanig dat $\vec{e}_3^2 = \vec{e}_3^1$. De basis \vec{e}^2 ontstaat uit \vec{e}^1 door \vec{e}^1 te roteren over een hoek β om \vec{e}_3^1 . Als $\beta=0$ geldt $\vec{e}^1 = \vec{e}^2$. Voor de positievector \vec{r}_O geldt altijd $\vec{r}_O = r \cdot \vec{e}_2^2$.



figuur A1.1

In O draait het vliegwiel om de as O^0O , dus om de \vec{e}_2^2 -as. Het massamiddelpunt van het vliegwiel bevindt zich in O . De vectorbasis \vec{e}^3 zit vast aan het massamiddelpunt van het vliegwiel en ontstaat uit \vec{e}^2 door \vec{e}^2 te roteren over een hoek α om \vec{e}_2^2 . Als $\alpha=0$ geldt $\vec{e}^2 = \vec{e}^3$.



figuur A1.2

De afleiding van de bewegingsvergelijkingen van het vliegwiel wordt in 4 delen gesplitst. In A2.1 worden de oriëntaties van \vec{e}^3 en \vec{e}^2 ten opzichte van \vec{e}^1 beschreven, in A2.2 worden de hoeksnelheidsvector en de hoekversnellingsvector van het vliegwiel afgeleid, die in A2.3 ingevuld worden in de bewegingsvergelijkingen van het vliegwiel. In A2.4 worden tenslotte de in H2 genoemde gegevens ingevuld in de bewegingsvergelijkingen.

A2.1 De oriëntatie van \vec{e}_2 en \vec{e}_3 t.o.v. \vec{e}_1 .

Voor de beschrijving van de oriëntatie van \vec{e}_2 en \vec{e}_3 ten opzichte van \vec{e}_1 maken we gebruik van Euler-hoeken. Deze zijn als volgt gedefinieerd:

- rotatie van e^1 om de e_3^1 -as levert e'
- rotatie van e' om de e_1^1 -as levert e''
- rotatie van e'' om de e_3^1 -as levert e^2

Voor \vec{e}_2 t.o.v. \vec{e}_1 geldt in tensornotatie

$$\vec{e}_2^Y = \underline{R}^\beta \cdot \vec{e}_1^Y$$

De matrixrepresentatie van \underline{R}^β is met gebruikmaking van $\underline{R}^\beta = \vec{e}_2^Y \cdot \vec{e}_1^Y$ te schrijven als

$$\underline{R}^\beta = \vec{e}_1^Y \cdot \underline{R}^\beta \cdot \vec{e}_1^Y = \vec{e}_1^Y \cdot \vec{e}_2^Y \cdot \vec{e}_1^Y = \vec{e}_1^Y \cdot \vec{e}_2^Y$$

dus

$$\vec{e}_2^Y = \vec{e}_1^Y \cdot \underline{R}^\beta$$

met

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

De Euler-hoeken voor \vec{e}_2 t.o.v. \vec{e}_1 zijn dus $[\beta, 0, 0]^Y$

Wanneer we de oriëntatie van \vec{e}_3 t.o.v. \vec{e}_1 willen beschrijven met (onder andere) de eerder genoemde β , moeten we een hulpbasis \vec{e}_2^{\prime} introduceren, die ontstaat uit \vec{e}_2 door

\vec{e}_2 te roteren over $\pi/2$ om de \vec{e}_1^1 -as.

$$\vec{e}_2^{\prime Y} = \underline{R}^\gamma \cdot \vec{e}_2^Y = \vec{e}_2^Y \cdot \underline{R}^\gamma \cdot \vec{e}_2^Y = \vec{e}_2^Y \cdot \underline{R}^\gamma$$

met

$$\underline{R}^\gamma = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

De vectorbasis \vec{e}_3 ontstaat uit \vec{e}_2^{\prime} door \vec{e}_2^{\prime} te roteren over een hoek α om \vec{e}_3^{\prime}

$$\vec{e}_3^Y = \underline{R}^\alpha \cdot \vec{e}_2^{\prime Y} = \vec{e}_2^{\prime Y} \cdot \underline{R}^\alpha \cdot \vec{e}_2^{\prime Y} = \vec{e}_2^{\prime Y} \cdot \underline{R}^\alpha$$

met

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

De Euler-hoeken voor \vec{e}_3 t.o.v. \vec{e}_1 zijn dus $[\beta, \pi/2, \alpha]^T$

De rotatie van \vec{e}_3 t.o.v. \vec{e}_1 kan beschreven worden met de matrix \underline{R} waarvoor geldt

$$\vec{e}_3^T = \vec{e}_2'^T \cdot \underline{R}^\alpha = \vec{e}_2^T \cdot \underline{R}^\gamma \underline{R}^\alpha = \vec{e}_1^T \cdot \underline{R}^\beta \underline{R}^\gamma \underline{R}^\alpha = \vec{e}_1^T \cdot \underline{R}$$

met

$$\underline{R} = \begin{bmatrix} \cos \alpha \cos \beta & -\sin \alpha \cos \beta & \sin \beta \\ \cos \alpha \sin \beta & -\sin \alpha \sin \beta & -\cos \beta \\ \sin \alpha & \cos \alpha & 0 \end{bmatrix}$$

A2.2 Bepaling van de hoeksnelheidsvector en de hoekversnellingsvector

Het vliegwiel roteert om de \vec{e}_3 en de \vec{e}_1 . De hoeksnelheidsvector van het vliegwiel is dus

$$\begin{aligned} \vec{\omega} &= \dot{\alpha} \vec{e}_3 + \dot{\beta} \vec{e}_1 = \dot{\alpha} \vec{e}_3' + \dot{\beta} \vec{e}_3 \\ &= \vec{e}_2'^T \begin{bmatrix} 0 \\ 0 \\ \dot{\alpha} \end{bmatrix} + \vec{e}_2^T \begin{bmatrix} 0 \\ 0 \\ \dot{\beta} \end{bmatrix} \end{aligned}$$

Transformeren naar de vectorbasis \vec{e}_2

$$\vec{\omega} = \vec{e}_2^T \underline{R}^\gamma \begin{bmatrix} 0 \\ 0 \\ \dot{\alpha} \end{bmatrix} + \vec{e}_2^T \begin{bmatrix} 0 \\ 0 \\ \dot{\beta} \end{bmatrix} = \vec{e}_2^T \begin{bmatrix} 0 \\ -\dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \vec{e}_2^T \cdot {}_2\omega$$

