

Het DADS linearisatie element

Citation for published version (APA):

Siemerink, H. W. (1992). *Het DADS linearisatie element*. (DCT rapporten; Vol. 1992.089). 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.

HET DADS LINEARISATIE ELEMENT.

WFW-rapportnr: 92089

auteur: H.W. Siemerink
juli 1992.

SAMENVATTING.

In het kader van mijn stage aan de TU Eindhoven, Vakgroep WFW, heb ik onderzoek gedaan naar de werking van het linearisatie element in het multibody dynamica programma DADS.

In dit verslag wordt eerst de theoretische achtergrond van dit linearisatie element behandeld. Vervolgens wordt de werking van het DADS linearisatie element uitgewerkt. Aan de hand van een aantal voorbeelden wordt het gebruik van dit element toegelicht. Ten slotte wordt er ingegaan op het gebruik van dit element bij het toepassen van regelingen op modellen binnen DADS en de effecten van het gebruik van flexibele body's op de linearisaties.

INHOUDSOPGAVE.

| | |
|---|----|
| SAMENVATTING. | 1 |
| H 1: INLEIDING. | 3 |
| H 2: THEORETISCHE ACHTERGROND. | 4 |
| H 3: HET LINEARISATIE ELEMENT. | |
| 3.1: Algemeen. | 8 |
| 3.2: Benodigde data. | 8 |
| H 4: DE MODELVORMING. | 10 |
| H 5: VOORBEELDEN. | |
| 5.1: Model kwart auto-ophanging. | 12 |
| 5.2: Omgekeerde dubbele slinger. | |
| 5.2.1: Inleiding. | 13 |
| 5.2.2: Bepaling A en B matrices. | 14 |
| 5.2.3: Regeling met amplifiserende control elementen. | 17 |
| 5.2.4: Regeling met het general control element. | 18 |
| 5.3: Het toepassen van flexibele elementen. | 19 |
| H 6: VERIFICATIE VAN DE REGELRESULTATEN. | 22 |
| BESPREKING VAN DE RESULTATEN. | 26 |
| LITERATUURLIJST. | 27 |
| BIJLAGEN. | 28 |

H 1: INLEIDING.

Het software pakket DADS bestaat uit een aantal programma's die gebruikt kunnen worden om een mechanisch systeem te modelleren en waarmee vervolgens het dynamische gedrag van dit systeem te simuleren is. Het programma genereert een mathematisch model van het echte systeem en berekent posities, snelheden, versnellingen, krachten en momenten die op de verschillende onderdelen van het systeem werken.

De bewegingen van het mechanische systeem worden over het algemeen beschreven door niet-lineaire algebraïsche en differentiaalvergelijkingen. De meeste regelaars zijn daarentegen ontworpen op basis van lineaire modellen. Voor het analyseren van de stabiliteit van een systeem wordt eveneens vaak een gelineariseerd model gebruikt. In de nieuwste versie op de TUE van DADS (versie 6.5) is een linearisatie algoritme geïmplementeerd waarmee op bepaalde tijdstippen in een simulatie een linearisatie van het model kan worden bepaald.

In dit verslag wordt de werking en het gebruik van deze linearisatie-module uitgewerkt en toegelicht aan de hand van een aantal voorbeelden.

H 2: THEORETISCHE ACHTERGROND.

Het DADS linearisatie algoritme maakt gebruik van een lokale linearisatie theorie, de perturbatie theorie, om een lineair model te vormen. Het lineaire model is bruikbaar in de directe omgeving van het werkpunt. Dit werkpunt is het punt waar rond gelineariseerd wordt. Een kwadratische benadering wordt gebruikt om alle partiële afgeleiden te beschrijven.

Een mechanisch systeem kan beschreven worden door een set gegeneraliseerde coördinaten. Deze coördinaten moeten in DADS onafhankelijk zijn omdat anders slecht geconditioneerde matrices en singulariteiten kunnen ontstaan. Met deze set coördinaten kan een stelsel niet-lineaire algebraïsche en tweede orde differentiaalvergelijkingen van een systeem worden opgesteld. Deze niet-lineaire algebraïsche en differentiaalvergelijkingen kunnen met behulp van Lagrange multipliers omgeschreven worden tot een stelsel niet-lineaire tweede orde differentiaalvergelijkingen van de vorm:

$$M(\underline{q}) \ddot{\underline{q}} + \Phi_{\underline{q}}^T(\underline{q}, t) \lambda = Q(\dot{\underline{q}}, \underline{q}, t) \quad (1)$$

Waarbij :

| | |
|--|--|
| \underline{q} | = kolom met gegeneraliseerde coördinaten |
| $M(\underline{q})$ | = massamatrix |
| $\Phi_{\underline{q}}(\underline{q}, t)$ | = constraint Jacobiaan matrix |
| λ | = Lagrange multipliers |
| $Q(\dot{\underline{q}}, \underline{q}, t)$ | = kolom met gegeneraliseerde krachten |

Dit stelsel vergelijkingen kan worden omgeschreven naar een stelsel niet-lineaire eerste orde differentiaalvergelijkingen door een toestandsvector te introduceren. Deze toestandsvector bevat de gegeneraliseerde coördinaten en hun tijdsafgeleiden. Het systeem is dan te schrijven als:

$$\dot{\underline{X}} = f(\underline{X}, U) \quad (2)$$

De toestandsvector wordt door \underline{X} (grootte= n , is twee keer de grootte van g) weergegeven terwijl \underline{U} (grootte m) een set ingangen bevat die op het systeem werken. Het niet-lineaire stelsel vergelijkingen kan gelineariseerd worden door formule (2) met een eerste orde Taylor reeks te benaderen. Voor elk element van de toestandsvector geldt dan:

$$\dot{X}_i = f_i(\underline{X}_0, \underline{U}_0) + \sum_{j=1}^n \left. \frac{\partial f_i}{\partial X_j} \right|_{\underline{X}_0, \underline{U}_0} [X_j - X_{j0}] + \sum_{k=1}^m \left. \frac{\partial f_i}{\partial U_k} \right|_{\underline{X}_0, \underline{U}_0} [U_k - U_{k0}] \quad (3)$$

Dit geldt voor $i=1..n$; verder is $f(\underline{X}_0, \underline{U}_0)$ de functiewaarde in het werkpunt van het systeem waar rond gelineariseerd wordt. Als we de volgende nieuwe coördinaten definiëren : $\delta X_j = X_j - X_{j0}$ en $\delta U_k = U_k - U_{k0}$ en vervolgens (3) herschrijven krijgen we:

$$\delta \dot{X}_i = \sum_{j=1}^n \left. \frac{\partial f_i}{\partial X_j} \right|_{\underline{X}_0, \underline{U}_0} \delta X_j + \sum_{k=1}^m \left. \frac{\partial f_i}{\partial U_k} \right|_{\underline{X}_0, \underline{U}_0} \delta U_k \quad i=1..n \quad (4)$$

of:

$$\delta \dot{\underline{X}} = A \delta \underline{X} + B \delta \underline{U} \quad (5)$$

met:

$$A_{ij} = \left. \frac{\partial f_i}{\partial X_j} \right|_{\underline{X}_0, \underline{U}_0} \quad \text{en} \quad B_{ik} = \left. \frac{\partial f_i}{\partial U_k} \right|_{\underline{X}_0, \underline{U}_0} \quad i=1..n \quad (6)$$

A wordt de systeem-matrix genoemd en B wordt de ingangsmatrix genoemd. In de toestandsvector \underline{X} zitten behalve de set onafhankelijke gegeneraliseerde coördinaten met bijbehorende afgeleiden ook nog de toestanden van eventueel gebruikte "control-elementen", de zogenaamde control states ([Lit.1] en bijlagen 3,4,5 en 6). In dat geval wordt de toestandsvector groter dan n .

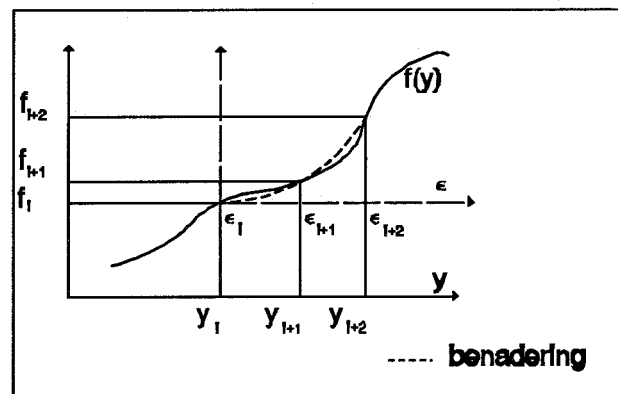


Fig.1

De partiële afgeleiden uit (6) worden benaderd door een kwadratische benaderingstechniek. In Fig.1 is te zien hoe de benadering wordt bepaald. De functie $f(y)$ wordt in het interval $y_i \leq y \leq y_{i+2}$ benaderd door de gestippelde lijn. $f(y)$ kan kwadratisch benaderd worden in de lokale ϵ -coördinaat volgens:

$$f(\epsilon) = [\epsilon^2 \ \epsilon \ 1] \begin{bmatrix} c \\ d \\ e \end{bmatrix} \quad (7)$$

De partiële afgeleide van f naar y is nu als volgt met de kettingregel te berekenen:

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial \epsilon} \frac{\partial \epsilon}{\partial y} \quad (8)$$

De functiewaarden uit Fig.1 zijn met (7) in de lokale coördinaat ϵ als volgt op te schrijven:

$$\begin{bmatrix} f_i \\ f_{i+1} \\ f_{i+2} \end{bmatrix} = \begin{bmatrix} \epsilon_i^2 & \epsilon_i & 1 \\ \epsilon_{i+1}^2 & \epsilon_{i+1} & 1 \\ \epsilon_{i+2}^2 & \epsilon_{i+2} & 1 \end{bmatrix} \begin{bmatrix} c \\ d \\ e \end{bmatrix} \quad (9)$$

Als we $[c \ d \ e]^T$ uit (9) oplossen en in (7) invullen ontstaat:

$$f(\epsilon) = [\epsilon^2 \ \epsilon \ 1] \begin{bmatrix} \epsilon_i^2 & \epsilon_i & 1 \\ \epsilon_{i+1}^2 & \epsilon_{i+1} & 1 \\ \epsilon_{i+2}^2 & \epsilon_{i+2} & 1 \end{bmatrix}^{-1} \begin{bmatrix} f_i \\ f_{i+1} \\ f_{i+2} \end{bmatrix} \quad (10)$$

Wordt de lokale coördinaat zo gekozen dat $\epsilon_i=0$, $\epsilon_{i+1}=0.5$ en $\epsilon_{i+2}=1$ dan geldt:

$$f(\epsilon) = [\epsilon^2 \ \epsilon \ 1] \begin{bmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1/2) \\ f(1) \end{bmatrix} \quad (11)$$

waarin f in de lokale coördinaat wordt uitgedrukt. Vervolgens is van (11) de partiële afgeleide naar de lokale coördinaat ϵ te berekenen:

$$\frac{\partial f(\epsilon)}{\partial \epsilon} = [2\epsilon \ 1 \ 0] \begin{bmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1/2) \\ f(1) \end{bmatrix} \quad (12)$$

Door uitschrijven van deze vergelijking volgt de partiële afgeleide van f naar de lokale coördinaat ϵ :

$$z = \frac{\partial f(\epsilon)}{\partial \epsilon} = -3f(0) + 4f(1/2) - f(1) + \epsilon(4f(0) - 8f(1/2) + 4f(1)) \quad (13)$$

De lokale coördinaat kan in de globale y coördinaat worden uitgedrukt door:

$$\epsilon = \frac{Y - Y_i}{Y_{i+2} - Y_i} \quad (14)$$

De afgeleide van de lokale naar de globale coördinaat is :

$$\frac{\partial \epsilon}{\partial Y} = \frac{1}{Y_{i+2} - Y_i} \quad (15)$$

De partiële afgeleide van f naar y is vervolgens te bepalen door (13) en (15) in (8) in te vullen. We krijgen dan:

$$\frac{\partial f}{\partial Y} = \frac{z}{Y_{i+2} - Y_i} \quad (16)$$

waarmee we een lineaire benadering voor de partiële afgeleiden hebben verkregen. Hiermee zijn dus de systeemmatrix A en ingangsmatrix B te bepalen.

H 3: HET LINEARISATIE-ELEMENT.

3.1 Algemeen.

De in DADS geïmplementeerde linearisatiemodule kan op of op de eerste tijdstap na elk vastgelegd printinterval de gelineariseerde vorm van de bewegingsvergelijkingen bepalen. Dit gebeurt als in de preprocessor bij de modelvorming het linearisatie-element 'LINEAR' geïntroduceerd wordt. Deze gelineariseerde vorm kan in de postprocessor bewerkt worden zodat de informatie in andere software programma's in te lezen is waar het eventueel bewerkt en/of geanalyseerd kan worden.

De bewegingsvergelijkingen die tijdens de dynamische analyse opgesteld worden zijn niet-lineaire algebraïsche en differentiaalvergelijkingen. Bij de linearisatie wordt een stelsel lineaire eerste orde differentiaalvergelijkingen verkregen (zie(5)). Als het linearisatie element gebruikt wordt dan berekent DADS de matrix A. Als de matrix B bestaat wordt ook deze matrix bepaald. Deze matrices worden in de binaire outputfile gezet. Deze gelineariseerde vorm is alleen lokaal bruikbaar.

3.2 Benodigde data.

De gebruiker moet in het linearisatie-element LINEAR een aantal parameters specificeren. Deze parameters zijn de initiële perturbatiefactor, de minimale perturbatiefactor en de convergentiefactor. Een gedetailleerde beschrijving van deze factoren is in Bijlage.1 te vinden.

De gebruiker kan in het linearisatie-element zelf de gewenste gegeneraliseerde coördinaten vastleggen door middel van Number.states en File.name die dan nader gespecificeerd moeten worden (Bijlage.1). Het vastleggen van deze coördinaten doet men door Number.states de waarde van het aantal gegeneraliseerde coördinaten ('control-states' niet meenemen) te geven en File.name de naam te geven van de externe ASCII text file waarin de gegeneraliseerde coördinaten staan. Deze externe file wordt in een tekst editor buiten DADS aangemaakt. DADS

haalt uit deze file de elementen van de toestandsvector. De toestandsvector bestaat uit de coördinaten uit deze file plus de bijbehorende afgeleiden, die niet in de externe file beschreven staan. De toestanden van de in het model gebruikte regelonderdelen, de 'control-states', worden door DADS ook in de toestandsvector van het systeem opgenomen. Deze 'control-states' worden automatisch aan de toestandsvector toegevoegd. De toestandsvector van het systeem is dus minimaal twee keer zo groot als het aantal gegeneraliseerde coördinaten uit de externe input file.

Als de opgegeven coördinaten afhankelijk zijn of als het aantal coördinaten niet gelijk is aan het aantal coördinaten dat DADS zelf vindt, wordt er een foutmelding in de informatie file gegeven. Vervolgens genereert DADS automatisch een set onafhankelijke coördinaten. De uiteindelijk voor de linearisatie gebruikte coördinaten worden in de informatie file geschreven. De volgorde die in de informatie file aangehouden wordt is ook de volgorde in de toestandsvector. De 'control-states' worden apart boven in de information file gezet. De namen van de knooppunten van de control-elementen ([Lit.1] of Bijlagen 3,4,5 en 6) worden door DADS genummerd met een knooppuntnummer.

Het soort coördinaat dat gedefinieerd kan worden staat in Bijlage.2 beschreven. In de externe ASCII text file wordt elke coördinaat op één regel beschreven. Eerst wordt de naam van het soort element gegeven (bijvoorbeeld body1, rsda2) en vervolgens wordt het type coördinaat ingetypt (bijvoorbeeld x, rel.angle). Tussen deze gegevens hoort een spatie te staan. Voor de volgende coördinaat moet men een nieuwe regel gebruiken.

