

## Dynamische analyse van een klepmechanisme

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

Meulen, van der, F. (1994). *Dynamische analyse van een klepmechanisme*. (DCT rapporten; Vol. 1994.149). Technische Universiteit Eindhoven.

**Document status and date:**

Gepubliceerd: 01/01/1994

**Document Version:**

Uitgevers PDF, ook bekend als Version of Record

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

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

[Link to publication](#)

**General rights**

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

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

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

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

**Take down policy**

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

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

providing details and we will investigate your claim.

Dynamische Analyse  
van een  
Klepmechanisme

Frits van der Meulen

14 november 1994

# Dynamische Analyse van een Klepmechanisme

Frits van der Meulen

Stageverslag over de periode  
juni 1994 tot november 1994

In opdracht van  
ir. M. Dhaenens en dr.ir. J.G.A.M.v. Heck (DAF-Trucks)

Onder begeleiding van  
dr.ir. A.A.H.J. Sauren (TUE)  
ir. H.J. Giesen (TUE)  
ir. M. Dhaenens (DAF-Trucks)  
dr.ir. J.G.A.M.v. Heck (DAF-Trucks)

WFW-rapport 94.149

Technische Universiteit Eindhoven  
vakgroep Fundamentele Werktuigkunde

# Inhoudsopgave

---

<b>1</b>	<b>Inleiding</b>	<b>1</b>
<b>2</b>	<b>Het klepmechanisme</b>	<b>3</b>
2.1	Functie van het klepmechanisme . . . . .	3
2.2	Klepmechanisme met onderliggende nokkenas . . . . .	6
2.2.1	Het huidige ontwerp . . . . .	6
2.2.2	Het ontwerp met de inverse dynamische analyse . . . . .	6
<b>3</b>	<b>Simulatie modellen in de literatuur</b>	<b>11</b>
3.1	Het dynamische model . . . . .	11
<b>4</b>	<b>Starre model in DADS</b>	<b>13</b>
4.1	Afweging tussen modelleren in 2D of 3D . . . . .	13
4.2	De nok . . . . .	14
4.3	De tuimelaar . . . . .	14
4.3.1	Het contactelement . . . . .	15
4.4	De veer . . . . .	16
4.5	De klep . . . . .	17
4.6	Simulatie . . . . .	17
4.6.1	Kleppositie en snelheid bij variabele speling . . . . .	18
4.6.2	Variatie van contactdemping . . . . .	18
<b>5</b>	<b>DADS-model met flexibele klep</b>	<b>23</b>
5.1	De NASTRAN som . . . . .	23
5.2	IP en de problemen bij DAF . . . . .	23
5.3	Resultaten van de simulaties . . . . .	25
<b>6</b>	<b>Conclusies en aanbevelingen</b>	<b>31</b>
<b>A</b>	<b>Gebruikte parameters en instellingen</b>	<b>35</b>
<b>B</b>	<b>Resultaten simulaties met starre klep</b>	<b>36</b>
<b>C</b>	<b>Resultaten simulaties met flexibele klep</b>	<b>41</b>

<b>D Nastran headers</b>	<b>48</b>
D.1 Headers bij combinatie statische en normal modes . . . . .	48
D.1.1 Normal modes . . . . .	48
D.1.2 statische modes . . . . .	49
D.2 Alleen de normal modes meegenomen . . . . .	50
<b>E Dads definition file</b>	<b>53</b>

De dynamische simulatie is tegenwoordig een belangrijk hulpmiddel geworden voor de ontwikkeling van hoog belaste mechanismen. De rekencapaciteit die tegenwoordig voor handen is maakt het mogelijk om complexe mechanismen te simuleren wat tot een beter inzicht kan voeren. Geavanceerde software voor multi-body dynamica gecombineerd met software voor de eindige elementen methode lijken haast geen begrenzings te hebben.

Momenteel wordt bij DAF-Trucks N.V. de kleppentrein van de dieselmotoren alleen invers dynamisch doorgekend: Op grond van de voorgeschreven nokverplaatsing en de massa's van de diverse star veronderstelde componenten worden de krachten bepaald.