Voor het bepalen van $\dot{\vec{\omega}}$ wordt $\vec{\omega}$ eerst bepaald t.o.v. de vaste basis \vec{e}_1 , omdat differentiëren binnen deze basis gemakkelijker is.

$$\vec{\omega} = \vec{e}_1^T \underline{R}^\beta \cdot {}_2\omega = \vec{e}_1^T \begin{bmatrix} \dot{\alpha} \cdot \sin \beta \\ -\dot{\alpha} \cdot \cos \beta \\ \dot{\beta} \end{bmatrix} = \vec{e}_1^T \cdot {}_1\omega$$

Differentiëren levert

$$\dot{\vec{\omega}} = \frac{d}{dt} (\vec{e}^{1T} \dot{\vec{\omega}}) = \vec{e}^{1T} \begin{bmatrix} \ddot{\alpha} \cdot \sin\beta + \dot{\alpha} \cdot \beta \cdot \cos\beta \\ -\ddot{\alpha} \cdot \cos\beta + \dot{\alpha} \cdot \beta \cdot \cos\beta \\ \ddot{\beta} \end{bmatrix} = \vec{e}^{1T} \dot{\vec{\omega}}^1$$

Geschreven t.o.v. de vectorbasis \vec{e}^2

$$\dot{\vec{\omega}} = \vec{e}^{2T} \underline{R}^{\beta} \dot{\vec{\omega}} = \vec{e}^{2T} \begin{bmatrix} \dot{\alpha} \cdot \beta \\ -\ddot{\alpha} \\ \ddot{\beta} \end{bmatrix} = \vec{e}^{2T} \dot{\vec{\omega}}^2$$

A2.3 De bewegingsvergelijkingen van het vliegwiel

De bewegingsvergelijkingen van het vliegwiel worden bepaald met de impulsmomentstelling en de tweede wet van Newton. Beide vergelijkingen worden geschreven t.o.v. de basis \vec{e}^2 omdat de resultaten dan gemakkelijk te interpreteren zijn.

De impulsmomentstelling voor het vliegwiel t.o.v. het vaste punt O^o luidt

$$\vec{M}_{O^o} = \dot{\vec{L}}_{O^o}$$

met

$$\vec{L}_{O^o} = m \cdot (\vec{r}_v \times \dot{\vec{r}}_v) + J_v \cdot \dot{\vec{\omega}}_v$$

J_v is de traagheidstensor van het vliegwiel t.o.v. zijn zwaartepunt. Invullen levert

$$\vec{M}_{O^o} = m (\vec{r}_v \times \ddot{\vec{r}}_v) + \dot{\vec{\omega}}_v \times (J_v \cdot \dot{\vec{\omega}}_v) + J_v \cdot \ddot{\vec{\omega}}_v$$

geschreven t.o.v. de vectorbasis \vec{e}^2

$$\vec{e}^{2T} \dot{\vec{M}} = \vec{e}^{2T} \{ m (\vec{r}_v \times \ddot{\vec{r}}_v) + \dot{\vec{\omega}}_v \times J_{v2} \cdot \dot{\vec{\omega}}_v + J_{v2} \cdot \ddot{\vec{\omega}}_v \}$$

De tweede wet van Newton luidt

$$\vec{F} = \frac{d}{dt} (m \dot{\vec{r}}) = m \frac{d}{dt} \dot{\vec{r}} \quad (\text{mits } m = \text{constant})$$

geschreven t.o.v. \vec{e}^2 wordt dit

$$\vec{e}^{2T} \dot{\vec{F}} = m \ddot{\vec{r}}_v$$

Voor de positievector \vec{r}_v geldt (zie figuur A1.1)

$$\vec{r}_v = \vec{e}^{2T} \begin{bmatrix} 0 \\ r \\ 0 \end{bmatrix} = \vec{e}^{2T} \dot{\vec{r}}_v$$

en geschreven t.o.v. de vectorbasis \vec{e}^1

$$\vec{r}_v = \vec{e}^1 \underline{R}^\beta \vec{r} = \vec{e}^1 \begin{bmatrix} -r \cdot \sin \beta \\ r \cdot \cos \beta \\ 0 \end{bmatrix} = \vec{e}^1 \vec{r}$$

Tweemaal differentiëren levert (mits $r = \text{constant}$)

$$\ddot{\vec{r}}_v = \vec{e}^1 \begin{bmatrix} r \cdot \ddot{\beta} \cdot \sin \beta - r \cdot \dot{\beta} \cdot \cos \beta \\ -r \cdot \ddot{\beta} \cdot \cos \beta - r \cdot \dot{\beta} \cdot \sin \beta \\ 0 \end{bmatrix} = \vec{e}^1 \ddot{\vec{r}}$$

geschreven t.o.v. de vectorbasis \vec{e}^2

$$\vec{r}_v = \vec{e}^2 \underline{R}^\beta \vec{r} = \vec{e}^2 \begin{bmatrix} -r \cdot \dot{\beta} \\ -r \cdot \beta^2 \\ 0 \end{bmatrix}$$

\underline{J}_{v2} is de matrixrepresentatie van J_v t.o.v. de vectorbasis \vec{e}^2 . Er geldt $J_v = \vec{e}^2 \underline{J}_{v2} \vec{e}^2$. De matrixrepresentatie \underline{J}_{v3} van J_v t.o.v. de vectorbasis \vec{e}^3 is een diagonaalmatrix, omdat de richting van de hoofdtraagheidsassen van het vliegwiel samenvallen met de basisvectoren van \vec{e}^3 . Bovendien zijn de twee traagheidsmomenten om de \vec{e}_1^3 en \vec{e}_2^3

gelijk (door symmetrie), dus

$$\underline{J}_{v3} = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_x & 0 \\ 0 & 0 & J_z \end{bmatrix}$$