Men moet wel beseffen dat de toestand een perturbatie toestand is en dat deze dus niet met de werkelijke coördinaten beschreven wordt.

H 4: DE MODELVORMING.

Met het linearisatie element is dus van een systeem de systeemmatrix A en eventueel de ingangsmatrix B te bepalen. Op deze matrices kunnen in andere programma's bewerkingen worden uitgevoerd waarna er op het systeem bijvoorbeeld een regeling kan worden toegepast, of waarna eventueel verbeteringen kunnen worden aangebracht. In dit hoofdstuk wordt besproken hoe te werk te gaan bij het opstellen van een model van een systeem waarvan het gelineariseerde model in de vorm van de matrices A en B bepaald moet worden.

Er wordt begonnen met het modelleren van het mechanische systeem in de preprocessor. Dit gebeurt met body's, scharnieren, veren, dempers etc. Dit model wordt het mechanische model genoemd. Als de gelineariseerde vorm in de uitvoer gewenst is moet het linearisatie-element 'LINEAR' worden toegepast, zoals in hoofdstuk 3 beschreven is. Het is eventueel mogelijk dat er op het systeem een kracht of koppel werkt waardoor het systeem enigszins stuurbaar wordt. Dan bestaat er ook een ingangsmatrix B. Deze kracht of dit koppel wordt door middel van een output control element op het mechanische model uitgeoefend. Een output control element oefent een kracht/koppel uit op een body. Deze kracht of dit koppel heeft de waarde van een knooppuntsvariabele. De waarde van deze knooppuntsvariabele wordt bepaald door een input control element. Het output control element en het input control element worden verder beschreven in respectievelijk Bijlage 3 en Bijlage 4. Om nu de waarde van de ingangsmatrix te bepalen moet in het gebruikte input control element de waarde van 'Linear.flag' op TRUE worden gezet, anders wordt deze matrix niet bepaald (dit staat niet in de handleiding van DADS).

Van bovenstaand model moet nu een dynamische analyse gemaakt worden. Er moet bovendien, in tegenstelling tot wat er in de handleiding staat, minimaal één tijdstap gedaan worden om een linearisatie te krijgen (ending-time in het system data element moet ongelijk aan nul zijn). Dit is nodig omdat in de postprocessor bij het wegschrijven van de systeem- en ingangsmatrices een tijdstap op moet worden gegeven die tussen 0 en het totaal aantal tijdstappen moet liggen. Aangezien er bij ending.time=0 nul tijdstappen gedaan zijn is er nog geen linearisatie mogelijk.

Op de eerste tijdstap na elk printinterval of op het printinterval zelf wordt nu een

linearisatie uitgevoerd. Er wordt gelineariseerd op het printinterval als in het element dynamic data print.method op interpolated staat. Als print.method daarentegen op actual staat wordt gelineariseerd op de eerst volgende tijdstap na het printinterval.

Is het bovenstaande model in z'n geheel ingevoerd dan kan men de preprocessor verlaten en een simulatie uitvoeren met de commando's DADS2D/DADS3D. De resultaten kunnen bekeken worden in de postprocessor. In deze postprocessor is het mogelijk de matrices A en B weg te schrijven naar diverse files. Deze files zijn door andere programmatuur in te lezen of zijn in een tekst editor te bekijken.

De files kunnen van de volgende soort zijn:

- * Een text file, te lezen in een gewone tekst editor.
- * Een input file voor Easy5.
- * Een input file voor Matrixx.
- * Een binaire inputfile voor Matlab.
- * Een inputfile voor het DADS general control element.

Het commando voor het wegschrijven van A,B naar een externe file is:

Write linear [type] [timestep] [filename].

Hierin geeft [type] het type file aan, dit kan dus zijn : text,easy5,matrixx,matlab of dads. [timestep] moet een integer zijn en geeft het printinterval aan waarop of waarna gelineariseerd wordt. Tenslotte is [filename] de naam van de file waar de systeemmatrix A naar toe wordt geschreven. Als er een ingangsmatrix B bestaat wordt deze bij een textfile automatisch in bovenstaande file weggeschreven, in de andere gevallen wordt er naar een andere filename gevraagd waar de B matrix naar toe moet worden geschreven.

Het is nu mogelijk om deze matrices in de diverse programma's in te lezen en te bewerken. Bijvoorbeeld inlezen in matlab gaat met load 'filename'. In het volgende hoofdstuk wordt een voorbeeld behandeld waarin een regeling is verwerkt die gebaseerd is op de optimale regelwet. Met de functie lqr is in matlab de optimale terugkoppeling te bepalen. Met de resultaten uit de berekeningen in matlab is vervolgens in DADS een regeling op het systeem toe te passen. Hiervoor wordt het oude model gebruikt plus enige aanvullingen. In het volgende hoofdstuk zullen enige verhelderende voorbeelden behandeld worden.

H 5: VOORBEELDEN.

Om te beginnen wordt een eenvoudig lineair systeem behandeld dat op analytische wijze eenvoudig te controleren is. Vervolgens wordt er een ingewikkelder niet-lineair systeem behandeld waarvan het gelineariseerde model bepaald wordt. Met behulp van dit gelineariseerde model wordt een regeling ontworpen. Deze regeling wordt in DADS op twee verschillende manieren toegepast. Verder is er gekeken of bij toepassing van flexibele elementen linearisaties mogelijk bleven. Ten slotte is het nuttig om een vergelijking te maken tussen een regeling toegepast op een model in DADS en hetzelfde geregelde model in Matlab. Dit laatste wordt in het volgende hoofdstuk beschreven.

5.1: Model kwart auto-ophanging.

Het eerste voorbeeld stelt een vereenvoudigd lineair model van een auto-ophanging voor. Er wordt een kwart van de auto gemodelleerd. Het model bestaat uit 2 massa's, 2 veren en een demper. Dit model is in Fig.2 getekend. De gebruikte waarden voor de massa's, stijfheden en dempingsconstanten staan in Fig.2. x_g is de ingang van het systeem, massa M1 stelt de massa

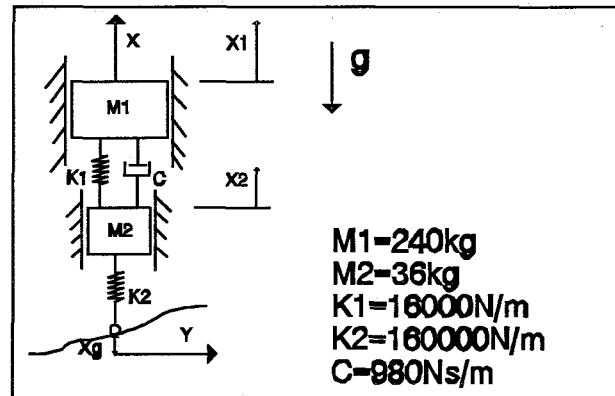


Fig.2

van de auto voor en massa M2 stelt de massa van het wiel voor. De bewegingsvergelijkingen voor dit systeem zijn als $x_1=x_2=0$ de rust toestand is:

$$M_1 \ddot{x}_1 = K_1 (x_2 - x_1) + C (\dot{x}_2 - \dot{x}_1) \quad (17)$$

en

$$M_2 \ddot{x}_2 = K_2 (x_g - x_2) + K_1 (x_1 - x_2) + C (\dot{x}_1 - \dot{x}_2) \quad (18)$$

Uitschrijven in toestandsvorm:

$$\dot{\underline{X}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{K_1}{M_1} & \frac{K_1}{M_1} & -\frac{C}{M_1} & \frac{C}{M_1} \\ \frac{K_1}{M_2} & -\frac{K_2+K_1}{M_2} & \frac{C}{M_2} & -\frac{C}{M_2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K_2}{M_2} \end{bmatrix} x_g \quad (19)$$

De door DADS berekende systeem matrix en ingangsmatrix zijn gelijk aan de analytisch afgeleide matrices namelijk:

$$\dot{\underline{X}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -66.7 & 66.7 & -4.1 & 4.1 \\ 444.4 & -4888.9 & 27.22 & -27.22 \end{bmatrix} \underline{X} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 4444.4 \end{bmatrix} x_g \quad (20)$$

Het bijbehorende programma is in Bijlage 7 gegeven. De handelingen die nodig zijn om deze linearisatie te verkrijgen staan in hoofdstuk 4 en worden in de volgende paragraaf voor een veel uitgebreider model nog eens stap voor stap behandeld.

5.2: Omgekeerde dubbele slinger.

5.2.1: Inleiding.

Het tweede voorbeeld is een niet-lineair systeem waarvoor het opstellen van de bewegingsvergelijkingen tijdrovend maar niet onmogelijk is. Het gaat hier om een omgekeerde dubbele slinger. Op dit systeem passen we een regeling toe om de slingers in een bepaalde positie te manoeuvreren. De meeste ontwerpmethoden voor regelingen vereisen een lineair of gelineariseerd

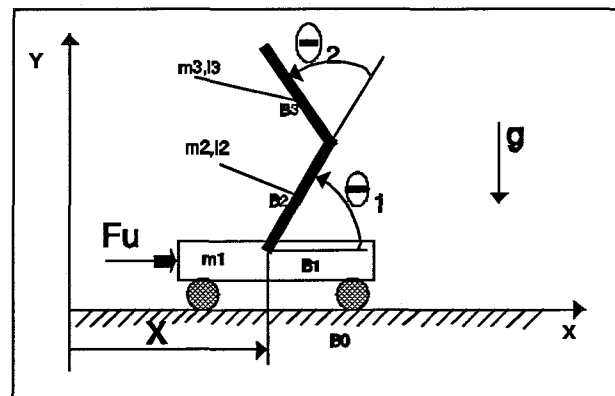


Fig. 3

model. Het model wordt daarom gelineariseerd. De regeling wordt vervolgens met behulp van de optimale regeltheorie ontworpen en toegepast.

Een tekening van de dubbele slinger is in Fig.3 te zien. Hij bestaat uit een tweetal gekoppelde slingers (slingers B2 en B3). Het uiteinde van één slinger (slinger B2) is aan een karretje bevestigd. Dit karretje kan in horizontale richting heen en weer bewegen en wordt bestuurd door een kracht F_u . F_u is dus de ingang van het systeem. Er zijn een drietal gegeneraliseerde coördinaten te definiëren. Deze coördinaten zijn in dit voorbeeld: x , Θ_1 en Θ_2 . Verder zijn de massa's en lengtes van de slingers gedefinieerd. Voor de massa's en lengtes is gekozen: $m_1=0.2\text{kg}$, $m_2=0.01\text{kg}$, $l_2=0.5\text{m}$, $m_3=0.01\text{kg}$, $l_3=0.7\text{m}$.

In dit voorbeeld wordt een regeling toegepast om de slingers in een verticale recht-opstaande stand te houden. Met behulp van de optimale regelwet [Lit.2] wordt de kracht F_u bepaald. Hierbij wordt gebruik gemaakt van de gelineariseerde toestandsvorm in het werkpunt (in dit voorbeeld: $x=0$, $\Theta_1=90^\circ$ en $\Theta_2=0^\circ$). Bij de optimale regelwet wordt de volgende functie geminimaliseerd:

$$J = \int_0^{\infty} (\delta \underline{X}^T Q \delta \underline{X} + \delta \underline{U}^T R \delta \underline{U}) dt \quad (21)$$

\underline{X} stelt de toestand van het systeem voor, \underline{U} de ingang en δ geeft aan dat het om de gelineariseerde toestandsvector en de gelineariseerde ingang gaat, die dus alleen rond het werkpunt gelden. De vector $\delta \underline{X}$ is dus de toestand van het systeem minus de werkpuntwaarden. Voor $\delta \underline{U}$ geldt hetzelfde. De matrices Q en R zijn weegmatrices die respectievelijk de toestand en de ingang van het systeem wegen. Minimaliseren van (21) levert de optimale terugkoppolversterkingsmatrix K op die de ingang $\delta \underline{U}$ als volgt bepaald:

$$\delta \underline{U} = -K \delta \underline{X} \quad (22)$$

Deze matrix K kan bijvoorbeeld berekend worden in het programma matlab. In dit voorbeeld is de kracht F_u de ingang van het systeem (er geldt dus $F_u = \delta \underline{U}$).

Het toepassen van de regeling op het model in DADS kan op twee manieren gebeuren namelijk met behulp van amplifier control elementen of met behulp van een general control element. In de volgende paragrafen wordt eerst de systeemmatrix A, de ingangsmatrix B en de terugkoppolversterkingsmatrix K bepaald. Vervolgens wordt besproken hoe de regeling toe is te passen.

5.2.2: Bepalen A en B matrices.

In eerste instantie wordt er weer begonnen met het opstellen van het mechanische model in de preprocessor. Dit gebeurt door middel van vier body's, twee rotational joints en één translational joint. Tevens is er een linearisatie element nodig om de gelineariseerde vorm te verkrijgen. Om de matrix B te verkrijgen moet met behulp van een input control element, verbonden aan een output control element, de kracht F_u op het karretje worden uitgeoefend. In het input control element moet daarvoor LINEAR.FLAG op 'TRUE' worden gezet. De knooppuntswaarde van het input control element mag bij een linearisatie niet gelijk worden gemaakt aan een gegeneraliseerde coördinaat van het mechanische systeem. DADS zal dan een foutmelding geven en de simulatie afbreken.

DADS kiest automatisch zelf voor de linearisatie als gegeneraliseerde coördinaten:

- * Van body B2 (de onderste slinger) de eerste eulerparameter E0
- * Van body B3 (de bovenste slinger) de x-coördinaat
- * Van body B3 de eerste eulerparameter E0

Als de in Fig.3 getekende gegeneraliseerde coördinaten x, θ_1 en θ_2 in de linearisatie gebruikt moeten worden dan moeten deze coördinaten worden vastgelegd in de externe file die in het linearisatie-element wordt gedefinieerd. De soorten coördinaten die in deze file gedefinieerd kunnen worden staan in Bijlage.2 beschreven. Er blijkt dat θ_1 en θ_2 niet direct als coördinaten te definiëren zijn en dat daarvoor extra rsda-elementen moeten worden gebruikt. De twee hoeken zijn makkelijk te definiëren met een rsda element in elke revolute joint waarbij alle parameters in het rsda element op nul worden gezet. Bij het gebruik van de hoeken van de rsda elementen als coördinaten blijkt dat het systeem in DADS in 3D gemodelleerd moet worden (Bijlage.2), de rsda elementen hebben verder geen nadelige effecten op het systeem. De gegeneraliseerde coördinaten kunnen nu in de externe input-file voor het linearisatie-element worden vastgelegd. Dit gebeurt voor x door eerst de naam van de body van het wagentje neer te schrijven en vervolgens de naam van

de coördinaat van de body. Voor Θ_1 en Θ_2 moeten op de 2 volgende regels eerst als body-naam de rsda-naam in worden gevuld en vervolgens moet rel.angle gebruikt worden als coördinaat naam.