Beproevingresultaten lijken echter aan te geven dat er in de kleppentrein dynamische krachten optreden die aanzienlijk hoger zijn dan de berekende krachten. Om meer inzicht te krijgen in het dynamische gedrag van de kleppentrein is een dynamische analyse noodzakelijk.

In dit verslag zal worden beschreven hoe met behulp van het multi-body pakket DADS en het eindige elementen pakket MSC/NASTRAN een simulatiemodel is verkregen van het klepmechanisme. Eerst zullen de afzonderlijke onderdelen van het mechanisme star worden gemodelleerd met behulp van DADS. Vervolgens zal er flexibiliteit aan worden toegevoegd met behulp van het eindige elementen pakket MSC/NASTRAN.



In dit hoofdstuk zal gekeken worden naar de algemene werking van het klepmechanisme en de manier waarop dit bij DAF geïmplementeerd is.

## 2.1 Functie van het klepmechanisme

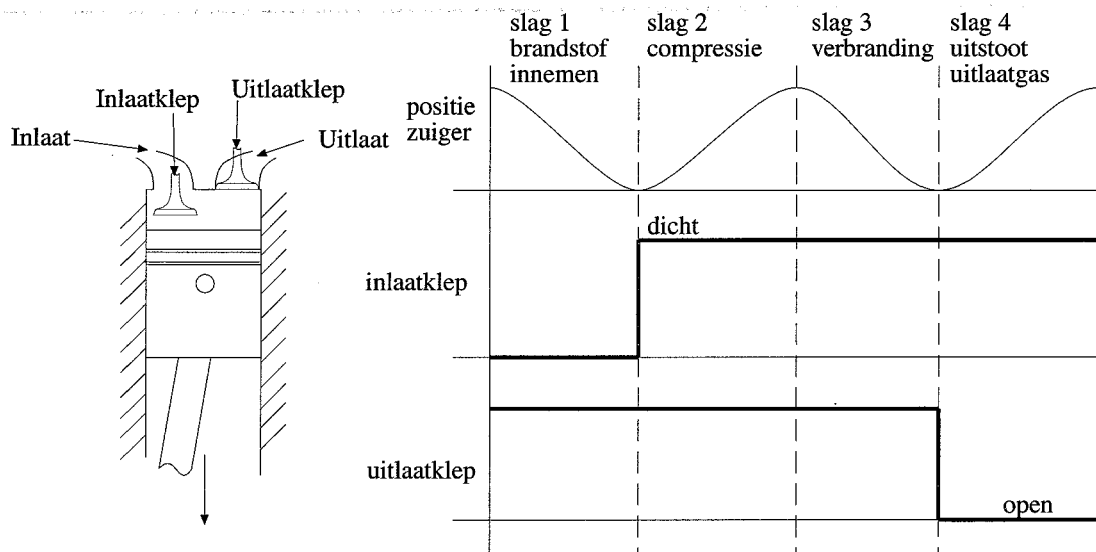
Binnen de vier-takt verbrandingsmotor speelt het klepmechanisme een belangrijke rol. Het zorgt ervoor dat op het juiste tijdstip het brandbare mengsel de cylinder in kan en de uitlaatgassen naar buiten kunnen, terwijl tijdens de compressie en de ontbranding de cylinder afgesloten is. Het op het juiste moment openen en sluiten van de kleppen, de klep-timing, is van essentieel belang voor het goed functioneren van de motor. Een onjuiste timing kan bijvoorbeeld tot een lager rendement leiden, maar ook in het ongunstigste geval tot schade aan de motor.

Heel globaal kan de klepbeweging worden beschreven met het plaats-tijd diagram zoals in figuur 2.1 staat. Hierin zijn de volgende standen te onderscheiden:

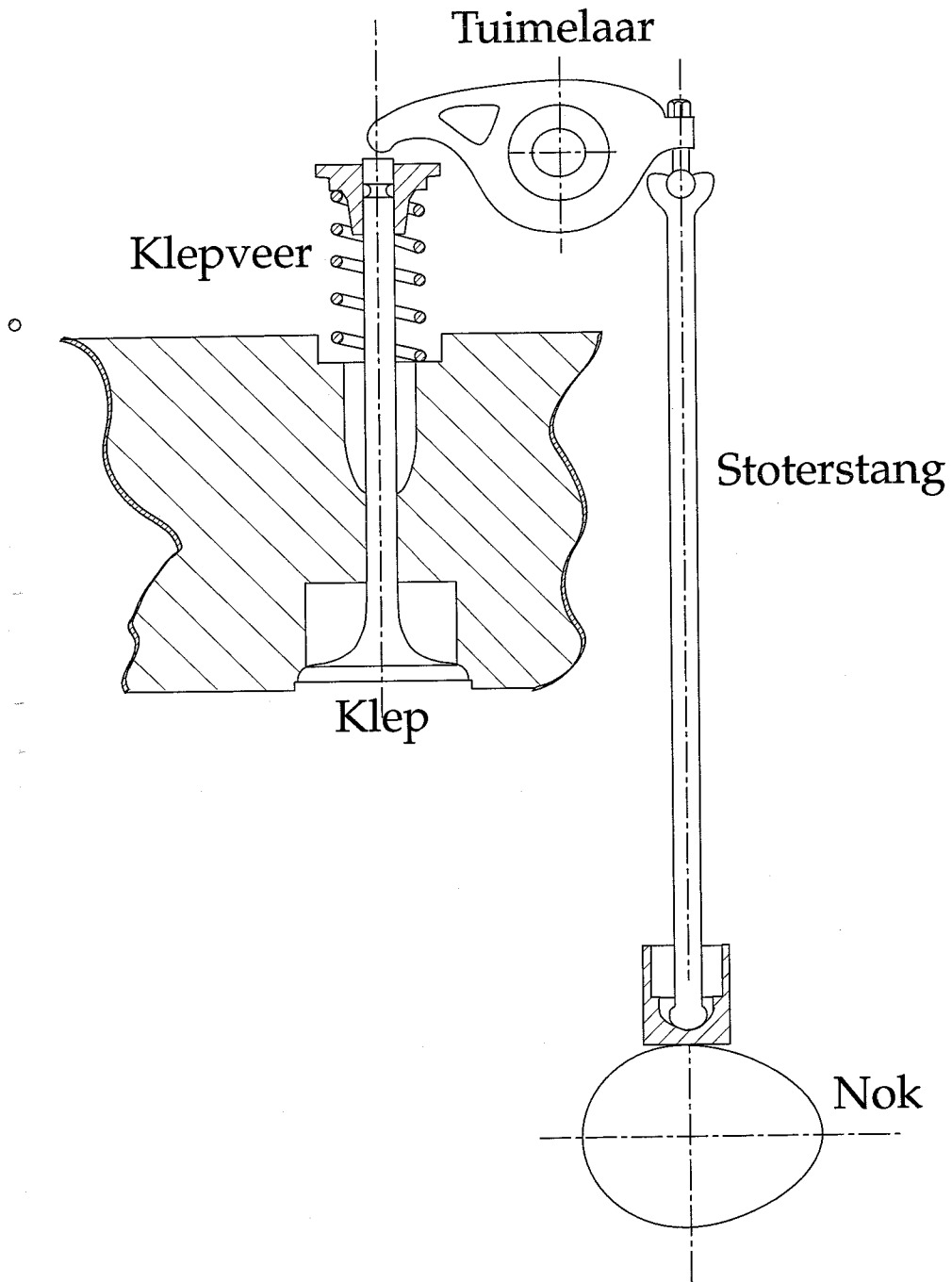
- Tijdens de eerste slag moet het brandbare mengsel vrij de cylinder in kunnen stromen: inlaatklep open.
- Tijdens de compressie en de verbranding moeten beide kleppen dicht zijn.
- Tijdens de laatste slag moeten de verbrandingsgassen de cylinder vrij kunnen verlaten: uitlaatklep open.

De beweging volgens het plaats-tijd diagram brengt echter oneindige versnellingen met zich mee. Natuurlijk is dit in de praktijk niet mogelijk en zal de beweging worden beperkt door de maximaal mogelijke versnellingen. Bij een lichter mechanisme zijn deze in het algemeen groter, immers de optredende belastingen zijn rechtevenredig met de massa.