Transformeren van \underline{J}_{v2} naar \underline{J}_{v3} gaat als volgt

$$J_v = \vec{e}^2 \underline{J}_{v2} \vec{e}^2 = \vec{e}^3 \underline{J}_{v3} \vec{e}^3 = \vec{e}^2 \underline{R}^\gamma \underline{R}^\alpha \underline{J}_{v3} \underline{R}^{\alpha^T} \underline{R}^{\gamma^T} \vec{e}^2$$

$$\Rightarrow \underline{J}_{v2} = \underline{R}^\gamma \underline{R}^\alpha \underline{J}_{v3} \underline{R}^{\alpha^T} \underline{R}^{\gamma^T}$$

invullen levert

$$\underline{J}_{v2} = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_z & 0 \\ 0 & 0 & J_x \end{bmatrix}$$

dit alles invullen in de impulsmomentstelling en in de tweede wet van Newton geeft respectievelijk

$${}^2\vec{M} = \begin{bmatrix} {}^2M_x \\ {}^2M_y \\ {}^2M_z \end{bmatrix} = \begin{bmatrix} \dot{\alpha} \cdot \beta \cdot J_z \\ -\dot{\alpha} \cdot J_z \\ \ddot{\beta} \cdot (m \cdot r^2 + J_x) \end{bmatrix}$$

$${}^2\tilde{\mathbf{F}} = \begin{bmatrix} {}^2F_x \\ {}^2F_y \\ {}^2F_z \end{bmatrix} = \begin{bmatrix} -m \cdot r \cdot \ddot{\beta} \\ -m \cdot r \cdot \dot{\beta}^2 \\ 0 \end{bmatrix}$$

A2.4 De kwantitatieve weergave van de bewegingsvergelijkingen

Met gebruikmaking van $r=6$ dus

$${}^2\tilde{\mathbf{r}} = \begin{bmatrix} 0 \\ 6 \\ 0 \end{bmatrix}$$

en verder de in Appendix D genoemde gegevens volgt voor de bewegingsvergelijkingen

$$\begin{bmatrix} {}^2M_x \\ {}^2M_y \\ {}^2M_z \end{bmatrix} = \begin{bmatrix} 543.27 \\ 0 \\ 0 \end{bmatrix} \quad \text{en} \quad \begin{bmatrix} {}^2F_x \\ {}^2F_y \\ {}^2F_z \end{bmatrix} = \begin{bmatrix} 0 \\ 64.125 \\ 0 \end{bmatrix}$$

Appendix B : Transformatie van de rotatiematrix naar Eulerparameters

In DADS moeten de begincondities, voor wat betreft de beginhoeken in het inertiaalstelsel, opgegeven worden in Eulerparameters en hun eerste afgeleide. Voor de oriëntatie van \vec{e}_3^r ten opzichte van \vec{e}_1^r geldt (zie Appendix A)

$\vec{e}_3^r = \vec{e}_1^r \underline{R}$ met

$$\underline{R} = \begin{bmatrix} \cos\alpha \cos\beta & -\sin\alpha \cos\beta & \sin\beta \\ \cos\alpha \sin\beta & -\sin\alpha \sin\beta & -\cos\beta \\ \sin\alpha & \cos\alpha & 0 \end{bmatrix}$$

Volgens [2] is de bepaling van de Eulerparameters als volgt

$$q_0^2 = (\text{tr}\underline{R} + 1)/4$$

$$q_i^2 = (2R_{ii} - \text{tr}\underline{R} + 1)/4 \quad i=1, 2, 3$$

invullen levert

$$q_0 = \pm 1/2 \sqrt{\cos\alpha \cdot \cos\beta - \sin\alpha \cdot \sin\beta + 1}$$

$$q_1 = \pm 1/2 \sqrt{\cos\alpha \cdot \cos\beta + \sin\alpha \cdot \sin\beta + 1}$$

$$q_2 = \pm 1/2 \sqrt{-\cos\alpha \cdot \sin\beta - \sin\alpha \cdot \cos\beta + 1}$$

$$q_3 = \pm 1/2 \sqrt{-\cos\alpha \cdot \cos\beta + \sin\alpha \cdot \sin\beta + 1}$$

volgens [4] zijn de tekens van q_1 , q_2 en q_3 te bepalen, als het teken van q_0 gekozen is.

op $t=t_0$ geldt $\alpha=\beta=0$, dan

$$q_0 = \pm 1/2 \sqrt{2}$$

kiezen we $q_0 = +1/2 \sqrt{2}$

dan is

$$q_1 = 1/2 \sqrt{2}$$

$$q_2 = 0$$

$$q_3 = 0$$

Voor de hoeksnelheidsvector $\vec{\omega}$ van \vec{e}^3 t.o.v. \vec{e}^1 geldt

$$\vec{\omega} = \vec{e}^3 \begin{bmatrix} 0 \\ \beta \\ \dot{\alpha} \end{bmatrix}$$

Volgens [1] is de bepaling van de eerste afgeleide van de Eulerparameters als volgt

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = 1/2 \begin{bmatrix} 0 & \vdots & -\underline{\omega}^T \\ \dots & \dots & \dots \\ \underline{\omega} & \vdots & -\underline{\Omega} \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

met

$$\underline{\Omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

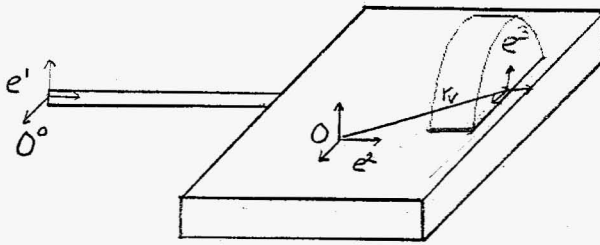
invullen levert

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1/4 \sqrt{2} (\beta - \dot{\alpha}) \\ 1/4 \sqrt{2} (\beta + \dot{\alpha}) \end{bmatrix}$$

Appendix C : Afleiding van de bewegingsvergelijkingen van het model aandrijfteenheid

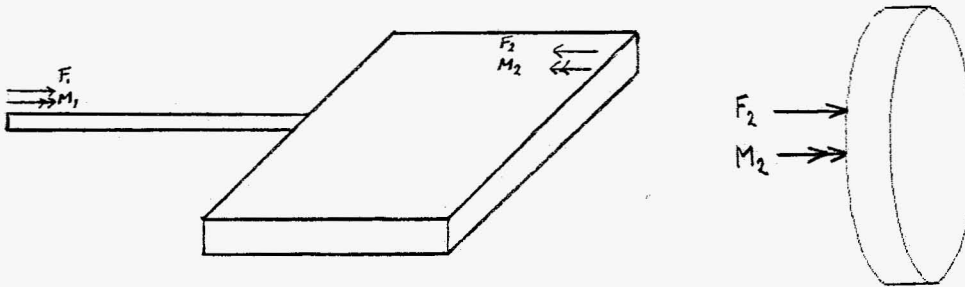
C.1 Afleiding van de bewegingsvergelijkingen