Door nu uit de preprocessor te gaan en een simulatie uit te voeren zijn de systeem- en ingangsmatrix te bepalen. Dit kan gebeuren door in de postprocessor met het commando write linear, als in hoofdstuk 4 beschreven staat, de matrices weg te schrijven in files die leesbaar zijn voor Matlab. In Matlab zijn deze matrices in te lezen. De gebruikte toestandsvector is:

$$\underline{X} = [x \quad \theta_1 \quad \theta_2 \quad \dot{x} \quad \dot{\theta}_1 \quad \dot{\theta}_2]^T \quad (23)$$

De berekende matrices voor linearisaties rond $\underline{X}_0 = [0 \quad \pi/2 \quad 0 \quad 0 \quad 0 \quad 0]^T$ en $Fu_0 = 0$ zijn:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0.8287 & -0.1036 & 0 & 0 & 0 \\ 0 & 27.35 & -25.50 & 0 & 0 & 0 \\ 0 & -33.86 & 73.61 & 0 & 0 & 0 \end{bmatrix} \quad \text{en} \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 4.92 \\ 12.68 \\ -15.70 \end{bmatrix} \quad (24)$$

In Matlab is met deze matrices en de functie $[K,S] = \text{lqr}(A,B,Q,R)$ de terugkoppelmatrix K te berekenen. Q en R zijn weegmatrices. Voor toestandsweging Q is een 6×6 eenheidsmatrix genomen en voor ingangsweging R is 1 genomen. De matrix S is de steady-state oplossing uit de Ricatti vergelijking ([Lit.2] formule 4.93). De terugkoppelmatrix die verkregen wordt is:

$$K = [1 \quad -13.325 \quad -47.775 \quad 2.039 \quad -6.9997 \quad -7.5636] \quad (25)$$

Dit resultaat moet dus als regeling ($Fu = -K\delta\underline{X}$ waarbij $\delta\underline{X}$ de perturbatietoestand is) op het model van de dubbele slinger worden uitgevoerd. De regeling is op twee manieren toe te passen. In de volgende paragrafen worden deze toepassingen beschreven.

5.2.3: Regeling met amplifler control elementen.

Bij het toepassen van de regeling op het model kan het in de voorgaande paragraaf gebruikte model worden uitgebreid. Eerst moeten het linearisatie-element en het gebruikte input control element worden verwijderd. Vervolgens kan met behulp van input control elementen de toestandsvector van het mechanische model bepaald worden. De waarden van x , \dot{x} , $\dot{\Theta}_1$ en $\dot{\Theta}_2$ kunnen direct of via een sommatie worden bepaald met input control elementen. De waarden van Θ_1 en Θ_2 kunnen vervolgens door integreren van $\dot{\Theta}_1$ en $\dot{\Theta}_2$ worden bepaald. Dit gebeurt met behulp van integrator control elementen. Via amplifler control elementen waarin de waarden van de in Matlab berekende terugkoppelmatrix zijn verwerkt en een aantal summer control elementen kan de kracht F_u bepaald worden. Deze kracht wordt ten slotte door middel van een output control element op het karretje uitgeoefend.

Een schema van deze regeling is in Bijlage.8 getekend. De verbose file van het model is in Bijlage.9 te zien. Verder staat de beschrijving van de gebruikte control elementen voor het:

- * Output control element in Bijlage.3
- * Input control element in Bijlage.4
- * Amplifler control element in Bijlage.5
- * Summer control element in Bijlage.5
- * Integrator control element in Bijlage.5.

Ten slotte zijn de resultaten gegeven van een simulatie waarbij de begincondities op $t=0$ zijn: $x=0$, $\Theta_1=80^\circ$, $\Theta_2=0^\circ$, $\dot{x}=0$, $\dot{\Theta}_1=0$ en $\dot{\Theta}_2=0$. De resultaten van deze

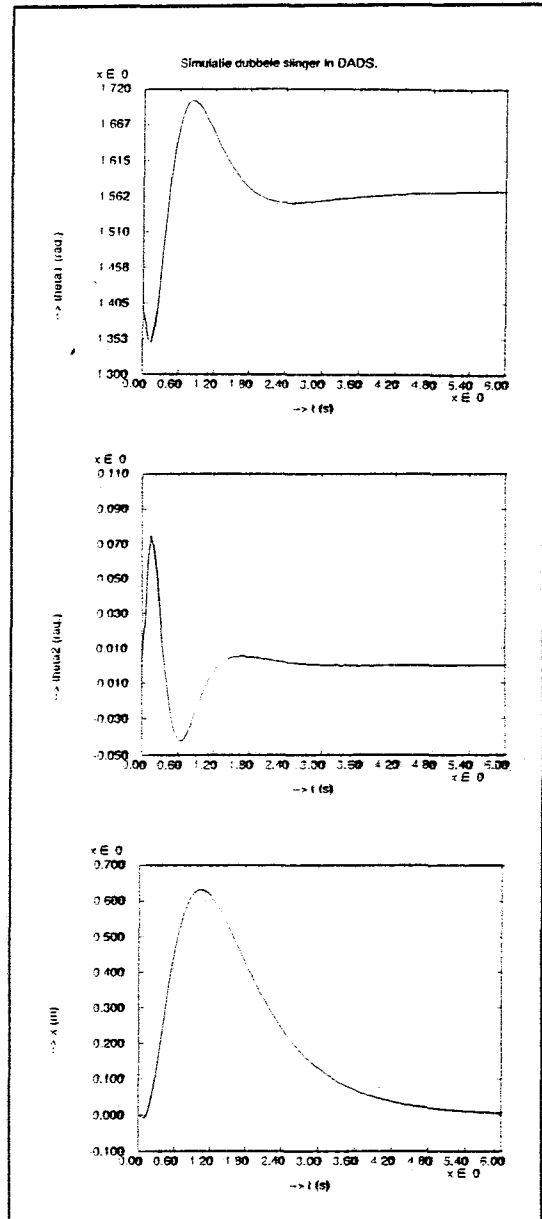


Fig.4

simulatie staan in Fig.4. Er blijkt dat de slinger netjes gebalanceerd wordt, want Θ_1 nadert tot $\pi/2$ radialen Θ_2 nadert tot 0 radialen en x nadert tot 0 meter.

5.2.4: Regeling met het general control element.

Een sneller toe te passen en vooral sneller aan te passen regeling is de regeling die gebruik maakt van een general control element. Vooral bij modelleren met een grote toestandsvector is deze toepassing aan te bevelen. Een uitgebreide beschrijving van het general control element is in Bijlage.6 te zien.

Het general control element modelleert een systeem in de toestandsvorm. Dit systeem ziet er als volgt uit:

$$\begin{aligned}\dot{\underline{X}}_g(t) &= \underline{A}_g \underline{X}_g(t) + \underline{B}_g \underline{U}_g \\ \underline{Y}_g(t) &= \underline{C}_g \underline{X}_g(t) + \underline{D}_g \underline{U}_g\end{aligned}\tag{26}$$

Hierin is \underline{X}_g de toestand van het general control element, \underline{Y}_g de uitgang van het general control element en \underline{U}_g de ingang van het general control element (De benedenindex g slaat op het general control element!). De matrices \underline{A}_g , \underline{B}_g , \underline{C}_g en \underline{D}_g kunnen door de gebruiker worden ingevoerd. Verder is het mogelijk de ingang \underline{U}_g op te geven. Dit gebeurt door met behulp van een text editor de formatted file van het model aan te passen. Hoe en in welke vorm dit gebeuren moet staat in Bijlage.6.

Door nu de waarden van de toestand \underline{X} van het mechanische model van de dubbele slinger (deze is: $(x, \Theta_1, \Theta_2, \dot{x}, \dot{\Theta}_1, \dot{\Theta}_2)$) in de ingang \underline{U}_g van het general control element te stoppen en voor de matrix \underline{D}_g de optimale terugkoppelmatrix $-K$ te nemen is eenvoudig de grootte van de kracht F_u te bepalen. Hiervoor moet bij een één-dimensionale toestandsvector \underline{X}_g van het general control element (een nul-dimensionale toestandsvector laat het general control element niet toe) $\underline{C}_g=0$ genomen worden. Verder zijn de matrices \underline{A}_g en \underline{B}_g niet van belang en worden ook gelijk aan 0 gemaakt.

Een schema van deze regeling is in Bijlage.10 te zien en in Bijlage.11 zijn de bijbehorende verbose en aangepaste formatted file te zien. De resultaten van een

simulatie met deze regeling zijn identiek aan de resultaten uit paragraaf 5.2.3.

Het voordeel van deze toepassing is dat de terugkoppelmatrix K die in Matlab bepaald is, na aanpassen aan de correcte syntax (Bijlage.6), direct in de formatted file van het model kan worden gekopieerd. Bij aanpassingen van het model is deze toepassing dus aan te raden omdat in het model uit paragraaf 5.2.3 bij veranderingen in de K-matrix alle control amplifier elementen moeten worden aangepast. Bij de toepassing van het general control element moet alleen de aangepaste matlab file voor de K-matrix via een editor in de formatted file geladen worden, wat bij grote matrices veel voordeel biedt. Bij het gebruik van een kleine toestandsvector is het voor beide methoden even bewerkelijk om aanpassingen uit te voeren. Verder is het uitbreiden van het model (bijvoorbeeld een grotere toestandsvector) voor beide toepassingen even bewerkelijk.

5.3: Het toepassen van flexibele elementen.

Er is tevens nagegaan of er problemen ontstaan bij het gebruik van het linearisatie element als er in plaats van starre body's gebruik wordt gemaakt van flexibele body's. Hiervoor is onder andere weer gekeken naar een linearisatie van een omgekeerde dubbele slinger (Fig.3) waarbij de twee slingers flexibel worden verondersteld.

Bij de modellering van flexibele lichamen verwacht DADS gegevens over het flexibele lichaam in de vorm van resultaten van een statische of dynamische EEM-analyse. Er bestaan interfaces die de resultaten van zo'n EEM-analyse verwerken tot een set gegevens waarmee DADS overweg kan. In dit verslag is alleen van belang of er problemen bij de linearisaties op zullen treden. Er is dan ook gebruik gemaakt van een aanpak volgens de "assumed modes" methode waarbij de door DADS gevraagde gegevens op een EEM-achtige manier geformuleerd worden. Voor meer informatie over deze methode verwijs ik naar [Lit.4] en [Lit.5] hoofdstuk.7.

Het programma FLEX is een programma dat gebruik maakt van deze "assumed modes" methode. Het genereert de voor DADS benodigde informatie van een flexibele balk. Bij het opstarten van het programma FLEX moet worden opgegeven of er met één statische mode of één of meer dynamische modes gewerkt moet

worden. Vervolgens moet het aantal knooppunten en het aantal mee te nemen modes voor de betreffende body ingevoerd worden. Het programma maakt de datafile FLEX.FLE aan die gebaseerd is op een balk met een massa van 0.0667557kg, een lengte van 0.3m en een massatraagheidsmoment van 0.000501 kgm² opgedeeld in het opgegeven aantal knooppunten. Bij een balk (body) van andere afmetingen moeten de afmetingen van de balk in het programma FLEX worden aangepast.

Omdat FLEX alleen in 2D werkt moet het model in de preprocessor in 2D gemodelleerd worden. Bij de body's die flexibel zijn moet in het body-element de flag "flexible" op true worden gezet. Tevens moet er voor elke flexibele body een flexibele body element gegenereerd worden volgens:

```
name = naam flexibel body element
body.name = naam van de betreffende body
file.name = FLEX.FLE (of andere naam.FLE)
number.modes = aantal modes dat mee moet worden genomen voor deze body
rest = default
```

Voor elke mode moet een initial condition element worden aangemaakt volgens:

```
name = naam initial condition element
body.1.name = naam van de betreffende body
type.initial.cond. = mode.shape
extra.coord = nummer van mode
```

Ten slotte moeten in de revolute joints de waarden van node.1 en node.2 als het nodig is een waarde krijgen.

Er is dus opnieuw een omgekeerde dubbele slinger in DADS gemodelleerd. In dit geval wordt een 2D model gebruikt en kunnen dus de hoeken van de rsda elementen niet als vrijheidsgraden gedefinieerd worden (Bijlage.2). Er wordt aan DADS overgelaten met welke vrijheidsgraden het model gelineariseerd wordt. De twee slingers worden als flexibele body's gedefinieerd met 11 knooppunten en 1 dynamische mode per slinger. Voor het karretje wordt de massa=0.2kg genomen, verder zijn de massa's van de slingers 0.0667557kg en de lengtes van de slingers 0.3m. De verbose file van dit model is in Bijlage.12 te zien.

Bij het lineariseren neemt DADS als linearisatie coördinaten van de twee slingers de hoeken van de lokale body assenstelsels ten opzichte van het globale assenstelsel (hoek phi in DADS-body, starre rotatie), de x-positie van het zwaartepunt

van de bovenste slinger, en de twee modes van de twee flexibele body's (vervorming flexibele body's). Deze laatste twee coördinaten zijn dus het gevolg van het gebruik van flexibele body's. De toestandsvector van het systeem wordt dus uitgebreid met de modes en de afgeleiden van de modes. Linearisatie op basis van de hoeken Θ_1 (hoek tussen karretje en het uiteinde van de onderste slinger) en Θ_2 (hoek tussen het andere uiteinde van de onderste slinger en het uiteinde van de bovenste slinger) uit Fig.3 is niet meer mogelijk. Uitgangen van hoekopnemers kunnen dus geen toestanden meer zijn.

H 6: VERIFICATIE VAN DE REGELRESULTATEN.

Om enige controle op de regeling van een mechanisch model in DADS te hebben zijn er een aantal simulaties gedaan in een programma dat veel binnen de regeltechniek gebruikt wordt. PC-Matlab is zo'n programma. In dit hoofdstuk wordt een klassiek voorbeeld, de omgekeerde slinger, in DADS en in PC-Matlab gesimuleerd.

In fig.5 is een tekening van dit model te zien. voor de massa's en lengtes zijn de volgende waarden genomen: $m_1 = 0.2\text{kg}$, $m_2 = 0.01\text{kg}$ en $l_2 = 0.5\text{m}$. De kolom met gegeneraliseerde coördinaten is : $\underline{q} = [x \ \Theta]^T$. Het model is in DADS eenvoudig te verkrijgen door het model van de dubbele slinger uit het vorige hoofdstuk te gebruiken waarbij de gedeelten (slinger en regeling) die

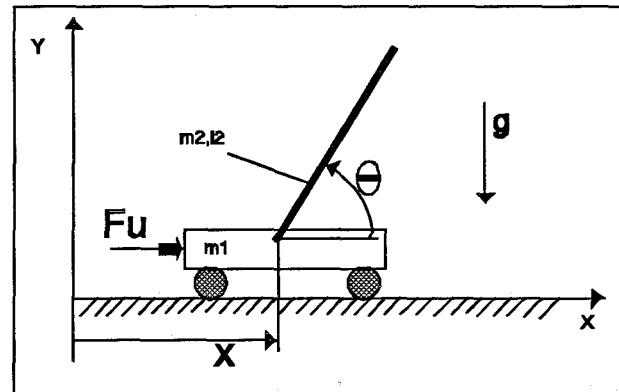


Fig.5

bij de bovenste slinger horen weg kunnen worden gelaten. De waarden van de amplifler control elementen moeten wel worden aangepast met de opnieuw te berekenen elementen van terugkoppelmatrix K.