Figuur 2.1: Theoretische klepstanden als functie van de zuigerbeweging



Figuur 2.2: Klepmechanisme

## 2.2 Klepmechanisme met onderliggende nokkenas

### 2.2.1 Het huidige ontwerp

Het huidige ontwerp van het klepmechanisme staat in figuur 2.2. Het concept van dit ontwerp met onderliggende nokkenas is al oud en werd al gebruikt in de motoren van de Trucks uit de jaren 50. In de loop van de jaren is dit ontwerp verder geoptimaliseerd, maar het concept is nooit veranderd.

In de automobiel-industrie wordt tegenwoordig dit concept niet of nauwelijks meer gebruikt. De bovenliggende nokkenas al dan niet met hydraulische klepstoters heeft het in de benzine motoren van personenwagens gewonnen van het oude concept. De oorzaak hiervan is dat de motoren binnen deze industrie sneller lopen en daarom meer eisen stellen aan het klepmechanisme. Het klinkt triviaal om dit concept ook over te nemen voor de dieselmotoren bij DAF, echter de problemen die ontstaan bij de implementatie zijn niet eenvoudig. Er zal hier verder niet al te veel op deze concepten worden ingegaan. Toch kan er veel geleerd worden van deze vooruitlopende industrie-tak.

Om tot een wat beter inzicht te komen van de werking van het huidige mechanisme zullen er enkele aspecten volgen die belangrijk zijn voor het ontwerp:

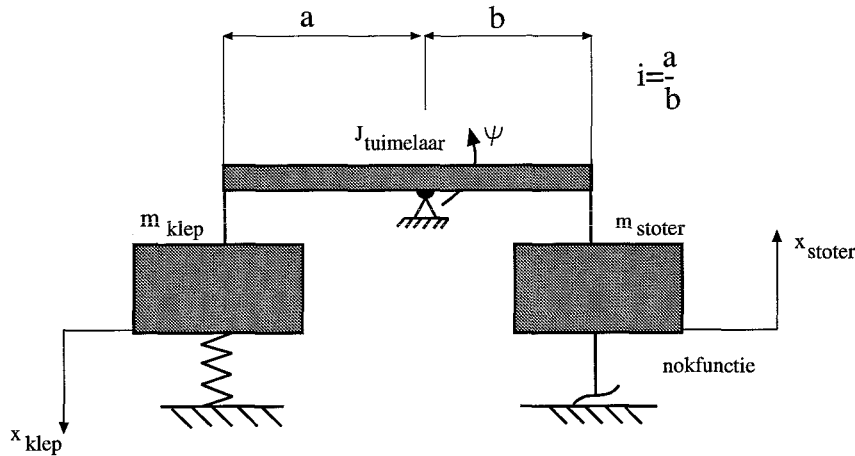
- Tussen de tuimelaar en de gesloten klep moet speling aanwezig zijn. Deze speling garandeert dat de klep sluit. Dit is noodzakelijk omdat de klep haar warmte via de zitting afstaat. Zou de klep niet sluiten dan verbrandt deze.
- Bij gesloten klep zorgt de klepveer voor het sluiten van de klep.
- Bij geopende klep zorgt de klepveer dat de stoterstang (bij vertragen van de opgaande beweging en versnellen van de neergaande beweging) op de nok blijft aanliggen, zodat klapperen van de stang op de nok vermeden wordt en de klep de nokfunctie volgt.

### 2.2.2 Het ontwerp met de inverse dynamische analyse

De nokfunctie legt als het ware een beweging op aan de verschillende onderdelen. Door de versnellingen van deze verschillende onderdelen in de bewegingsvergelijking in te vullen worden de krachten in het mechanisme verkregen. De verschillende onderdelen van het mechanisme worden hierbij star verondersteld. Verder zijn de onderdelen hierbij aan elkaar gekoppeld. Speling wordt dus bijvoorbeeld niet meegenomen. Het mechanisme kan dan worden gereduceerd tot het mechanisme in figuur 2.3.