Analoog aan appendix A voeren we een inertiaalbasis \vec{e}^1 en een vectorbasis \vec{e}^2 in. Het massamiddelpunt van de motor bevindt zich in O. De vectorbasis \vec{e}^3 zit weer vast aan het vliegwiel, maar het massamiddelpunt van het vliegwiel zit nu niet in O.



figuur C1.1

Na "losmaken" van het vliegwiel kunnen de bewegingsvergelijkingen voor de motor en voor het vliegwiel opgesteld worden.



figuur C1.2

Voor het vliegwiel zijn de impulsmomentstelling en de tweede wet van Newton respectievelijk

$$\vec{M}_2 + \vec{r}_2 \times \vec{F}_2 = m_v (\vec{r}_v \times \ddot{\vec{r}}_v) + \vec{\omega}_v \times (J_v \cdot \vec{\omega}_v) + J_v \cdot \dot{\vec{\omega}}_v$$

$$\vec{F}_2 = m_v \ddot{\vec{r}}_v$$

Evenzo voor de motor

$$\vec{M}_1 + \vec{r}_1 \times \vec{F}_1 - \vec{M}_2 - \vec{r}_2 \times \vec{F}_2 = m_m (\vec{r}_m \times \vec{r}_m) + \vec{\omega}_m \times (J_m \cdot \vec{\omega}_m) + J_m \cdot \dot{\vec{\omega}}_m$$

$$\vec{F}_1 - \vec{F}_2 = m_m \vec{r}_m$$

Voor \vec{r}_v , \vec{r}_m , \vec{r}_1 en \vec{r}_2 geldt

$$\vec{r}_v = \vec{e}^2 \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}, \quad \vec{r}_m = \vec{e}^2 \begin{bmatrix} 0 \\ r \\ 0 \end{bmatrix}, \quad \vec{r}_1 = \vec{e}^2 \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \vec{r}_2 = \vec{r}_v$$

en voor de hoeksnelheidsvectoren

$$\vec{\omega}_v = \vec{e}^2 \begin{bmatrix} 0 \\ -\dot{\alpha} \\ \beta \end{bmatrix}, \quad \vec{\omega}_m = \vec{e}^2 \begin{bmatrix} 0 \\ 0 \\ \beta \end{bmatrix}$$

Differentiëren van \vec{r}_v en \vec{r}_m gaat analoog aan Appendix A. Het resultaat

$$\dot{\vec{r}}_v = \vec{e}^2 \begin{bmatrix} -\beta^2 r_x \\ -\beta^2 r_y \\ 0 \end{bmatrix}, \quad \dot{\vec{r}}_m = \vec{e}^2 \begin{bmatrix} 0 \\ -\beta^2 r \\ 0 \end{bmatrix} \quad (\text{mits } \beta = \text{constant})$$

idem voor $\dot{\vec{\omega}}_v$ en $\dot{\vec{\omega}}_m$

$$\dot{\vec{\omega}}_v = \vec{e}^2 \begin{bmatrix} \dot{\alpha} \beta \\ 0 \\ 0 \end{bmatrix}, \quad \dot{\vec{\omega}}_m = \vec{e}^2 \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (\text{mits } \beta = \text{constant})$$

De traagheidsmatrices van vliegwiel en motor t.o.v. \vec{e}^2 zijn respectievelijk

$$\underline{J}_v = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_z & 0 \\ 0 & 0 & J_x \end{bmatrix}, \quad \underline{J}_m = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{xy} & J_{yy} & J_{yz} \\ J_{xz} & J_{yz} & J_{zz} \end{bmatrix}$$

invullen in de bewegingsvergelijkingen

$$\vec{M}_2 = \vec{e}^2 \begin{bmatrix} \dot{\alpha} \beta J_z \\ 0 \\ 0 \end{bmatrix}, \quad \vec{F}_2 = \vec{e}^2 \begin{bmatrix} -m_v \beta^2 r_x \\ -m_v \beta^2 r_y \\ 0 \end{bmatrix}$$

en

$$\vec{M}_1 = \vec{e}^{2r} \begin{bmatrix} \dot{\alpha} \beta J_z + m_v \beta^2 r_y r_z - J_{yz} \beta^2 \\ -m_v \beta^2 r_z + J_{xz} \beta^2 \\ 0 \end{bmatrix}, \vec{F}_1 = \vec{e}^{2r} \begin{bmatrix} -m_v \beta^2 r_x \\ -m_v \beta^2 r_y - m_m \beta^2 r \\ 0 \end{bmatrix}$$

C.2 Kwantitatieve weergave van de bewegingsvergelijkingen

Invullen van de in appendix D genoemde gegevens

$$\vec{M}_2 = \vec{e}^{2r} \begin{bmatrix} 543.27 \\ 0 \\ 0 \end{bmatrix}, \vec{F}_2 = \vec{e}^{2r} \begin{bmatrix} 0.81225 \\ -65.3327 \\ 0 \end{bmatrix}$$

en

$$\vec{M}_1 = \vec{e}^{2r} \begin{bmatrix} 553.850 \\ -4.045E-3 \\ 0 \end{bmatrix}, \vec{F}_1 = \vec{e}^{2r} \begin{bmatrix} 0.81225 \\ -844.283 \\ 0 \end{bmatrix}$$