Om een model in PC-Matlab te realiseren zijn de bewegingsvergelijkingen van het model nodig. Voor de kinetische energie T van het systeem geldt:

$$T = \frac{1}{2}(m_1 + m_2) \dot{x}^2 - \frac{1}{2}m_2 l_2 \dot{\Theta} x \sin \Theta + \frac{1}{2}m_2 l_2^2 \dot{\Theta}^2 / 6 \quad (27)$$

Voor de potentiële energie V geldt:

$$V = \frac{1}{2}l_2 m_2 g \sin \Theta \quad (28)$$

met $g=9.81\text{m/s}^2$. Voor de kolom met de op de vrijheidsgraden betrokken gegeneraliseerde krachten Q geldt:

$$Q = [Fu \ 0]^T \quad (29)$$

Door de vergelijkingen van Lagrange te gebruiken zijn de bewegingsvergelijkingen af te leiden. De vergelijkingen van Lagrange zijn ([Lit.3] formule(2.5.18)):

$$\frac{d}{dt} (T_{,\dot{q}}) - T_{,q} + V_{,q} = Q \quad (30)$$

Door nu (27), (28) en (29) in (30) in te vullen en verder uit te werken krijgen we de bewegingsvergelijkingen:

$$(m_1 + m_2) \ddot{x} - \frac{1}{2} m_2 l_2 \ddot{\theta} \sin \theta - \frac{1}{2} m_2 l_2 \dot{\theta}^2 \cos \theta = Fu \quad (31)$$

en

$$-\frac{1}{2} m_2 l_2 \ddot{x} \sin \theta + m_2 l_2^2 \ddot{\theta} / 3 + \frac{1}{2} l_2 m_2 g \cos \theta = 0 \quad (32)$$

Na het introduceren van een toestandsvector $\underline{X} = [x \ \theta \ \dot{x} \ \dot{\theta}]^T$ volgt uit (31) en (32):

$$\dot{\underline{X}} = f(\underline{X}, Fu) = \begin{bmatrix} \dot{x} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{\theta} \\ x_{\text{dotdot}} \\ \frac{3}{2l_2} (x_{\text{dotdot}} \sin \theta - g \cos \theta) \end{bmatrix} \quad (33)$$

Met

$$x_{\text{dotdot}} = \frac{Fu - \frac{3}{4} m_2 g \sin \theta \cos \theta + \frac{1}{2} m_2 l_2 \dot{\theta}^2 \cos \theta}{m_1 + m_2 - \frac{3}{4} m_2 \sin^2 \theta} \quad (34)$$

Met formule (33) en (34) is het niet lineaire systeem in Matlab met behulp van de

functie ode23 te simuleren. De ingang F_u moet wel eerst bepaald worden en is weer met behulp van de optimale regelwet te bepalen. Door het systeem te lineariseren zijn de systeemmatrix A en de ingangsmatrix B weer te bepalen. Deze worden dan vervolgens na invullen van de lengte en massa's:

$$A = \frac{\partial f(\underline{X}, F_u)}{\partial \underline{q}} \Big|_{\underline{X}_0, F_{u_0}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{3gm_2}{4m_1+m_2} & 0 & 0 \\ 0 & \frac{3g}{2I_2} + \frac{9gm_2}{2I_2(4m_1+m_2)} & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0.363 & 0 & 0 \\ 0 & 30.52 & 0 & 0 \end{bmatrix} \quad (35)$$

en

$$B = \frac{\partial f(\underline{X}, F_u)}{\partial F_u} \Big|_{\underline{X}_0, F_{u_0}} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m_1+m_2/4} \\ \frac{3}{2I_2(m_1+m_2/4)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 4.938 \\ 14.81 \end{bmatrix} \quad (36)$$

De linearisatie vindt plaats in het werkpunt $\underline{X}_0 = [0 \ \pi/2 \ 0 \ 0]^T$ en $F_{u_0} = 0$. De berekende matrices kunnen ook in DADS met het linearisatie element worden bepaald. Met behulp van deze twee matrices is de optimale terugkoppel matrix K in matlab te bepalen. Hierdoor is $F_u = -K\delta\underline{X}$ ook weer bepaald. De optimale terugkoppelmatrix wordt dan:

$$K = [-1 \quad 10,169 \quad -1,629 \quad 2,105] \quad (37)$$

Nu de ingang F_u ook bekend is kan het systeem in matlab en in DADS worden gesimuleerd. Het programma dat voor de simulatie in matlab nodig is staat in Bijlage.13. De DADS verbose-file is met enige aanpassingen eenvoudig uit het model uit hoofdstuk 5 te verkrijgen.

De resultaten van de simulaties in Matlab en in DADS staan in Fig.6. De resultaten uit Matlab worden met een lijn weergegeven. De plusjes (+) geven de resultaten uit

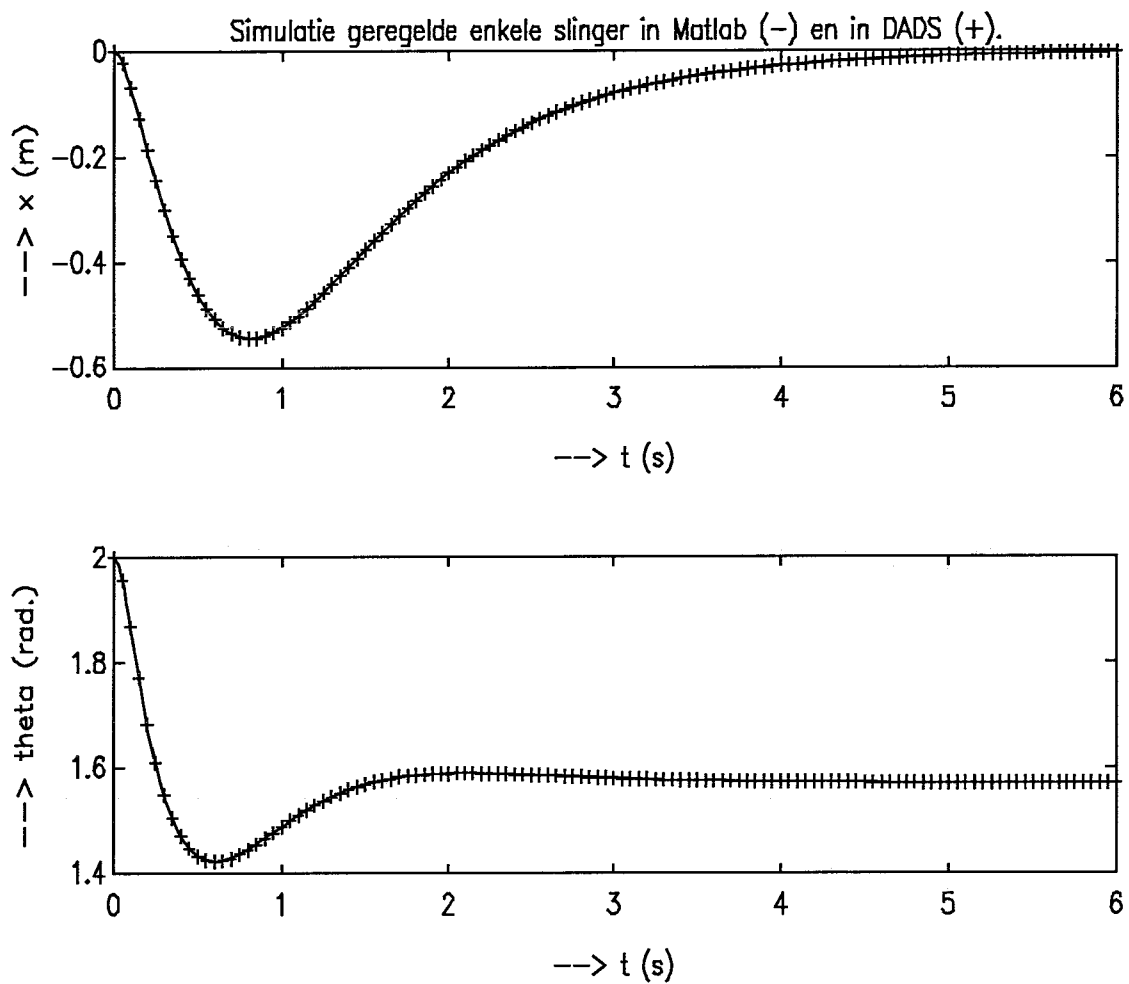


Fig.6

DADS weer. In Fig.6 is de x-positie van het wagentje tegen de tijd weergegeven en is de rotatie θ van de slinger tegen de tijd uitgezet. De resultaten uit Matlab en DADS blijken goed met elkaar overeen te komen. Er ontstaan geen opmerkelijke verschillen tussen de simulaties in de twee programma's.

BESPREKING VAN DE RESULTATEN.

Om een linearisatie op een systeem in DADS uit te voeren is het nodig je model uit te breiden met een linearisatie element. Als DADS niet de coördinaten genereert die bij de linearisatie gewenst zijn moeten deze coördinaten in een externe file gedefinieerd worden. Tenslotte moet er na een simulatie het gelineariseerde model in de postprocessor naar een file worden weggeschreven. Kortom er moeten nogal wat handelingen verricht worden voordat er een linearisatie bepaald wordt.

Het soort coördinaat dat je bij de linearisatie kunt definiëren is beperkt. Bijvoorbeeld een nogal voor de hand liggende coördinaat die niet gebruikt kan worden is `rel.angle` in een `rsda` element in het 2D geval.

Het is door deze linearisatiemodule mogelijk geworden om bijvoorbeeld een regeling met optimale terugkoppeling voor een ingewikkeld niet-lineair systeem te ontwerpen.

De resultaten van de bewerkingen in andere programmatuur (bijvoorbeeld het ontwerpen van een optimale regeling in Matlab) is op het DADS-model toe te passen. Dit gebeurt op een omslachtige manier. Bij het gebruik van het `general control` element moet de data die in Matlab bepaald is worden aangepast aan een bepaalde syntax voordat het in DADS te gebruiken is. Bij het gebruik van `amplifier control` elementen moeten alle versterkingsfactoren handmatig worden aangepast.

LITERATUURLIJST.

- [Lit.1] M.W.C.M. van de Oetelaar, Het simuleren van geregelde mechanische systemen met DADS, WFW-rapport TU Eindhoven nr.89006 januari 1989.
- [Lit.2] J.J. Kok, Werktuigkundige regeltechniek 2, Collegedictaat TU Eindhoven nr.4594 1990.
- [Lit.3] D.H. van Campen en A. de Kraker, Het dynamisch gedrag van constructies, Collegedictaat TU Eindhoven nr.4552 1989/1990.
- [Lit.4] W.P. Koppens, The dynamics of systems of deformable body's, Dissertatie TU Eindhoven, 1989
- [Lit.5] A.Sauren, Multibody dynamica, Collegedictaat TU Eindhoven nr.4659 september 1990.

Bijlage.1

De beschrijvingen van de te definiëren parameters in het linearisatie element zijn:

Variabele: number.states.

Number.states moet een integer zijn die het aantal gegeneraliseerde coördinaten weergeeft. Deze coördinaten moeten door de gebruiker in een externe file gedefinieerd worden. Bij een waarde groter dan nul moet onderstaande file, aangegeven met file.name, een naam krijgen. Bij invullen van een nul hoeft er geen naam voor de file te worden opgegeven en gebruikt DADS de intern gegenereerde coördinaten. Als number.states ongelijk is aan het aantal onafhankelijke coördinaten of de opgegeven coördinaten zijn afhankelijk dan gebruikt DADS tevens de intern gegenereerde coördinaten.

Variabele: file.name.

Voor file.name moet de naam in worden gevuld van de externe ASCII file waarin de coördinaten gedefinieerd zijn die voor de linearisatie gebruikt worden (alleen de coördinaten, niet hun tijdsafgeleiden). Deze file.name is maximaal 80 karakters lang. De toestand van de gebruikte control-elementen kan ook hier gedefinieerd worden. De file moet de namen van de elementen bevatten en de gegeneraliseerde coördinaten van elk element dat voor de linearisatie gebruikt wordt, gescheiden door een spatie. Voor elke coördinaat moet dit op een nieuwe regel gebeuren.

Variabele: perturb.factor.

Deze factor (real) definieert de initiële perturbatiefactor die gebruikt wordt in de numerieke methode die de gelineariseerde A en B matrices bepaalt. De initiële perturbatie van elke toestand krijg je door deze factor te vermenigvuldigen met de toestand. Bij de volgende iteraties wordt deze factor gehalveerd totdat het minimum bereikt wordt. De initiële factor is kleiner dan 0.01 en de default waarde is 0.0001.

Variabele: minimum.perturb.

Deze factor (real) geeft het minimum weer dat door de bovenstaande perturbatie factor bereikt mag worden. Bij een kleinere perturbatie factor wordt deze minimum perturbatiefactor gebruikt. De default waarde is 0.00001.

Variabele: convergence.tol.

Dit is een real getal dat aangeeft wanneer de linearisatie iteraties moeten worden afgebroken. Als de numerieke waarden van de coördinaten van twee opeenvolgende iteratiestappen minder verschillen dan de convergence.tol dan wordt het iteratieproces gestopt. Als er meer dan 5 iteraties worden uitgevoerd zonder te convergeren dan wordt er een foutmelding gegeven. De default waarde is 0.0001.

Bijlage.2

De coördinaten van bepaalde elementen die in de externe ASCII file gedefinieerd kunnen worden zijn:

| <u>Element type</u> | <u>Coördinaat naam</u> |
|----------------------------------|-----------------------------------|
| Voor 2D-modellen: | |
| Rigid body | x, y, phi |
| Point follower | rel.angle |
| General cam/follower | rel.angle.1, rel.angle.2 |
| Cam/flat-faced follower | rel.angle |
| Rack/convex pinion | rel.angle |
| Flexible body | mode.n, n=1...aantal modes |
| Voor 3D-modellen: | |
| Rigid body | x, y, z, e0, e1, e2, e3 |
| Rsda | rel.angle |
| Rack and pinion | rel.angle |
| Screw joint | rel.angle |
| Gear joint | rel.angle.1, rel.angle.2 |
| Flexible body | mode.n, n=1...aantal modes |
| Track superelement | track.loop, road.wh.n, n=1...12 |
| Driver | rel.angle |
| Voor 2D en 3D control elementen: | |
| First order system | state.var |
| Second order system | st.var.1, st.var.2 |
| Integrator | state.var |
| General | st.var.n, n=1...aantal toestanden |
| Servo valve | st.var.1, st.var.2 |
| Single acting actuator | state.var |
| Double acting actuator | st.var.1, st.var.2 |
| Accumulator | state.var |

Bijlage.3

HET OUTPUT CONTROL ELEMENT.

Dit element oefent een kracht of moment met de waarde van een knooppuntsvariabele uit op een DADS-body. De variabelen die gedefinieerd moeten worden zijn:

| | |
|-----------------|--|
| NAME | = naam van het output control element. |
| OUTPUT.NODE | = naam van de knooppuntsvariabele waarvan de grootte als belasting op een body in het mechanische systeem zal worden uitgeoefend. |
| TYPE | = Type belasting dat uit zal worden geoefend. Er kan gekozen worden uit: X.FORCE Y.FORCE Z.FORCE XL.FORCE YL.FORCE ZL.FORCE XL.TORQUE YL.TORQUE ZL.TORQUE FORCE.DIFF TORQUE.DIFF MASS IXX, IYY, IZZ, IXY, IXZ, IYZ |
| BODY.1.NAME | = naam van de eerste body waarmee dit element verbonden is. |
| BODY.2.NAME | = naam van eventuele tweede body waarmee dit element verbonden is. |
| P.ON.BODY.1 | = Het punt op body 1 waarop het output element werkt. Dit punt kan gedefinieerd worden in het lokale body reference frame, het non-centroidal body-fixed reference frame of het globale reference-frame. |
| P.ON.BODY.2 | = Het punt op body 2 waarop het output element werkt. Dit punt kan gedefinieerd worden in het lokale body reference frame, het non-centroidal body-fixed reference frame of het globale reference-frame. |
| JOINT.NAME | = Moet waarde 'NONE' hebben. |
| FLEXIBLE.NODE.1 | = Een niet negatieve integer die het knooppunt van het eventueel gebruikte eerste flexibele body aangeeft waarop het output element werkt. |
| FLEXIBLE.NODE.2 | = Een niet negatieve integer die het knooppunt van het eventueel gebruikte tweede flexibele body aangeeft waarop het output element werkt. |

Bijlage.4

HET INPUT CONTROL ELEMENT.

Dit element kent een waarde toe aan een knooppuntsvariabele. Deze waarde kan een functie zijn van de tijd of kan gelijk worden gemaakt aan de waarde van een gegeneraliseerde coördinaat uit DADS. Bij de laatste optie is het nodig het type coördinaat op te geven en de naam van de body te geven. De variabelen die gedefinieerd moeten worden zijn:

| | |
|---------------------|--|
| NAME | = Naam input control element. |
| NODE.NAME | = Naam van de knooppuntsvariabele die in dit element een waarde krijgt. |
| TYPE | = Type knooppuntsvariabele. De keuze is uit: Input van een DADS coördinaten: X Y Z XD YD ZD XDD YDD ZDD XL.OMEGA YL.OMEGA ZL.OMEGA XL.OMEGAD YL.OMEGAD ZL.OMEGAD DISTANCE DISTANCED DISTANCEDD LOCAL.TRANS.VEL LOCAL.TRANS.ACC LOCAL.ROT.VEL LOCAL.ROT.ACC EXTRA LAMBDA Input van functies: STEP HARMONIC RAMP CURVE POLYNOMIAL USER.INPUT |
| BODY.1.NAME | = Naam van eerste body waarvan de coördinaat gelezen kan worden. |
| BODY.2.NAME | = Naam van eventuele tweede body. |
| FUNCTION.PARAMETERS | = Parameters die het tijdsgedrag van de speciale functies bepalen. Deze parameters worden gebruikt als het type step, ramp, polynomial of harmonic gebruikt wordt. |
| P.ON.BODY.1 | = Het punt op body 1 waarop het input element werkt. Het punt kan worden gedefinieerd in het lokale assenstelsel, het non-centroidal body-fixed assenstelsel of het globale assenstelsel. |
| P.ON.BODY.2 | = Het punt op het body 2 waarop het input element werkt. Dit kan als P.ON.BODY.1 gedefinieerd worden. |
| Q.ON.BODY.1 | = Locatie van het punt Q op body 1. Dit punt definieert de as PQ waarlangs de volgende grootheden werken: |

| | |
|-----------------|--|
| | LOCAL.TRANS.VEL, LOCAL.TRANS.ACC LOCAL.ROT.VEL, LOCAL.ROT.ACC. Het punt kan weer worden gedefinieerd in een lokaal, globaal of een non-centroidal body-fixed assenstelsel. |
| Q.ON.BODY.2 | = Zie Q.on.body.1. |
| CURVE.NAME | = Naam van de curve die gebruikt wordt als het TYPE op CURVE wordt gezet. |
| JOINT.NAME | = Wordt niet gebruikt, NONE invullen |
| FLEXIBLE.NODE.1 | = Een niet negatieve integer dat het nummer aangeeft van het knooppunt op de eerste body waarop dit input element werkt. Een waarde 0 geeft aan dat de body niet flexibel is. |
| FLEXIBLE.NODE.2 | = Een niet negatieve integer dat het nummer aangeeft van het knooppunt op de tweede body waarop dit input element werkt. |
| ANGULAR.UNITS | = Type hoekeenheid dat gebruikt wordt. |
| EXTNUM | = Geeft het nummer van een gegeneraliseerd coördinaat aan, uit de lijst met gegeneraliseerde coördinaten of Lagrange multipliers, dat als input gebruikt kan worden. |
| SAMPLE.RATE | = De sample rate als dit element als een digital element gedefinieerd is. |
| DIGITAL.FLAG | = Vlag die aangeeft dat het input element als digital element wordt beschouwd. |
| LINEAR.FLAG | = Vlag die aangeeft dat bij een linearisatie de ingangsmatrix B moet worden bepaald. |

Bijlage.5

HET AMPLIFIER CONTROL ELEMENT.

Dit element bepaalt de uitgangknooppuntsvariabele als functie van de ingang-knooppuntsvariabele. Deze functie kan lineair of niet lineair zijn. In een FORTRAN routine kan elke willekeurige functie worden gedefinieerd. De volgende variabelen moeten gedefinieerd worden:

| | |
|-------------|---|
| NAME | = Naam van amplifier control element. |
| INPUT.NODE | = Naam van ingangknooppuntsvariabele. |
| OUTPUT.NODE | = Naam van uitgangknooppuntsvariabele. |
| TYPE | = Definieert het type van de versterking. Er kan gekozen worden tussen CONSTANT, CURVE, USER en CURVE_OUT. |
| CURVE.NAME | = Naam van curve element dat de overbrenging van de amplifier als functie van de ingangknooppuntsvariabele bepaald. |
| GAIN | = Waarde van de constante versterking als het TYPE CONSTANT gebruikt wordt. |

HET SUMMER CONTROL ELEMENT.

Het summer control element telt maximaal drie knooppuntsvariabelen op (of trekt ze van elkaar af) om de uitkomst vervolgens in de uitgangknooppuntsvariabele te zetten. De variabelen die gedefinieerd moeten worden zijn:

| | |
|---------------|---|
| NAME | = Naam summer control element. |
| OUTPUT.NODE | = Naam van uitgangknooppuntsvariabele. De waarde van deze variabele is de som van alle ingangknooppuntsvariabelen die vermenigvuldigd zijn met bijbehorende coëfficiënten (± 1) |
| INPUT.NODE.1 | = Naam van de eerste ingangknooppuntsvariabele. |
| INPUT.NODE.2 | = Naam van de tweede ingangknooppuntsvariabele. |
| INPUT.NODE.3 | = Naam van de derde ingangknooppuntsvariabele. |
| COEFFICIËNT.1 | = Teken van de eerste ingangknooppuntsvariabele. |
| COEFFICIËNT.2 | = Teken van de tweede ingangknooppuntsvariabele. |
| COEFFICIËNT.3 | = Teken van de derde ingangknooppuntsvariabele. |

INTEGRATOR CONTROL ELEMENT.

Integreert een variabele die als ingang wordt opgegeven in de tijd. De uitgang is de integraal van de ingangknooppuntsvariabele. Hier moet een beginvoorwaarde worden meegegeven. De volgende variabelen moeten worden gedefinieerd.

| | |
|-------------|---|
| NAME | = Naam integrator control element. |
| INPUT.NODE | = Naam ingangknooppuntsvariabele. |
| OUTPUT.NODE | = Naam uitgangknooppuntsvariabele. |
| S.ZERO | = Beginwaarde uitgangknooppuntsvariabele. |

Bijlage.6

HET GENERAL CONTROL ELEMENT.

Dit multi input multi output element modelleert een N^e orde linear control system in algemene toestandsvorm:

$$\begin{aligned}\dot{\underline{X}}_g(t) &= \underline{A}_g * \underline{X}_g(t) + \underline{B}_g * \underline{U}_g \\ \underline{Y}_g(t) &= \underline{C}_g * \underline{X}_g(t) + \underline{D}_g * \underline{U}_g\end{aligned}$$

Waarin

- \underline{U}_g = De inputvector met dimensie M.
- \underline{X}_g = De toestandsvector met dimensie N.
- \underline{Y}_g = De outputvector met dimensie L.
- \underline{A}_g = De systeemmatrix met dimensie N bij N.
- \underline{B}_g = De inputmatrix met dimensie N bij M.
- \underline{C}_g = De outputmatrix met dimensie L bij N.
- \underline{D}_g = De doorverbindingmatrix met dimensie L bij M.

De index g geeft aan dat de matrix of vector op het general control element slaat. De matrices \underline{A}_g , \underline{B}_g , \underline{C}_g en \underline{D}_g kunnen door de gebruiker in worden gevoerd. De dimensies van \underline{U}_g , \underline{X}_g en \underline{Y}_g moeten gespecificeerd worden in het general control element. De minimale dimensie die opgegeven mag worden is 1. Als het systeem in de onderstaande standaardvorm gedefinieerd wordt hoeft alleen de onderste rij van de \underline{A}_g matrix gespecificeerd te worden.

De variabelen die in dit element gedefinieerd moeten worden zijn:

- NAME = Naam general control element.
- NDIM = Dimensie van de toestandsvector.
- MDIM = Dimensie van de inputvector.
- LDIM = Dimensie van de outputvector.
- IFLAG = Vlag die specificeert of de begintoestandsvector van de onderstaande standaardvorm is. Als IFLAG=TRUE dan wordt voor de begintoestandsvector de nulvector genomen. Voor IFLAG=FALSE moet er zelf een begintoestandsvector gegeven worden.
- AFLAG = Vlag die aangeeft of de A-matrix van de onderstaande standaardvorm is. Als AFLAG=TRUE dan is A in de standaardvorm geschreven. Als AFLAG=FALSE moet de gebruiker de waarden van de matrix invoeren als hieronder beschreven staat.
- BFLAG = Als AFLAG maar dan voor matrix B.
- CFLAG = Als AFLAG maar dan voor matrix C.
- DFLAG = Als AFLAG maar dan voor matrix D.

De standaard vorm ziet er als volgt uit:

$$\dot{\underline{X}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \dots & 0 & 1 \\ -a_1 & -a_2 & -a_3 & \dots & \dots & \dots & -a_N \end{bmatrix} \underline{X} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ 1 \end{bmatrix} U$$

en

$$Y = [1 \quad 0 \quad 0 \quad \dots \quad 0 \quad 0] \underline{X}$$

Er kunnen dus vijf vlaggen aangepast worden. Bij een waarde TRUE van alle vlaggen wordt de bovenstaande standaardvorm gebruikt waarbij alleen de laatste rij van de A_g -matrix en de ingangsvector \underline{U}_g gedefinieerd moeten worden. Als een vlag op FALSE wordt gezet dan betekent dit dat de bijbehorende matrix of vector niet in de standaardvorm staat; nu moet de gehele matrix/vector door de gebruiker opgegeven worden.

Het invullen van de matrices en initiële vector gebeurt door het aanpassen van de formatted file in een editor. Eerst wordt het model in de Preprocessor gemaakt, inclusief de bovenstaande data voor het general control element. Na opslag van dit model moet de formatted file in een editor gelezen worden. Zoek vervolgens de regel op waarin GENERAL staat met de naam van het element en bijbehorende data. Op de volgende regel direct onder GENERAL moet alle benodigde data worden ingevoerd. Elke keer als het model in de preprocessor is aangepast moet dit weer opnieuw gebeuren. Het is daarom handig om alle onderstaande data in een aparte file te zetten en deze file in z'n geheel op de juiste plaats in de formatted file te plaatsen.

We beginnen met het invoeren van de namen van de ingangsknooppunten van de vector \underline{U}_g in hoofdletters. Als de naam niet 20 karakters lang is moet deze aan worden gevuld met spaties tot 20 karakters. De volgende naam wordt er direct achter gezet en moet weer aangevuld worden met spaties tot 20 karakters. Op een regel kunnen maximaal 4 namen. Bij meer dan vier ingangsknooppunten moet er aan het eind van de regel een [enter] worden gegeven en vervolgens moet er op de volgende regel met de vijfde naam verder worden gegaan.

Als de namen van de ingangsknooppuntsvariabelen van vector \underline{U}_g zijn ingevoerd moet op een zelfde wijze op een nieuwe regel worden verder gegaan met de invoer van de namen van de uitgangsknooppuntsvariabelen van vector \underline{Y}_g . Vervolgens moeten de eventueel in te voeren niet-standaard matrices op een nieuwe

regel worden ingetypt. Als een van de vijf vlaggen IFLAG, AFLAG, BFLAG, CFLAG en/of DFLAG FALSE zijn moet voor elke vlag de bijbehorende matrix/ vector worden ingevoerd. Deze invoer gaat weer met vier waarden per regel waarbij rij voor rij wordt ingevoerd.

De syntax voor deze invoer is:

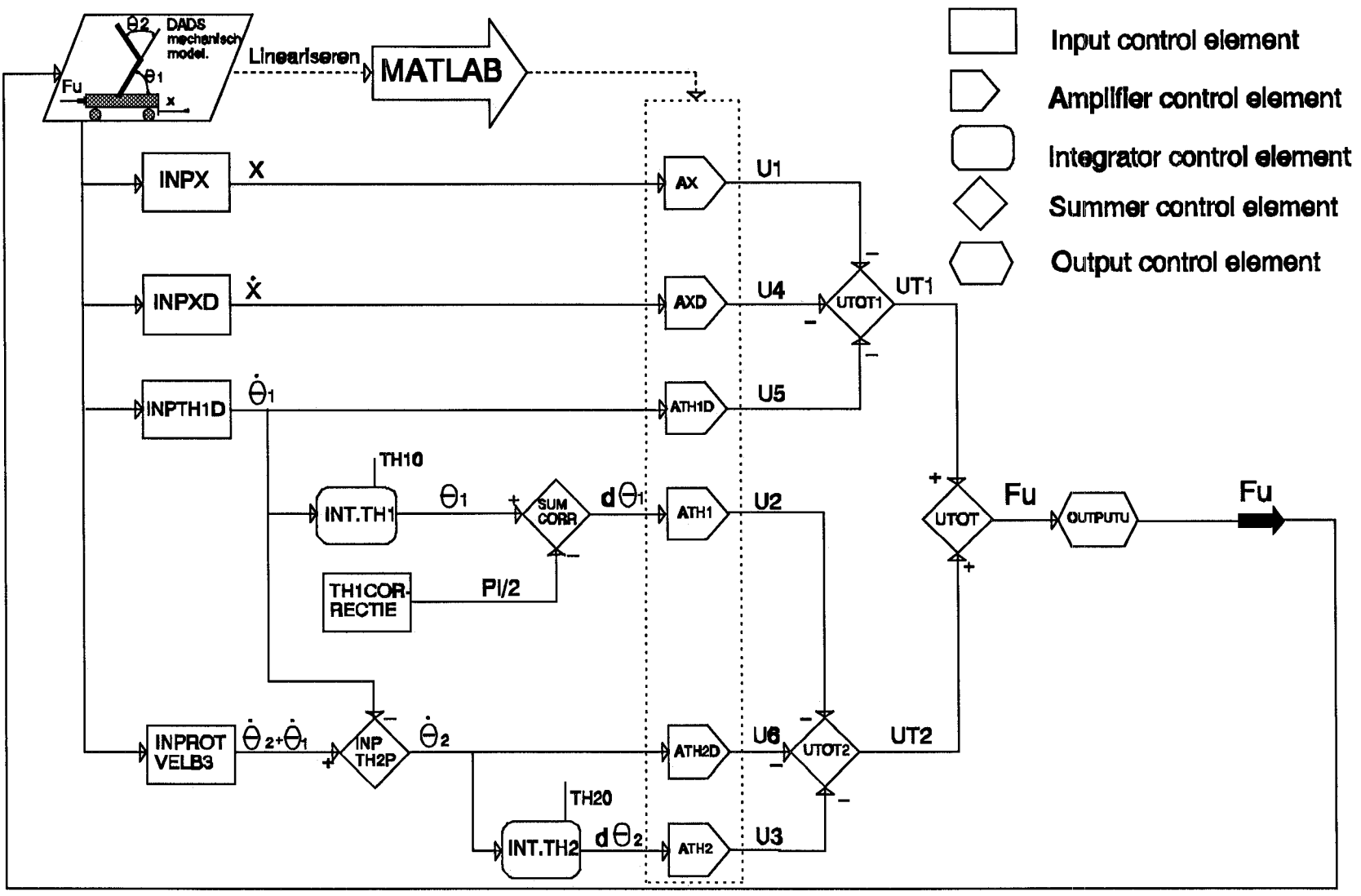
s.mmmmmmmmmmmmmEsee

waarbij s het teken (+/-) is van de mantissa (m) en de exponent (e). De mantissa wordt van de exponent gescheiden door E. Elke matrix begint op een nieuwe lijn en wordt in volgende volgorde ingevoerd:

- 1) begintoestandsvector
- 2) A matrix
- 3) B matrix
- 4) C matrix
- 5) D matrix

Deze data moet dus voor de matrices worden ingevoerd waarvan de bijbehorende vlag op FALSE staat. Als alle vlaggen op TRUE staan moeten alleen de ingang- en uitgangknooppuntsvariabelen ingevoerd worden. De laatste rij van de A-matrix moet tevens ingevoerd worden. Als de editor dit toe laat kunnen bijvoorbeeld externe matrices (bijvoorbeeld in matlab aangemaakt) direct in z'n geheel in de formatted file worden ingevoerd. De data moet dan wel eerst worden aangepast aan bovenstaande syntax.