Appendix D : De invoerfile van het model voertuig op weg

```

ANALYSIS
CREATE SYSTEM.DATG
  UNITS                :- 'SI'
  ANALYSIS.TYPE        :- 'DYNAMIC'
  STARTING.TIME        :- '0.0'
  ENDING.TIME          :- '0.2'
  PRINT.INTERVAL       :- '0.001'
  GRAVITY.SEA.LEVEL    :- '9.80665'
  X.GRAVITY            :- '0.0'
  Y.GRAVITY            :- '0.0'
  Z.GRAVITY            :- '-1.0'
  SCALE.GRAVITY.COEF   :- '1.0'
  MATRIX.OPERATIONS    :- 'SPARSE'
  REDUNDANCY.CHECK     :- 'TRUE'
  LU.TOL               :- '1.0E-12'
  ASSEMBLY.TOL         :- '1.0E-3'
  BYPASS.ASSEMBLY     :- 'FALSE'
  OUTPUT.FILE          :- 'BINARY'
  REFERENCE.FRAME      :- 'LOCAL'
  DEBUG.FLAG           :- 'FALSE'

UP
CREATE DYNAMIC.DATG
  REACTION.FORCES      :- 'TRUE'
  FORCE.COORDINATES     :- 'BODY'
  PRINT.METHOD        :- 'INTERPOLATED'
  MAX.INT.STEP         :- '0.001'
  SOLUTION.TOL         :- '0.001'
  INTEGRATION.TOL      :- '0.0001'

UP
UP
FORCE
CREATE RSSD
  NAME                 :- 'ROLAS-VEER-DEMPER'
  JOINT.NAME           :- 'ROLAS'
  ORIENTATION.ANGLE    :- '0.0'
  SPRING.CONSTANT      :- '55000'
  DAMPING.COEFFICIENT  :- '3700'
  ACTUATOR.TORQUE      :- '0.0'
  CURVE.SPRING         :- 'NONE'
  CURVE.DAMPER         :- 'NONE'
  CURVE.ACTUATOR       :- 'NONE'
  ANGULAR.UNITS        :- 'RADIANS'

UP
CREATE TSDG
  NAME                 :- 'V1'
  BODY.1.NAME          :- 'FRAME'
  BODY.2.NAME          :- 'MOTOR'
  SPRING.CONSTANT      :- '160000'
  FREE.LENGTH.SPRING   :- '0.0'
  DAMPING.COEFFICIENT  :- '200'
  ACTUATOR.FORCE       :- '0.0'
  F.ON.BODY.1          :- ( 0.670, -0.319, 0.030 )
  F.ON.BODY.2          :- ( 0.245, -0.365, -0.141 )
  Q.ON.BODY.1          :- ( 0.670, -0.319, 1.030 )
  Q.ON.BODY.2          :- ( 0.245, -0.365, 0.859 )
  R.ON.BODY.1          :- ( 1.670, -0.319, 0.030 )
  R.ON.BODY.2          :- ( 1.245, -0.365, -0.141 )
  CURVE.SPRING         :- 'NONE'
  CURVE.DAMPER         :- 'NONE'
  CURVE.ACTUATOR       :- 'NONE'

```

```

NODE.1      := '0'
NODE.2      := '0'
UP
CREATE TSDA
NAME        := 'U2'
BODY.1.NAME := 'FRAME'
BODY.2.NAME := 'MOTOR'
SPRING.CONSTANT := '160000'
FREE.LENGTH.SPRING := '0.0'
DAMPING.COEFFICIENT := '200'
ACTUATOR.FORCE := '0.0'
F.ON.BODY.1 := ( 0.670, 0.338, 0.030 )
F.ON.BODY.2 := ( 0.245, 0.292, -0.141 )
Q.ON.BODY.1 := ( 0.670, 0.338, 0.883 )
Q.ON.BODY.2 := ( 0.245, 0.292, 0.883 )
R.ON.BODY.1 := ( 1.670, 0.338, 0.030 )
R.ON.BODY.2 := ( 1.245, 0.292, -0.141 )
CURVE.SPRING := 'NONE'
CURVE.DAMPER := 'NONE'
CURVE.ACTUATOR := 'NONE'
NODE.1      := '0'
NODE.2      := '0'
UP
CREATE TSDA
NAME        := 'U3'
BODY.1.NAME := 'FRAME'
BODY.2.NAME := 'MOTOR'
SPRING.CONSTANT := '160000'
FREE.LENGTH.SPRING := '0.0'
DAMPING.COEFFICIENT := '200'
ACTUATOR.FORCE := '0.0'
F.ON.BODY.1 := ( 0.000, 0.028, 0.055 )
F.ON.BODY.2 := ( -0.425, -0.018, -0.116 )
Q.ON.BODY.1 := ( 1.000, 0.028, 0.055 )
Q.ON.BODY.2 := ( 0.575, -0.018, -0.116 )
R.ON.BODY.1 := ( 0.000, 0.028, 1.055 )
R.ON.BODY.2 := ( -0.425, -0.018, 0.884 )
CURVE.SPRING := 'NONE'
CURVE.DAMPER := 'NONE'
CURVE.ACTUATOR := 'NONE'
NODE.1      := '0'
NODE.2      := '0'
UP
CREATE TSDA
NAME        := 'U1B'
BODY.1.NAME := 'FRAME'
BODY.2.NAME := 'MOTOR'
SPRING.CONSTANT := '150000'
FREE.LENGTH.SPRING := '0.0'
DAMPING.COEFFICIENT := '200'
ACTUATOR.FORCE := '0.0'
F.ON.BODY.1 := ( 0.670, -0.319, 0.030 )
F.ON.BODY.2 := ( 0.245, -0.365, -0.141 )
Q.ON.BODY.1 := ( 0.670, 0.681, 0.030 )
Q.ON.BODY.2 := ( 0.245, 0.635, -0.141 )
R.ON.BODY.1 := ( 1.670, -0.319, 0.030 )
R.ON.BODY.2 := ( 1.245, -0.365, -0.141 )
CURVE.SPRING := 'NONE'
CURVE.DAMPER := 'NONE'
CURVE.ACTUATOR := 'NONE'

```

```

NODE.1          :=" '0'
NODE.2          :=" '0'
UP
CREATE TSDA
NAME            :=" 'U1C'
BODY.1.NAME     :=" 'FRAME'
BODY.2.NAME     :=" 'MOTOR'
SPRING.CONSTANT :=" '130000'
FREE.LENGTH.SPRING :=" '0.0'
DAMPING.COEFFICIENT :=" '200'
ACTUATOR.FORCE  :=" '0.0'
F.ON.BODY.1     :=" ( 0.670, -0.319, 0.030 )
F.ON.BODY.2     :=" ( 0.245, -0.365, -0.141 )
Q.ON.BODY.1     :=" ( 1.670, -0.319, 0.030 )
Q.ON.BODY.2     :=" ( 1.245, -0.365, -0.141 )
R.ON.BODY.1     :=" ( 0.670, 0.681, 0.030 )
R.ON.BODY.2     :=" ( 0.245, 0.635, -0.141 )
CURVE.SPRING    :=" 'NONE'
CURVE.DAMPER    :=" 'NONE'
CURVE.ACTUATOR  :=" 'NONE'
NODE.1          :=" '0'
NODE.2          :=" '0'

```

```

UP
CREATE TSDA
NAME            :=" 'U2B'
BODY.1.NAME     :=" 'FRAME'
BODY.2.NAME     :=" 'MOTOR'
SPRING.CONSTANT :=" '130000'
FREE.LENGTH.SPRING :=" '0.0'
DAMPING.COEFFICIENT :=" '200'
ACTUATOR.FORCE  :=" '0.0'
F.ON.BODY.1     :=" ( 0.670, 0.338, 0.030 )
F.ON.BODY.2     :=" ( 0.245, 0.292, -0.141 )
Q.ON.BODY.1     :=" ( 0.670, 1.338, 0.030 )
Q.ON.BODY.2     :=" ( 0.245, 1.292, -0.141 )
R.ON.BODY.1     :=" ( 1.670, 0.338, 0.030 )
R.ON.BODY.2     :=" ( 1.245, 0.292, -0.141 )
CURVE.SPRING    :=" 'NONE'
CURVE.DAMPER    :=" 'NONE'
CURVE.ACTUATOR  :=" 'NONE'
NODE.1          :=" '0'
NODE.2          :=" '0'

```

```

UP
CREATE TSDA
NAME            :=" 'U2C'
BODY.1.NAME     :=" 'FRAME'
BODY.2.NAME     :=" 'MOTOR'
SPRING.CONSTANT :=" '130000'
FREE.LENGTH.SPRING :=" '0.0'
DAMPING.COEFFICIENT :=" '200'
ACTUATOR.FORCE  :=" '0.0'
F.ON.BODY.1     :=" ( 0.670, 0.338, 0.030 )
F.ON.BODY.2     :=" ( 0.245, 0.292, -0.141 )
Q.ON.BODY.1     :=" ( 1.670, 0.338, 0.030 )
Q.ON.BODY.2     :=" ( 1.245, 0.292, -0.141 )
R.ON.BODY.1     :=" ( 0.670, 1.338, 0.030 )
R.ON.BODY.2     :=" ( 0.245, 1.292, -0.141 )
CURVE.SPRING    :=" 'NONE'
CURVE.DAMPER    :=" 'NONE'
CURVE.ACTUATOR  :=" 'NONE'

```

```

NODE.1           := '0'
NODE.2           := '0'
UP
CREATE TSDA
NAME             := 'V3B'
BODY.1.NAME     := 'FRAME'
BODY.2.NAME     := 'MOTOR'
SPRING.CONSTANT := '300000'
FREE.LENGTH.SPRING := '0.0'
DAMPING.COEFFICIENT := '200'
ACTUATOR.FORCE  := '0.0'
F.ON.BODY.1     := ( 0.000, 0.028, 0.055 )
F.ON.BODY.2     := ( -0.425, -0.018, -0.116 )
Q.ON.BODY.1     := ( 0.000, 1.028, 0.055 )
Q.ON.BODY.2     := ( -0.425, 0.982, 0.884 )
R.ON.BODY.1     := ( 1.000, 0.028, 0.055 )
R.ON.BODY.2     := ( 0.575, -0.018, -0.116 )
CURVE.SPRING    := 'NONE'
CURVE.DAMPER    := 'NONE'
CURVE.ACTUATOR := 'NONE'
NODE.1         := '0'
NODE.2         := '0'
UP
CREATE TSDA
NAME             := 'V3C'
BODY.1.NAME     := 'FRAME'
BODY.2.NAME     := 'MOTOR'
SPRING.CONSTANT := '300000'
FREE.LENGTH.SPRING := '0.0'
DAMPING.COEFFICIENT := '200'
ACTUATOR.FORCE  := '0.0'
F.ON.BODY.1     := ( 0.000, 0.028, 0.055 )
F.ON.BODY.2     := ( -0.425, -0.018, -0.116 )
Q.ON.BODY.1     := ( 0.000, 0.028, 1.055 )
Q.ON.BODY.2     := ( -0.425, -0.018, 0.884 )
R.ON.BODY.1     := ( 1.000, 0.028, 0.055 )
R.ON.BODY.2     := ( 0.575, -0.018, -0.116 )
CURVE.SPRING    := 'NONE'
CURVE.DAMPER    := 'NONE'
CURVE.ACTUATOR := 'NONE'
NODE.1         := '0'
NODE.2         := '0'
UP
UP
JOINTS
CREATE REVOLUTE.JOINT
NAME             := 'REV1'
BODY.1.NAME     := 'AARDE'
BODY.2.NAME     := 'HULPLICHAAM'
F.ON.BODY.1     := ( 0.0, 0.0, 0.0 )
F.ON.BODY.2     := ( 0.425, -5.954, 0.581 )
Q.ON.BODY.1     := ( 0.0, 0.0, 1.0 )
Q.ON.BODY.2     := ( 0.425, -5.954, 1.581 )
R.ON.BODY.1     := ( 1.0, 0.0, 0.0 )
R.ON.BODY.2     := ( 1.425, -5.954, 0.581 )
NODE.1         := '0'
NODE.2         := '0'
UP
CREATE REVOLUTE.JOINT
NAME             := 'REV2'