BILAG8: SCHEMA REGELING DUBBELE SLINGER MET AMPLIFIER ELEMENTEN.



Bijlage.9

VERBOSE FILE GEREGELDE DUBBELE SLINGER.

De regeling is met behulp van amplifier control elementen toegepast.

```
ANALYSIS
CREATE SYSTEM.DATA
UNITS                := 'SI'
ANALYSIS.TYPE        := 'DYNAMIC'
STARTING.TIME        := '0.0'
ENDING.TIME          := '6.5'
PRINT.INTERVAL       := '0.05'
GRAVITY.SEA.LEVEL    := '9.80665'
X.GRAVITY            := '0.0'
Y.GRAVITY            := '-1.0'
Z.GRAVITY            := '0.0'
SCALE.GRAVITY.COEFF := '1.0'
MATRIX.OPERATIONS    := 'SPARSE'
REDUNDANCY.CHECK     := 'TRUE'
LU.TOL               := '1.0D-12'
ASSEMBLY.TOL         := '1.0D-3'
BYPASS.ASSEMBLY      := 'FALSE'
OUTPUT.FILE          := 'ALL'
REFERENCE.FRAME       := 'LOCAL'
DEBUG.FLAG           := 'TRUE'

UP
CREATE DYNAMIC.DATA
REACTION.FORCES      := 'TRUE'
FORCE.COORDINATES    := 'BODY'
PRINT.METHOD        := 'INTERPOLATED'
MAX.INT.STEP         := '0.05'
SOLUTION.TOL         := '0.005'
INTEGRATION.TOL      := '0.001'
METHOD.INTEGRATION   := 'VARIABLE'
PRINT.FREQ           := '0'

UP
UP
FORCE
CREATE RSDA
NAME                 := 'RSDA1'
JOINT.NAME           := 'REV1'
ORIENTATION.ANGLE    := '0.0'
SPRING.CONSTANT      := '0.0'
DAMPING.COEFFICIENT := '0.0'
ACTUATOR.TORQUE      := '0.0'
CURVE.SPRING         := 'NONE'
CURVE.DAMPER         := 'NONE'
CURVE.ACTUATOR       := 'NONE'
ANGULAR.UNITS        := 'RADIANS'
TYPE                 := 'BIDIRECTIONAL'

UP
CREATE RSDA
NAME                 := 'RSDA2'
JOINT.NAME           := 'REV2'
ORIENTATION.ANGLE    := '0.0'
SPRING.CONSTANT      := '0.0'
DAMPING.COEFFICIENT := '0.0'
ACTUATOR.TORQUE      := '0.0'
CURVE.SPRING         := 'NONE'
CURVE.DAMPER         := 'NONE'
CURVE.ACTUATOR       := 'NONE'
ANGULAR.UNITS        := 'RADIANS'
TYPE                 := 'BIDIRECTIONAL'

UP
UP
JOINTS
CREATE REVOLUTE.JOINT
NAME                 := 'REV1'
BODY.1.NAME          := 'B1'
BODY.2.NAME          := 'B2'
P.ON.BODY.1          := ( 0.0, 0.0, 0.0 )
P.ON.BODY.2          := ( -0.25, 0.0, 0.0 )
Q.ON.BODY.1          := ( 0.0, 0.0, 1.0 )
Q.ON.BODY.2          := ( -0.25, 0.0, 1.0 )
R.ON.BODY.1          := ( 1.0, 0.0, 0.0 )
R.ON.BODY.2          := ( 0.75, 0.0, 0.0 )
NODE.1               := '0'
NODE.2               := '0'

UP
CREATE REVOLUTE.JOINT
NAME                 := 'REV2'
BODY.1.NAME          := 'B2'
BODY.2.NAME          := 'B3'
P.ON.BODY.1          := ( 0.25, 0.0, 0.0 )
P.ON.BODY.2          := ( -0.35, 0.0, 0.0 )
Q.ON.BODY.1          := ( 0.25, 0.0, 1.0 )
Q.ON.BODY.2          := ( -0.35, 0.0, 1.0 )
R.ON.BODY.1          := ( 1.25, 0.0, 0.0 )
R.ON.BODY.2          := ( 0.65, 0.0, 0.0 )
NODE.1               := '0'
NODE.2               := '0'

UP
CREATE TRANSLATIONAL.JOINT
NAME                 := 'TRANSL'
BODY.1.NAME          := 'B0'
BODY.2.NAME          := 'B1'
P.ON.BODY.1          := ( 0.0, 0.0, 0.0 )
P.ON.BODY.2          := ( 0.0, 0.0, 0.0 )
Q.ON.BODY.1          := ( 1.0, 0.0, 0.0 )
Q.ON.BODY.2          := ( 1.0, 0.0, 0.0 )
R.ON.BODY.1          := ( 0.0, 1.0, 0.0 )
R.ON.BODY.2          := ( 0.0, 1.0, 0.0 )
NODE.1               := '0'
NODE.2               := '0'

UP
CONTROLS
CREATE AMPLIFIER
NAME                 := 'AX'
INPUT.NODE           := 'X'
OUTPUT.NODE          := 'U1'
TYPE                 := 'CONSTANT'
GAIN                 := '1'
CURVE.NAME           := 'NONE'

UP
CREATE AMPLIFIER
NAME                 := 'AXD'
INPUT.NODE           := 'XD'
OUTPUT.NODE          := 'U4'
TYPE                 := 'CONSTANT'
GAIN                 := '2.0388'
CURVE.NAME           := 'NONE'

UP
CREATE AMPLIFIER
NAME                 := 'ATH1D'
INPUT.NODE           := 'THETA1D'
OUTPUT.NODE          := 'U5'
TYPE                 := 'CONSTANT'
GAIN                 := '-6.9997'
CURVE.NAME           := 'NONE'

UP
CREATE AMPLIFIER
NAME                 := 'ATH2D'
```

```

INPUT.NODE
OUTPUT.NODE
TYPE
GAIN
CURVE.NAME
UP
CREATE AMPLIFIER
NAME
INPUT.NODE
OUTPUT.NODE
TYPE
GAIN
CURVE.NAME
UP
CREATE AMPLIFIER
NAME
INPUT.NODE
OUTPUT.NODE
TYPE
GAIN
CURVE.NAME
UP
CREATE INPUT.FUNCTION
NAME
NODE.NAME
TYPE
BODY.1.NAME
BODY.2.NAME
FUNCTION.PARAMETERS
P.ON.BODY.1
P.ON.BODY.2
Q.ON.BODY.1
Q.ON.BODY.2
CURVE.NAME
JOINT.NAME
FLEXIBLE.NODE.1
FLEXIBLE.NODE.2
ANGULAR.UNITS
EXTNUM
SAMPLE.RATE
DIGITAL.FLAG
LINEAR.FLAG
UP
CREATE INPUT.FUNCTION
NAME
NODE.NAME
TYPE
BODY.1.NAME
BODY.2.NAME
FUNCTION.PARAMETERS
P.ON.BODY.1
P.ON.BODY.2
Q.ON.BODY.1
Q.ON.BODY.2
CURVE.NAME
JOINT.NAME
FLEXIBLE.NODE.1
FLEXIBLE.NODE.2
ANGULAR.UNITS
EXTNUM
SAMPLE.RATE
DIGITAL.FLAG
LINEAR.FLAG
UP
CREATE INPUT.FUNCTION
NAME
NODE.NAME
TYPE
BODY.1.NAME
BODY.2.NAME
FUNCTION.PARAMETERS
P.ON.BODY.1
P.ON.BODY.2
Q.ON.BODY.1
Q.ON.BODY.2
CURVE.NAME
JOINT.NAME
FLEXIBLE.NODE.1
FLEXIBLE.NODE.2
ANGULAR.UNITS
EXTNUM
SAMPLE.RATE
DIGITAL.FLAG
LINEAR.FLAG
UP
CREATE INPUT.FUNCTION
NAME
NODE.NAME
TYPE
BODY.1.NAME
BODY.2.NAME
FUNCTION.PARAMETERS
P.ON.BODY.1
P.ON.BODY.2
Q.ON.BODY.1
Q.ON.BODY.2
CURVE.NAME
JOINT.NAME
FLEXIBLE.NODE.1
FLEXIBLE.NODE.2
ANGULAR.UNITS
EXTNUM
SAMPLE.RATE
DIGITAL.FLAG
LINEAR.FLAG
UP
CREATE INPUT.FUNCTION
NAME
NODE.NAME
TYPE
BODY.1.NAME
BODY.2.NAME
FUNCTION.PARAMETERS
P.ON.BODY.1
P.ON.BODY.2
Q.ON.BODY.1
Q.ON.BODY.2
CURVE.NAME
JOINT.NAME
FLEXIBLE.NODE.1
FLEXIBLE.NODE.2
ANGULAR.UNITS
EXTNUM
SAMPLE.RATE
DIGITAL.FLAG
LINEAR.FLAG
UP
CREATE INPUT.FUNCTION
NAME
NODE.NAME
TYPE
BODY.1.NAME
BODY.2.NAME
FUNCTION.PARAMETERS
P.ON.BODY.1
P.ON.BODY.2
Q.ON.BODY.1
Q.ON.BODY.2
CURVE.NAME
JOINT.NAME
FLEXIBLE.NODE.1
FLEXIBLE.NODE.2
ANGULAR.UNITS
EXTNUM
SAMPLE.RATE
DIGITAL.FLAG
LINEAR.FLAG
UP
CREATE INTEGRATOR

```

```

== 'THETA2D'
== 'U6'
== 'CONSTANT'
== '-7.5636'
== 'NONE'

== 'ATH1'
== 'DELTATH1'
== 'U2'
== 'CONSTANT'
== '-13.3251'
== 'NONE'

== 'ATH2'
== 'THETA2'
== 'U3'
== 'CONSTANT'
== '-47.7753'
== 'NONE'

== 'INPX'
== 'X'
== 'B1'
== 'NONE'
== ( 0.0, 0.0, 0.0, 0.0 )
== ( 0.0, 0.0, 0.0 )
== ( 0.0, 0.0, 0.0 )
== ( 1.0, 0.0, 0.0 )
== ( 1.0, 0.0, 0.0 )
== 'NONE'
== 'NONE'
== '0'
== '0'
== 'RADIANS'
== '0.0'
== 'FALSE'
== 'FALSE'

== 'INPXD'
== 'XD'
== 'XD'
== 'B1'
== 'NONE'
== ( 0.0, 0.0, 0.0, 0.0 )
== ( 0.0, 0.0, 0.0 )
== ( 0.0, 0.0, 0.0 )
== ( 1.0, 0.0, 0.0 )
== ( 1.0, 0.0, 0.0 )
== 'NONE'
== 'NONE'
== '0'
== '0'
== 'RADIANS'
== '0.0'
== 'FALSE'
== 'FALSE'

== 'INPTHID'
== 'THETA1D'
== 'ZL.OMEGA'
== 'B2'
== 'NONE'
== ( 0.0, 0.0, 0.0, 0.0 )
== ( 0.0, 0.0, 0.0 )
== ( 0.0, 0.0, 0.0 )
== ( 1.0, 0.0, 0.0 )
== ( 1.0, 0.0, 0.0 )
== 'NONE'
== 'NONE'
== '0'
== '0'
== 'RADIANS'
== '0'
== '0.0'
== 'FALSE'
== 'FALSE'

== 'INPROTVELB3'
== 'ROTVELB3'
== 'ZL.OMEGA'
== 'B3'
== 'NONE'
== ( 0.0, 0.0, 0.0, 0.0 )
== ( 0.0, 0.0, 0.0 )
== ( 0.0, 0.0, 0.0 )
== ( 1.0, 0.0, 0.0 )
== ( 1.0, 0.0, 0.0 )
== 'NONE'
== 'NONE'
== '0'
== '0'
== 'RADIANS'
== '0.0'
== 'FALSE'
== 'FALSE'

== 'THICORRECTIE'
== 'THICORR'
== 'POLYNOMIAL'
== 'NONE'
== 'NONE'
== ( 1.570796327, 0, 0, 0 )
== ( 0.0, 0.0, 0.0 )
== ( 0.0, 0.0, 0.0 )
== ( 1.0, 0.0, 0.0 )
== ( 1.0, 0.0, 0.0 )
== 'NONE'
== 'NONE'
== '0'
== '0'
== 'DEGREES'
== '0.0'
== 'FALSE'
== 'FALSE'