```



```

BODY.1.NAME           := 'MOTOR'
BODY.2.NAME           := 'UV'
P.ON.BODY.1           := ( -0.097, 0.113, 0.171 )
P.ON.BODY.2           := ( 0.0, 0.0, 0.0 )
Q.ON.BODY.1           := ( -0.097, -0.887, 0.171 )
Q.ON.BODY.2           := ( 0.0, 0.0, 1.0 )
R.ON.BODY.1           := ( 0.903, 0.113, 0.171 )
R.ON.BODY.2           := ( 1.0, 0.0, 0.0 )
NODE.1                := '0'
NODE.2                := '0'

UP
CREATE REVOLUTE.JOINT
NAME                  := 'ROLAS'
BODY.1.NAME           := 'HULPLICHAAM'
BODY.2.NAME           := 'FRAME'
P.ON.BODY.1           := ( 0.0, 0.0, 0.0 )
P.ON.BODY.2           := ( 0.0, 0.0, -0.410 )
Q.ON.BODY.1           := ( 1.0, 0.0, 0.0 )
Q.ON.BODY.2           := ( 1.0, 0.0, -0.410 )
R.ON.BODY.1           := ( 0.0, 1.0, 0.0 )
R.ON.BODY.2           := ( 0.0, 1.0, -0.410 )
NODE.1                := '0'
NODE.2                := '0'

UP
UP
CREATE BODY
NAME                  := 'AARDE'
CENTER.OF.GRAVITY    := ( 0.0, 0.0, 0.0 )
TYPE.ANGULAR.COORD   := 'EULER'
ANGLE.1               := '0.0'
ANGLE.2               := '0.0'
ANGLE.3               := '0.0'
FIXED.TO.GROUND      := 'TRUE'
MASS                  := '1.0'
INERTIA.XXL           := '1.0'
INERTIA.YYL           := '1.0'
INERTIA.ZZL           := '1.0'
INERTIA.XYL           := '0.0'
INERTIA.XZL           := '0.0'
INERTIA.YZL           := '0.0'
XG.FORCE              := '0.0'
YG.FORCE              := '0.0'
ZG.FORCE              := '0.0'
XL.TORQUE             := '0.0'
YL.TORQUE             := '0.0'
ZL.TORQUE             := '0.0'
CURVE.XGF             := 'NONE'
CURVE.YGF             := 'NONE'
CURVE.ZGF             := 'NONE'
CURVE.XLT             := 'NONE'
CURVE.YLT             := 'NONE'
CURVE.ZLT             := 'NONE'
SIGN.EQ               := 'POSITIVE'
ANGULAR.UNITS         := 'RADIANS'
FLEXIBLE              := 'FALSE'
SUPERELEMENT          := 'FALSE'

UP
CREATE BODY
NAME                  := 'FRAME'
CENTER.OF.GRAVITY    := ( -0.425, 5.934, -0.171 )

```

```

TYPE, ANGULAR, COORD      := 'EULER'
ANGLE.1                   := '0.0'
ANGLE.2                   := '0.0'
ANGLE.3                   := '0.0'
FIXED, TO, GROUND        := 'FALSE'
MASS                      := '842'
INERTIA, XXL             := '380'
INERTIA, YYL            := '1.0'
INERTIA, ZZL            := '1.0'
INERTIA, XYL            := '0.0'
INERTIA, XZL            := '0.0'
INERTIA, YZL            := '0.0'
XG, FORCE                 := '0.0'
YG, FORCE                 := '0.0'
ZG, FORCE                 := '0.0'
XL, TORQUE               := '0.0'
YL, TORQUE               := '0.0'
ZL, TORQUE               := '0.0'
CURVE, XGF               := 'NONE'
CURVE, YGF               := 'NONE'
CURVE, ZGF               := 'NONE'
CURVE, XLT               := 'NONE'
CURVE, YLT               := 'NONE'
CURVE, ZLT               := 'NONE'
SIGN, EO                 := 'POSITIVE'
ANGULAR, UNITS           := 'RADIAN'S'
FLEXIBLE                 := 'FALSE'
SUPERELEMENT             := 'FALSE'

```

UP

CREATE BODY

```

NAME                      := 'MOTOR'
CENTER, OF, GRAVITY      := ( 0.0, 6.0, 0.0 )
TYPE, ANGULAR, COORD    := 'EULER'
ANGLE.1                   := '0.0'
ANGLE.2                   := '0.0'
ANGLE.3                   := '0.0'
FIXED, TO, GROUND        := 'FALSE'
MASS                      := '230.8'
INERTIA, XXL             := '18.93'
INERTIA, YYL            := '9.09'
INERTIA, ZZL            := '18.34'
INERTIA, XYL            := '2.08'
INERTIA, XZL            := '-0.35'
INERTIA, YZL            := '2.09'
XG, FORCE                 := '0.0'
YG, FORCE                 := '0.0'
ZG, FORCE                 := '0.0'
XL, TORQUE               := '0.0'
YL, TORQUE               := '0.0'
ZL, TORQUE               := '0.0'
CURVE, XGF               := 'NONE'
CURVE, YGF               := 'NONE'
CURVE, ZGF               := 'NONE'
CURVE, XLT               := 'NONE'
CURVE, YLT               := 'NONE'
CURVE, ZLT               := 'NONE'
SIGN, EO                 := 'POSITIVE'
ANGULAR, UNITS           := 'RADIAN'S'
FLEXIBLE                 := 'FALSE'
SUPERELEMENT             := 'FALSE'

```

UP

CREATE BODY

NAME	:= 'UV'
CENTER, OF, GRAVITY	:= (-0.097, 6.113, 0.171)
TYPE, ANGULAR, COORD	:= 'EULER'
ANGLE, 1	:= '0.0'
ANGLE, 2	:= '1.5707963'
ANGLE, 3	:= '0.0'
FIXED, TO, GROUND	:= 'FALSE'
MASS	:= '19.0'
INERTIA, XXL	:= '0.197'
INERTIA, YYL	:= '0.197'
INERTIA, ZZL	:= '0.364'
INERTIA, XYL	:= '0.0'
INERTIA, XZL	:= '0.0'
INERTIA, YZL	:= '0.0'
XG, FORCE	:= '0.0'
YG, FORCE	:= '0.0'
ZG, FORCE	:= '0.0'
XL, TORQUE	:= '0.0'
YL, TORQUE	:= '0.0'
ZL, TORQUE	:= '0.0'
CURVE, XGF	:= 'NONE'
CURVE, YGF	:= 'NONE'
CURVE, ZGF	:= 'NONE'
CURVE, XLT	:= 'NONE'
CURVE, YLT	:= 'NONE'
CURVE, ZLT	:= 'NONE'
SIGN, EO	:= 'POSITIVE'
ANGULAR, UNITS	:= 'RADIANB'
FLEXIBLE	:= 'FALSE'
SUPERELEMENT	:= 'FALSE'

UP

CREATE BODY

NAME	:= 'HULFLICHOAM'
CENTER, OF, GRAVITY	:= (-0.025, 5.954, -0.581)
TYPE, ANGULAR, COORD	:= 'EULER'
ANGLE, 1	:= '0.0'
ANGLE, 2	:= '0.0'
ANGLE, 3	:= '0.0'
FIXED, TO, GROUND	:= 'FALSE'
MASS	:= '1.0'
INERTIA, XXL	:= '1.0'
INERTIA, YYL	:= '1.0'
INERTIA, ZZL	:= '1.0'
INERTIA, XYL	:= '0.0'
INERTIA, XZL	:= '0.0'
INERTIA, YZL	:= '0.0'
XG, FORCE	:= '0.0'
YG, FORCE	:= '0.0'
ZG, FORCE	:= '0.0'
XL, TORQUE	:= '0.0'
YL, TORQUE	:= '0.0'
ZL, TORQUE	:= '0.0'
CURVE, XGF	:= 'NONE'
CURVE, YGF	:= 'NONE'
CURVE, ZGF	:= 'NONE'
CURVE, XLT	:= 'NONE'
CURVE, YLT	:= 'NONE'
CURVE, ZLT	:= 'NONE'

```

SIGN:EO                := 'POSITIVE'
ANGULAR.UNITS          := 'DEGREES'
FLEXIBLE               := 'FALSE'
SUPERELEMENT          := 'FALSE'

UP
CREATE INITIAL.CONDITION
NAME                   := 'IX'
BODY.1.NAME            := 'MOTOR'
BODY.2.NAME            := 'NONE'
ELEMENT.NAME           := 'NONE'
TYPE.INITIAL.COND     := 'X'
INITIAL.VALUE          := '0.0'
TIME.DERIVATIVE       := '-4.5'
OMEGA.Y                := '0.0'
OMEGA.Z                := '0.0'
P.ON.BODY.1            := ( 0.0, 0.0, 0.0 )
P.ON.BODY.2            := ( 0.0, 0.0, 0.0 )
EXTRA.COORD            := '0'
ANGULAR.UNITS          := 'RADIANS'

UP
CREATE INITIAL.CONDITION
NAME                   := 'IY'
BODY.1.NAME            := 'MOTOR'
BODY.2.NAME            := 'NONE'
ELEMENT.NAME           := 'NONE'
TYPE.INITIAL.COND     := 'Y'
INITIAL.VALUE          := '6.0'
TIME.DERIVATIVE       := '0.0'
OMEGA.Y                := '0.0'
OMEGA.Z                := '0.0'
P.ON.BODY.1            := ( 0.0, 0.0, 0.0 )
P.ON.BODY.2            := ( 0.0, 0.0, 0.0 )
EXTRA.COORD            := '0'
ANGULAR.UNITS          := 'RADIANS'

UP
CREATE INITIAL.CONDITION
NAME                   := 'IZ'
BODY.1.NAME            := 'MOTOR'
BODY.2.NAME            := 'NONE'
ELEMENT.NAME           := 'NONE'
TYPE.INITIAL.COND     := 'Z'
INITIAL.VALUE          := '0.0'
TIME.DERIVATIVE       := '0.0'
OMEGA.Y                := '0.0'
OMEGA.Z                := '0.0'
P.ON.BODY.1            := ( 0.0, 0.0, 0.0 )
P.ON.BODY.2            := ( 0.0, 0.0, 0.0 )
EXTRA.COORD            := '0'
ANGULAR.UNITS          := 'RADIANS'

UP
CREATE INITIAL.CONDITION
NAME                   := 'IUR'
BODY.1.NAME            := 'MOTOR'
BODY.2.NAME            := 'NONE'
ELEMENT.NAME           := 'NONE'
TYPE.INITIAL.COND     := 'ORIENTATION'
INITIAL.VALUE          := '0.0'
TIME.DERIVATIVE       := '0.0'
OMEGA.Y                := '0.0'
OMEGA.Z                := '0.75'

```

```

P.ON.BODY.1      := ( 0.0, 0.0, 0.0 )
P.ON.BODY.2      := ( 0.0, 0.0, 0.0 )
EXTRA.COORD      := '0'
ANGULAR.UNITS    := 'RADIANS'

UP
CREATE INITIAL.CONDITION
NAME             := 'INIFRAME'
BODY.1.NAME     := 'NONE'
BODY.2.NAME     := 'NONE'
ELEMENT.NAME    := 'ROLAS-VEER-DEMPER'
TYPE.INITIAL.COND := 'REL.ANGLE'
INITIAL.VALUE   := '0.0'
TIME.DERIVATIVE := '0.0'
OMEGA.Y        := '0.0'
OMEGA.Z        := '0.0'
P.ON.BODY.1     := ( 0.0, 0.0, 0.0 )
P.ON.BODY.2     := ( 0.0, 0.0, 0.0 )
EXTRA.COORD     := '0'
ANGULAR.UNITS   := 'RADIANS'

UP
CREATE DRIVER
NAME             := 'DRIVER1'
BODY.1.NAME     := 'NONE'
BODY.2.NAME     := 'NONE'
TYPE.DRIVER     := 'REL.ANGLE'
DRIVING.FUNCTION := 'POLYNOMIAL'
FUNCTION.PARAMETERS := ( 0.0, 0.75, 0.0, 0.0 )
P.ON.BODY.1     := ( 0.0, 0.0, 0.0 )
P.ON.BODY.2     := ( 0.0, 0.0, 0.0 )
Q.ON.BODY.1     := ( 0.0, 0.0, 1.0 )
Q.ON.BODY.2     := ( 0.0, 0.0, 1.0 )
R.ON.BODY.1     := ( 1.0, 0.0, 0.0 )
R.ON.BODY.2     := ( 1.0, 0.0, 0.0 )
CURVE.DRIVER    := 'NONE'
JOINT.NAME      := 'REV1'
ANGULAR.UNITS   := 'RADIANS'

UP
CREATE DRIVER
NAME             := 'DRIVER2'
BODY.1.NAME     := 'NONE'
BODY.2.NAME     := 'NONE'
TYPE.DRIVER     := 'REL.ANGLE'
DRIVING.FUNCTION := 'POLYNOMIAL'
FUNCTION.PARAMETERS := ( 0.0, 1990, 0.0, 0.0 )
P.ON.BODY.1     := ( 0.0, 0.0, 0.0 )
P.ON.BODY.2     := ( 0.0, 0.0, 0.0 )
Q.ON.BODY.1     := ( 0.0, 0.0, 1.0 )
Q.ON.BODY.2     := ( 0.0, 0.0, 1.0 )
R.ON.BODY.1     := ( 1.0, 0.0, 0.0 )
R.ON.BODY.2     := ( 1.0, 0.0, 0.0 )
CURVE.DRIVER    := 'NONE'
JOINT.NAME      := 'REV2'
ANGULAR.UNITS   := 'RADIANS'

UP

```