```

```

NAME                               := 'INT.TH1'
INPUT.NODE                          := 'THETA1D'
OUTPUT.NODE                          := 'THETA1'
S.ZERO                               := '1.3963'

UP
CREATE INTEGRATOR
NAME                                 := 'INT.TH2'
INPUT.NODE                           := 'THETA2D'
OUTPUT.NODE                           := 'THETA2'
S.ZERO                               := '0.0'

UP
CREATE OUTPUT
NAME                                 := 'OUTPUTU'
OUTPUT.NODE                           := 'FU'
TYPE                                  := 'X.FORCE'
BODY.1.NAME                           := 'B1'
BODY.2.NAME                           := 'NONE'
P.ON.BODY.1                           := ( 0.0, 0.0, 0.0 )
P.ON.BODY.2                           := ( 0.0, 0.0, 0.0 )
JOINT.NAME                             := 'NONE'
FLEXIBLE.NODE.1                       := '0'
FLEXIBLE.NODE.2                       := '0'

UP
CREATE SUMMER
NAME                                 := 'INPTH2D'
OUTPUT.NODE                           := 'THETA2D'
INPUT.NODE.1                          := 'ROTVELB3'
INPUT.NODE.2                          := 'THETA1D'
INPUT.NODE.3                          := 'NONE'
COEFFICIENT.1                         := '+'
COEFFICIENT.2                         := '-'
COEFFICIENT.3                         := '+'

UP
CREATE SUMMER
NAME                                 := 'UTOT2'
OUTPUT.NODE                           := 'UT2'
INPUT.NODE.1                          := 'U2'
INPUT.NODE.2                          := 'U3'
INPUT.NODE.3                          := 'U6'
COEFFICIENT.1                         := '-'
COEFFICIENT.2                         := '-'
COEFFICIENT.3                         := '-'

UP
CREATE SUMMER
NAME                                 := 'UTOT1'
OUTPUT.NODE                           := 'UT1'
INPUT.NODE.1                          := 'U4'
INPUT.NODE.2                          := 'U5'
INPUT.NODE.3                          := 'U1'
COEFFICIENT.1                         := '-'
COEFFICIENT.2                         := '-'
COEFFICIENT.3                         := '-'

UP
CREATE SUMMER
NAME                                 := 'UTOT'
OUTPUT.NODE                           := 'FU'
INPUT.NODE.1                          := 'UT1'
INPUT.NODE.2                          := 'UT2'
INPUT.NODE.3                          := 'NONE'
COEFFICIENT.1                         := '+'
COEFFICIENT.2                         := '+'
COEFFICIENT.3                         := '+'

UP
CREATE SUMMER
NAME                                 := 'INT.TH1'
INPUT.NODE                          := 'THETA1D'
OUTPUT.NODE                          := 'THETA1'
S.ZERO                               := '1.3963'

NAME                               := 'SUMCORR'
OUTPUT.NODE                          := 'DELTATH1'
INPUT.NODE.1                         := 'THETA1'
INPUT.NODE.2                         := 'TH1CORR'
INPUT.NODE.3                         := 'NONE'
COEFFICIENT.1                       := '+'
COEFFICIENT.2                       := '-'
COEFFICIENT.3                       := '+'

UP
UP
CREATE BODY
NAME                                 := 'B0'
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.E0                               := 'POSITIVE'
ANGULAR.UNITS                         := 'RADIAN'
FLEXIBLE                              := 'FALSE'
SUPERELEMENT                          := 'FALSE'

UP
CREATE BODY
NAME                                 := 'B1'
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                       := 'FALSE'
MASS                                   := '0.2'
INERTIA.XXL                           := '0.00001'
INERTIA.YYL                           := '0.00001'
INERTIA.ZZL                           := '0.00001'
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.E0        := 'POSITIVE'
ANGULAR.UNITS  := 'RADIANS'
FLEXIBLE       := 'FALSE'
SUPERELEMENT   := 'FALSE'

UP
CREATE BODY
NAME           := 'B2'
CENTER.OF.GRAVITY := ( 0.04341, 0.2462, 0.0 )
TYPE.ANGULAR.COORD := 'EULER'
ANGLE.1        := '1.39626'
ANGLE.2        := '0'
ANGLE.3        := '0'
FIXED.TO.GROUND := 'FALSE'
MASS           := '0.01'
INERTIA.XXL    := '0.000000001'
INERTIA.YYL    := '0.000000001'
INERTIA.ZZL    := '0.20833E-03'
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.E0        := 'POSITIVE'
ANGULAR.UNITS  := 'RADIANS'
FLEXIBLE       := 'FALSE'
SUPERELEMENT   := 'FALSE'

UP
CREATE BODY
NAME           := 'B3'
CENTER.OF.GRAVITY := ( 0.1476, 0.8371, 0.0 )
TYPE.ANGULAR.COORD := 'EULER'
ANGLE.1        := '1.39626'
ANGLE.2        := '0.0'
ANGLE.3        := '0.0'
FIXED.TO.GROUND := 'FALSE'
MASS           := '0.01'
INERTIA.XXL    := '0.000000001'
INERTIA.YYL    := '0.000000001'
INERTIA.ZZL    := '0.408E-03'
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.E0        := 'POSITIVE'
ANGULAR.UNITS  := 'RADIANS'
FLEXIBLE       := 'FALSE'
SUPERELEMENT   := 'FALSE'

UP
CREATE INITIAL.CONDITION
NAME           := 'INITTRANS'
BODY.1.NAME    := 'B0'
BODY.2.NAME    := 'B1'
ELEMENT.NAME    := 'NONE'
TYPE.INITIAL.COND := 'X.DIFF'
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'

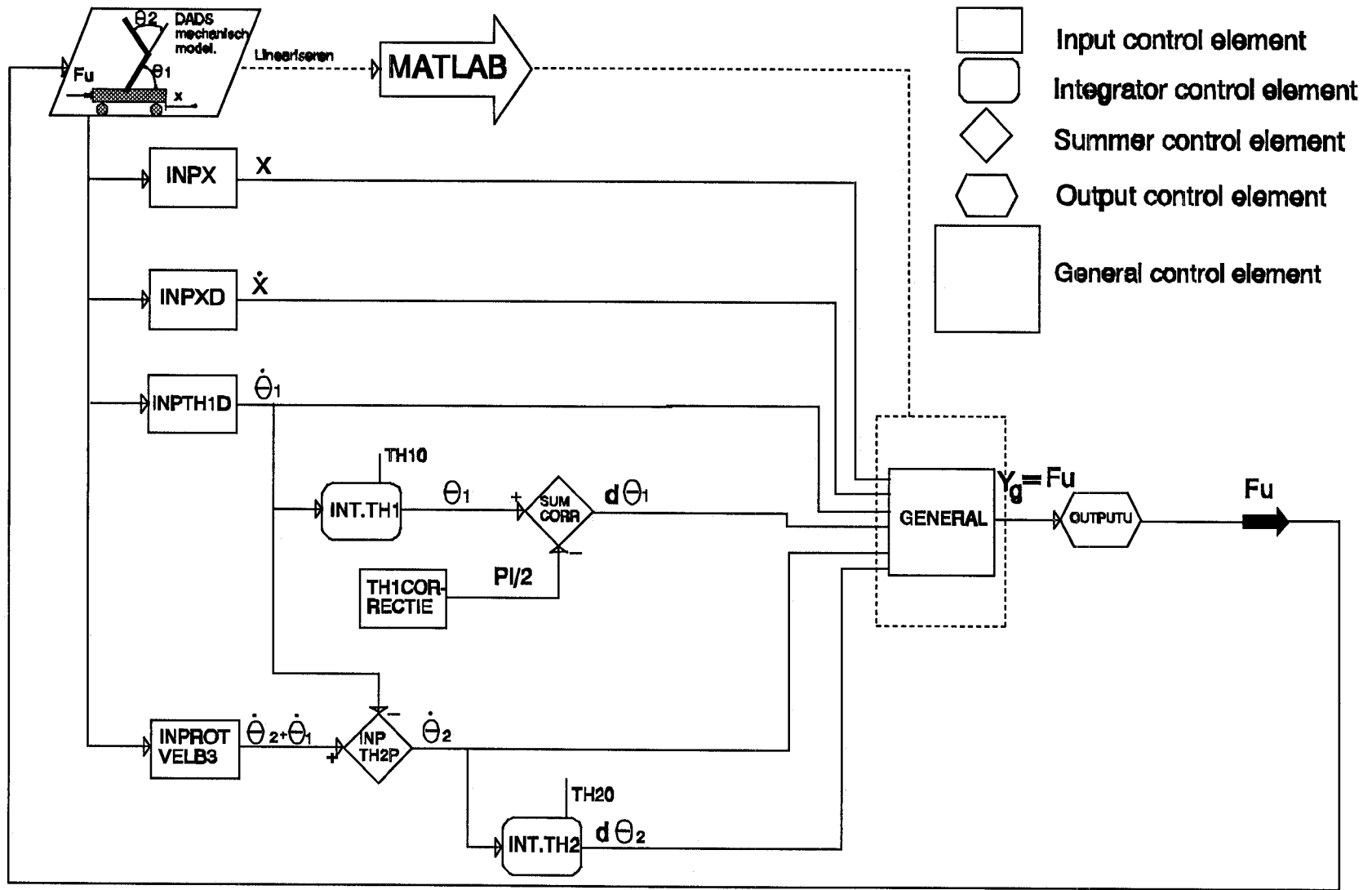
UP
CREATE INITIAL.CONDITION
NAME           := 'INIT1'
BODY.1.NAME    := 'NONE'
BODY.2.NAME    := 'NONE'
ELEMENT.NAME    := 'RSDA1'
TYPE.INITIAL.COND := 'REL.ANGLE'
INITIAL.VALUE   := '1.3963'
TIME.DERIVATIVE := '0.0'
OMEGA.Y        := '0.0'
OMEGA.Z        := '0.0'
P.ON.BODY.1    := ( 0.0, 0.0, 0.0 )
P.ON.BODY.2    := ( 0.0, 0.0, 0.0 )
EXTRA.COORD    := '0'
ANGULAR.UNITS  := 'RADIANS'






UP
CREATE INITIAL.CONDITION
NAME           := 'INIT2'
BODY.1.NAME    := 'NONE'
BODY.2.NAME    := 'NONE'
ELEMENT.NAME    := 'RSDA2'
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

```

BIJLAGE.10: SCHEMA REGELING DUBBELE SLINGER MET GENERAL CONTROL ELEMENT.



-  Input control element
-  Integrator control element
-  Summer control element
-  Output control element
-  General control element

Bijlage.11

VERBOSE FILE GEREGEELDE DUBBELE SLINGER.

De regeling is met behulp van een general control element toegepast.

```
ANALYSIS
CREATE SYSTEM.DATA
UNITS                := 'SI'
ANALYSIS.TYPE        := 'DYNAMIC'
STARTING.TIME        := '0.0'
ENDING.TIME          := '6.0'
PRINT.INTERVAL       := '0.05'
GRAVITY.SEA.LEVEL    := '9.80665'
X.GRAVITY            := '0.0'
Y.GRAVITY            := '-1.0'
Z.GRAVITY            := '0.0'
SCALE.GRAVITY.COEF   := '1.0'
MATRIX.OPERATIONS   := 'SPARSE'
REDUNDANCY.CHECK     := 'TRUE'
LUTOL                := '1.0D-12'
ASSEMBLY.TOL         := '1.0D-3'
BYPASS.ASSEMBLY      := 'FALSE'
OUTPUT.FILE          := 'ALL'
REFERENCE.FRAME      := 'LOCAL'
DEBUG.FLAG           := 'TRUE'

UP
CREATE DYNAMIC.DATA
REACTION.FORCES      := 'FALSE'
FORCE.COORDINATES    := 'BODY'
PRINT.METHOD        := 'INTERPOLATED'
MAX.INT.STEP         := '0.05'
SOLUTION.TOL         := '0.005'
INTEGRATION.TOL      := '0.001'
METHOD.INTEGRATION   := 'VARIABLE'
PRINT.FREQ           := '0'

UP
UP
FORCE
CREATE RSDA
NAME                 := 'RSDA1'
JOINT.NAME           := 'REV1'
ORIENTATION.ANGLE    := '0.0'
SPRING.CONSTANT      := '0.0'
DAMPING.COEFFICIENT := '0.0'
ACTUATOR.TORQUE      := '0.0'
CURVE.SPRING         := 'NONE'
CURVE.DAMPER         := 'NONE'
CURVE.ACTUATOR       := 'NONE'
ANGULAR.UNITS        := 'RADIANS'
TYPE                 := 'BIDIRECTIONAL'

UP
CREATE RSDA
NAME                 := 'RSDA2'
JOINT.NAME           := 'REV2'
ORIENTATION.ANGLE    := '0.0'
SPRING.CONSTANT      := '0.0'
DAMPING.COEFFICIENT := '0.0'
ACTUATOR.TORQUE      := '0.0'
CURVE.SPRING         := 'NONE'
CURVE.DAMPER         := 'NONE'
CURVE.ACTUATOR       := 'NONE'
ANGULAR.UNITS        := 'RADIANS'
TYPE                 := 'BIDIRECTIONAL'

UP
UP
JOINTS
CREATE REVOLUTE.JOINT
NAME                 := 'REV1'
BODY.1.NAME          := 'B1'

BODY.2.NAME          := 'B2'
P.ON.BODY.1          := ( 0.0, 0.0, 0.0 )
P.ON.BODY.2          := ( -0.25, 0.0, 0.0 )
Q.ON.BODY.1          := ( 0.0, 0.0, 1.0 )
Q.ON.BODY.2          := ( -0.25, 0.0, 1.0 )
R.ON.BODY.1          := ( 1.0, 0.0, 0.0 )
R.ON.BODY.2          := ( 0.75, 0.0, 0.0 )
NODE.1               := '0'
NODE.2               := '0'

UP
CREATE REVOLUTE.JOINT
NAME                 := 'REV2'
BODY.1.NAME          := 'B2'
BODY.2.NAME          := 'B3'
P.ON.BODY.1          := ( 0.25, 0.0, 0.0 )
P.ON.BODY.2          := ( -0.35, 0.0, 0.0 )
Q.ON.BODY.1          := ( 0.25, 0.0, 1.0 )
Q.ON.BODY.2          := ( -0.35, 0.0, 1.0 )
R.ON.BODY.1          := ( 1.25, 0.0, 0.0 )
R.ON.BODY.2          := ( 0.65, 0.0, 0.0 )
NODE.1               := '0'
NODE.2               := '0'

UP
CREATE TRANSLATIONAL.JOINT
NAME                 := 'TRANSL'
BODY.1.NAME          := 'B0'
BODY.2.NAME          := 'B1'
P.ON.BODY.1          := ( 0.0, 0.0, 0.0 )
P.ON.BODY.2          := ( 0.0, 0.0, 0.0 )
Q.ON.BODY.1          := ( 1.0, 0.0, 0.0 )
Q.ON.BODY.2          := ( 1.0, 0.0, 0.0 )
R.ON.BODY.1          := ( 0.0, 1.0, 0.0 )
R.ON.BODY.2          := ( 0.0, 1.0, 0.0 )
NODE.1               := '0'
NODE.2               := '0'

UP
UP
CONTROLS
CREATE INPUT.FUNCTION
NAME                 := 'INPX'
NODE.NAME            := 'X'
TYPE                 := 'X'
BODY.1.NAME          := 'B1'
BODY.2.NAME          := 'NONE'
FUNCTION.PARAMETERS := ( 0.0, 0.0, 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          := ( 1.0, 0.0, 0.0 )
Q.ON.BODY.2          := ( 1.0, 0.0, 0.0 )
CURVE.NAME           := 'NONE'
JOINT.NAME           := 'NONE'
FLEXIBLE.NODE.1      := '0'
FLEXIBLE.NODE.2      := '0'
ANGULAR.UNITS        := 'RADIANS'
EXTNUM               := '0'
SAMPLE.RATE          := '0.0'
DIGITAL.FLAG         := 'FALSE'
LINEAR.FLAG          := 'FALSE'

UP
CREATE INPUT.FUNCTION
NAME                 := 'INPKD'
NODE.NAME            := 'XD'
TYPE                 := 'XD'
BODY.1.NAME          := 'B1'
```

```

BODY.2.NAME                := 'NONE'
FUNCTION.PARAMETERS        := ( 0.0, 0.0, 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                := ( 1.0, 0.0, 0.0 )
Q.ON.BODY.2                := ( 1.0, 0.0, 0.0 )
CURVE.NAME                 := 'NONE'
JOINT.NAME                 := 'NONE'
FLEXIBLE.NODE.1            := '0'
FLEXIBLE.NODE.2            := '0'
ANGULAR.UNITS              := 'RADIANS'
EXTNUM                     := '0'
SAMPLE.RATE                := '0.0'
DIGITAL.FLAG               := 'FALSE'
LINEAR.FLAG                := 'FALSE'
UP
CREATE INPUT.FUNCTION
NAME                       := 'INPTH1D'
NODE.NAME                  := 'THETA1D'
TYPE                       := 'ZL.OMEGA'
BODY.1.NAME                := 'B2'
BODY.2.NAME                := 'NONE'
FUNCTION.PARAMETERS        := ( 0.0, 0.0, 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                := ( 1.0, 0.0, 0.0 )
Q.ON.BODY.2                := ( 1.0, 0.0, 0.0 )
CURVE.NAME                 := 'NONE'
JOINT.NAME                 := 'NONE'
FLEXIBLE.NODE.1            := '0'
FLEXIBLE.NODE.2            := '0'
ANGULAR.UNITS              := 'RADIANS'
EXTNUM                     := '0'
SAMPLE.RATE                := '0.0'
DIGITAL.FLAG               := 'FALSE'
LINEAR.FLAG                := 'FALSE'
UP
CREATE INPUT.FUNCTION
NAME                       := 'INPROTVELB3'
NODE.NAME                  := 'ROTVELB3'
TYPE                       := 'ZL.OMEGA'
BODY.1.NAME                := 'B3'
BODY.2.NAME                := 'NONE'
FUNCTION.PARAMETERS        := ( 0.0, 0.0, 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                := ( 1.0, 0.0, 0.0 )
Q.ON.BODY.2                := ( 1.0, 0.0, 0.0 )
CURVE.NAME                 := 'NONE'
JOINT.NAME                 := 'NONE'
FLEXIBLE.NODE.1            := '0'
FLEXIBLE.NODE.2            := '0'
ANGULAR.UNITS              := 'RADIANS'
EXTNUM                     := '0'
SAMPLE.RATE                := '0.0'
DIGITAL.FLAG               := 'FALSE'
LINEAR.FLAG                := 'FALSE'
UP
CREATE INPUT.FUNCTION
NAME                       := 'TH1CORRECTIE'
NODE.NAME                  := 'TH1CORR'
TYPE                       := 'POLYNOMIAL'
BODY.1.NAME                := 'NONE'
BODY.2.NAME                := 'NONE'
FUNCTION.PARAMETERS        := ( 1.570796327, 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                := ( 1.0, 0.0, 0.0 )
Q.ON.BODY.2                := ( 1.0, 0.0, 0.0 )
CURVE.NAME                 := 'NONE'
JOINT.NAME                 := 'NONE'
FLEXIBLE.NODE.1            := '0'
FLEXIBLE.NODE.2            := '0'
ANGULAR.UNITS              := 'DEGREES'
EXTNUM                     := '0'
SAMPLE.RATE                := '0.0'
DIGITAL.FLAG               := 'FALSE'
LINEAR.FLAG                := 'FALSE'
UP
CREATE INTEGRATOR
NAME                       := 'INT.TH1'
INPUT.NODE                 := 'THETA1D'
OUTPUT.NODE                := 'THETA1'
S.ZERO                     := '1.3963'
UP
CREATE INTEGRATOR
NAME                       := 'INT.TH2'
INPUT.NODE                 := 'THETA2D'
OUTPUT.NODE                := 'THETA2'
S.ZERO                     := '0.0'
UP
CREATE OUTPUT
NAME                       := 'OUTPUTU'
OUTPUT.NODE                := 'FU'
TYPE                       := 'X.FORCE'
BODY.1.NAME                := 'B1'
BODY.2.NAME                := 'NONE'
P.ON.BODY.1                := ( 0.0, 0.0, 0.0 )
P.ON.BODY.2                := ( 0.0, 0.0, 0.0 )
JOINT.NAME                 := 'NONE'
FLEXIBLE.NODE.1            := '0'
FLEXIBLE.NODE.2            := '0'
UP
CREATE SUMMER
NAME                       := 'INPTH2D'
OUTPUT.NODE                := 'THETA2D'
INPUT.NODE.1               := 'ROTVELB3'
INPUT.NODE.2               := 'THETA1D'
INPUT.NODE.3               := 'NONE'
COEFFICIENT.1              := '+'
COEFFICIENT.2              := '-'
COEFFICIENT.3              := '+'
UP
CREATE SUMMER
NAME                       := 'SUMCORR'
OUTPUT.NODE                := 'DELTATH1'
INPUT.NODE.1               := 'THETA1'
INPUT.NODE.2               := 'TH1CORR'
INPUT.NODE.3               := 'NONE'
COEFFICIENT.1              := '+'
COEFFICIENT.2              := '-'
COEFFICIENT.3              := '+'
UP
CREATE GENERAL
NAME                       := 'GENERAL'
NDIM                       := '1'
MDIM                       := '6'
LDIM                       := '1'

```

| | | | |
|--------------------|----------------------|--------------------|-----------------------------|
| IFLAG | := 'FALSE' | CURVE.YLT | := 'NONE' |
| AFLAG | := 'FALSE' | CURVE.ZLT | := 'NONE' |
| BFLAG | := 'FALSE' | SIGN.EO | := 'POSITIVE' |
| CFLAG | := 'FALSE' | ANGULAR.UNITS | := 'RADIANS' |
| DFLAG | := 'FALSE' | FLEXIBLE | := 'FALSE' |
| UP | | SUPERELEMENT | := 'FALSE' |
| UP | | UP | |
| CREATE BODY | | CREATE BODY | |
| NAME | := 'B0' | NAME | := 'B2' |
| CENTER.OF.GRAVITY | := (0.0, 0.0, 0.0) | CENTER.OF.GRAVITY | := (0.04341, 0.2462, 0.0) |
| TYPE.ANGULAR.COORD | := 'EULER' | TYPE.ANGULAR.COORD | := 'EULER' |
| ANGLE.1 | := '0.0' | ANGLE.1 | := '1.39626' |
| ANGLE.2 | := '0.0' | ANGLE.2 | := '0' |
| ANGLE.3 | := '0.0' | ANGLE.3 | := '0' |
| FIXED.TO.GROUND | := 'TRUE' | FIXED.TO.GROUND | := 'FALSE' |
| MASS | := '1.0' | MASS | := '0.01' |
| INERTIA.XXL | := '1.0' | INERTIA.XXL | := '0.000000001' |
| INERTIA.YYL | := '1.0' | INERTIA.YYL | := '0.000000001' |
| INERTIA.ZZL | := '1.0' | INERTIA.ZZL | := '0.20833E-03' |
| INERTIA.XYL | := '0.0' | INERTIA.XYL | := '0.0' |
| INERTIA.XZL | := '0.0' | INERTIA.XZL | := '0.0' |
| INERTIA.YZL | := '0.0' | INERTIA.YZL | := '0.0' |
| XG.FORCE | := '0.0' | XG.FORCE | := '0.0' |
| YG.FORCE | := '0.0' | YG.FORCE | := '0.0' |
| ZG.FORCE | := '0.0' | ZG.FORCE | := '0.0' |
| XL.TORQUE | := '0.0' | XL.TORQUE | := '0.0' |
| YL.TORQUE | := '0.0' | YL.TORQUE | := '0.0' |
| ZL.TORQUE | := '0.0' | ZL.TORQUE | := '0.0' |
| CURVE.XGF | := 'NONE' | CURVE.XGF | := 'NONE' |
| CURVE.YGF | := 'NONE' | CURVE.YGF | := 'NONE' |
| CURVE.ZGF | := 'NONE' | CURVE.ZGF | := 'NONE' |
| CURVE.XLT | := 'NONE' | CURVE.XLT | := 'NONE' |
| CURVE.YLT | := 'NONE' | CURVE.YLT | := 'NONE' |
| CURVE.ZLT | := 'NONE' | CURVE.ZLT | := 'NONE' |
| SIGN.EO | := 'POSITIVE' | SIGN.EO | := 'POSITIVE' |
| ANGULAR.UNITS | := 'RADIANS' | ANGULAR.UNITS | := 'RADIANS' |
| FLEXIBLE | := 'FALSE' | FLEXIBLE | := 'FALSE' |
| SUPERELEMENT | := 'FALSE' | SUPERELEMENT | := 'FALSE' |
| UP | | UP | |
| CREATE BODY | | CREATE BODY | |
| NAME | := 'B1' | NAME | := 'B3' |
| CENTER.OF.GRAVITY | := (0.0, 0.0, 0.0) | CENTER.OF.GRAVITY | := (0.1476, 0.8371, 0.0) |
| TYPE.ANGULAR.COORD | := 'EULER' | TYPE.ANGULAR.COORD | := 'EULER' |
| ANGLE.1 | := '0.0' | ANGLE.1 | := '1.39626' |
| ANGLE.2 | := '0.0' | ANGLE.2 | := '0.0' |
| ANGLE.3 | := '0.0' | ANGLE.3 | := '0.0' |
| FIXED.TO.GROUND | := 'FALSE' | FIXED.TO.GROUND | := 'FALSE' |
| MASS | := '0.2' | MASS | := '0.01' |
| INERTIA.XXL | := '0.00001' | INERTIA.XXL | := '0.000000001' |
| INERTIA.YYL | := '0.00001' | INERTIA.YYL | := '0.000000001' |
| INERTIA.ZZL | := '0.00001' | INERTIA.ZZL | := '0.408E-03' |
| INERTIA.XYL | := '0.0' | INERTIA.XYL | := '0.0' |
| INERTIA.XZL | := '0.0' | INERTIA.XZL | := '0.0' |
| INERTIA.YZL | := '0.0' | INERTIA.YZL | := '0.0' |
| XG.FORCE | := '0.0' | XG.FORCE | := '0.0' |
| YG.FORCE | := '0.0' | YG.FORCE | := '0.0' |
| ZG.FORCE | := '0.0' | ZG.FORCE | := '0.0' |
| XL.TORQUE | := '0.0' | XL.TORQUE | := '0.0' |
| YL.TORQUE | := '0.0' | YL.TORQUE | := '0.0' |
| ZL.TORQUE | := '0.0' | ZL.TORQUE | := '0.0' |
| CURVE.XGF | := 'NONE' | CURVE.XGF | := 'NONE' |
| CURVE.YGF | := 'NONE' | CURVE.YGF | := 'NONE' |
| CURVE.ZGF | := 'NONE' | CURVE.ZGF | := 'NONE' |
| CURVE.XLT | := 'NONE' | CURVE.XLT | := 'NONE' |


```

CURVE.YLT := 'NONE'
CURVE.ZLT := 'NONE'
SIGN.EO := 'POSITIVE'
ANGULAR.UNITS := 'RADIANS'
FLEXIBLE := 'FALSE'
SUPERELEMENT := 'FALSE'
UP
CREATE INITIAL.CONDITION
NAME := 'INITTRANS'
BODY.1.NAME := 'B0'
BODY.2.NAME := 'B1'
ELEMENT.NAME := 'NONE'
TYPE.INITIAL.COND := 'X.DIFF'
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 := 'INIT1'
BODY.1.NAME := 'NONE'
BODY.2.NAME := 'NONE'
ELEMENT.NAME := 'RSDA1'
TYPE.INITIAL.COND := 'REL.ANGLE'
INITIAL.VALUE := '1.3963'
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 := 'INIT2'
BODY.1.NAME := 'NONE'
BODY.2.NAME := 'NONE'
ELEMENT.NAME := 'RSDA2'
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

```



```

CENTER.OF.GRAVITY := ( 0.0, 0.0 )
PHI := '0.0'
FIXED.TO.GROUND := 'FALSE'
MASS := '0.2'
INERTIA := '0.00001'
XG.FORCE := '0.0'
YG.FORCE := '0.0'
TORQUE.CONSTANT := '0.0'
CURVE.XGF := 'NONE'
CURVE.YGF := 'NONE'
CURVE.TORQUE := 'NONE'
OUTLINE.SHAPE := 'NONE'
SHAPE.CENTER := ( 0.0, 0.0 )
ANGULAR.UNITS := 'DEGREES'
FLEXIBLE := 'FALSE'
SUPERELEMENT := 'FALSE'

UP
CREATE BODY
NAME := 'B2'
CENTER.OF.GRAVITY := ( 0.0, 0.15 )
PHI := '90'
FIXED.TO.GROUND := 'FALSE'
MASS := '0.0667557'
INERTIA := '0.000500668'
XG.FORCE := '0.0'
YG.FORCE := '0.0'
TORQUE.CONSTANT := '0.0'
CURVE.XGF := 'NONE'
CURVE.YGF := 'NONE'
CURVE.TORQUE := 'NONE'
OUTLINE.SHAPE := 'NONE'
SHAPE.CENTER := ( 0.0, 0.0 )
ANGULAR.UNITS := 'DEGREES'
FLEXIBLE := 'TRUE'
SUPERELEMENT := 'FALSE'

UP
CREATE BODY
NAME := 'B3'
CENTER.OF.GRAVITY := ( 0.0, 0.45 )
PHI := '90'
FIXED.TO.GROUND := 'FALSE'
MASS := '0.0667557'
INERTIA := '0.000500668'
XG.FORCE := '0.0'
YG.FORCE := '0.0'
TORQUE.CONSTANT := '0.0'
CURVE.XGF := 'NONE'
CURVE.YGF := 'NONE'
CURVE.TORQUE := 'NONE'
OUTLINE.SHAPE := 'NONE'
SHAPE.CENTER := ( 0.0, 0.0 )
ANGULAR.UNITS := 'DEGREES'
FLEXIBLE := 'TRUE'
SUPERELEMENT := 'FALSE'

UP
CREATE FLEXIBLE
NAME := 'FLEX3'
BODY.NAME := 'B3'
FILE.NAME := 'flex.file'
NUMBER.MODES := '1'
FORCE.NODE := '0'
XG.FORCE := '0.0'
YG.FORCE := '0.0'
RELATIVE.DAMPING := '0.0'

UP
CREATE FLEXIBLE
NAME := 'FLEX2'
BODY.NAME := 'B2'
FILE.NAME := 'flex.file'
NUMBER.MODES := '1'
FORCE.NODE := '0'
XG.FORCE := '0.0'
YG.FORCE := '0.0'
RELATIVE.DAMPING := '0.0'

UP
CREATE INITIAL.CONDITION
NAME := 'INIT1'
BODY.1.NAME := 'B1'
BODY.2.NAME := 'NONE'
TYPE.INITIAL.COND := 'X'
INITIAL.VALUE := '0.0'
TIME.DERIVATIVE := '0.0'
P.ON.BODY.1 := ( 0.0, 0.0 )
P.ON.BODY.2 := ( 0.0, 0.0 )
EXTRA.COORD := '0'
ANGULAR.UNITS := 'DEGREES'

UP
CREATE INITIAL.CONDITION
NAME := 'INIT2'
BODY.1.NAME := 'B2'
BODY.2.NAME := 'NONE'
TYPE.INITIAL.COND := 'Y'
INITIAL.VALUE := '0.15'
TIME.DERIVATIVE := '0.0'
P.ON.BODY.1 := ( 0.0, 0.0 )
P.ON.BODY.2 := ( 0.0, 0.0 )
EXTRA.COORD := '0'
ANGULAR.UNITS := 'DEGREES'

UP
CREATE INITIAL.CONDITION
NAME := 'INIT3'
BODY.1.NAME := 'B3'
BODY.2.NAME := 'NONE'
TYPE.INITIAL.COND := 'Y'
INITIAL.VALUE := '0.45'
TIME.DERIVATIVE := '0.0'
P.ON.BODY.1 := ( 0.0, 0.0 )
P.ON.BODY.2 := ( 0.0, 0.0 )
EXTRA.COORD := '0'
ANGULAR.UNITS := 'DEGREES'

UP
CREATE INITIAL.CONDITION
NAME := 'INITF3'
BODY.1.NAME := 'B3'
BODY.2.NAME := 'NONE'
TYPE.INITIAL.COND := 'MODE.SHAPE'
INITIAL.VALUE := '0.0'
TIME.DERIVATIVE := '0.0'
P.ON.BODY.1 := ( 0.0, 0.0 )
P.ON.BODY.2 := ( 0.0, 0.0 )
EXTRA.COORD := '1'
ANGULAR.UNITS := 'DEGREES'

UP
CREATE INITIAL.CONDITION
NAME := 'INITF2'
BODY.1.NAME := 'B2'
BODY.2.NAME := 'NONE'
TYPE.INITIAL.COND := 'MODE.SHAPE'
INITIAL.VALUE := '0.0'
TIME.DERIVATIVE := '0.0'
P.ON.BODY.1 := ( 0.0, 0.0 )
P.ON.BODY.2 := ( 0.0, 0.0 )
EXTRA.COORD := '1'
ANGULAR.UNITS := 'DEGREES'

UP

```

Bijlage.13

PC-MATLAB PROGRAMMA'S VOOR DE ENKELE SLINGER.

% Programma esling.m regelt initialisatie en aanroep ode23.

```
clear
t0=0; %begintijdstip
tf=6.0; %eindtijdstip
x0=[0.00000 2.0 0.000000 0.00000]'; %initial conditions
[t,x] = ode23('eslvgl',t0,tf,x0); %aanroep ode23
subplot(211),plot(t,x(:,1))
title('Simulatie enkele slinger in Matlab.')
xlabel('-> t')
ylabel('-> x'),pause
subplot(212),plot(t,x(:,2))
xlabel('-> t')
ylabel('-> theta')
```

% Programma eslvgl.m met niet lineaire toestandsvergelijkin-
% gen die in ode23 geïntegreerd worden.

```
function xdot = eslvgl(t,x)
m1=0.2;
m2=0.01;
g=9.81;
l2=0.5;
u=x(1)-10.1686*(x(2)-pi/2)+1.6291*x(3)-2.1047*x(4); %ingang
xdotdot=(u-0.75*m2*g*cos(x(2))*sin(x(2))+0.5*m2*l2*cos(x(2))*
(x(4)^2))/(m1+m2-0.75*m2*(sin(x(2))^2));
xdot(1)=x(3);
xdot(2)=x(4);
xdot(3)=xdotdot;
xdot(4)=3*(xdotdot*sin(x(2))-g*cos(x(2)))/(2*l2);
```