Voor de gesloten klep geldt:

$$F_{zitting} = F_{veer} \tag{2.1}$$



Figuur 2.3: Simpele weergave van het systeem

Voor de geopende klep geldt, met als eis  $F_{N\text{ klep-tuimelaar}} > 0$ :

$$F_{N\text{ klep-tuimelaar}} = F_{veer} + m_{klep}\ddot{x}_{klep} \quad (2.2)$$

Voor een overbrengingsverhouding  $i=1$ , met als eis  $F_{nok} > 0$ :

$$F_{nok} = F_{N\text{ klep-tuimelaar}} + m_{stoter}\ddot{x}_{stoter} + J_{tuimelaar}\ddot{\psi} \quad (2.3)$$

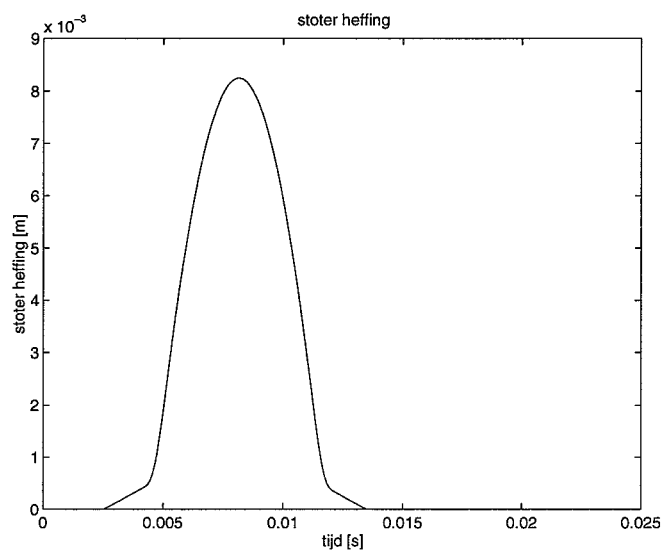
De nokfunctie zoals deze thans gebruikt wordt voor de inlaatklep ziet er uit zoals in figuur 2.4 tot en met 2.6.

De keuze van de juiste nokfunctie is erg ingewikkeld. Er zijn tal van factoren die van invloed zijn op een zo goed mogelijk compromis. Hiervan zijn er hieronder een aantal gegeven.

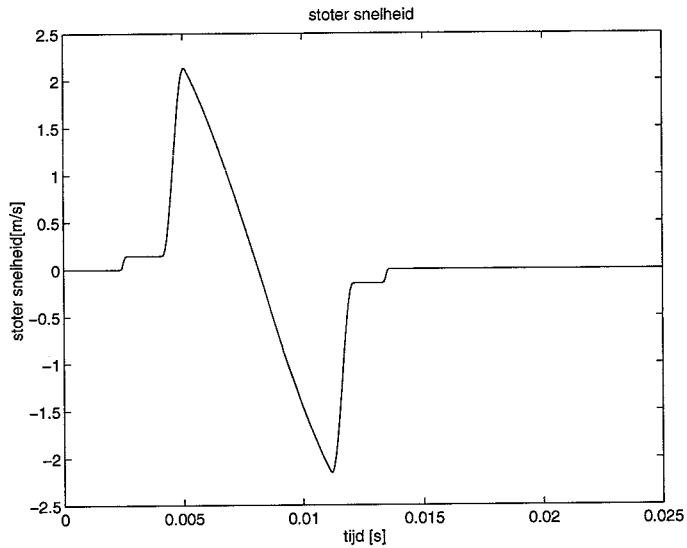
1. De nokfunctie begint en eindigt met een relatief lage constante snelheid om de klepspeling te doorlopen (zie figuur 2.5). De grootte van deze snelheid bepaalt de klap waarmee de tuimelaar op de klep slaat. Een te lage snelheid betekent dat de speling te traag wordt doorlopen. Een te hoge snelheid betekent een te grote belasting voor het mechanisme t.g.v. de botsingskrachten
2. De maximale positieve versnelling tesamen met de veerkracht bepalen de maximale kracht op de nok. Een te grote kracht op de nok betekent dat deze snel slijt. Te lage versnellingen kunnen er toe leiden dat de klep niet snel genoeg open staat.

3. De maximale negatieve versnelling bepaalt de benodigde veerkracht. Een hoge veerkracht leidt weer tot een hoge belasting van de nok en hiermee tot verhoogde slijtage. Een te lage veerkracht kan leiden tot het loskomen van de nokvolger van de nok.

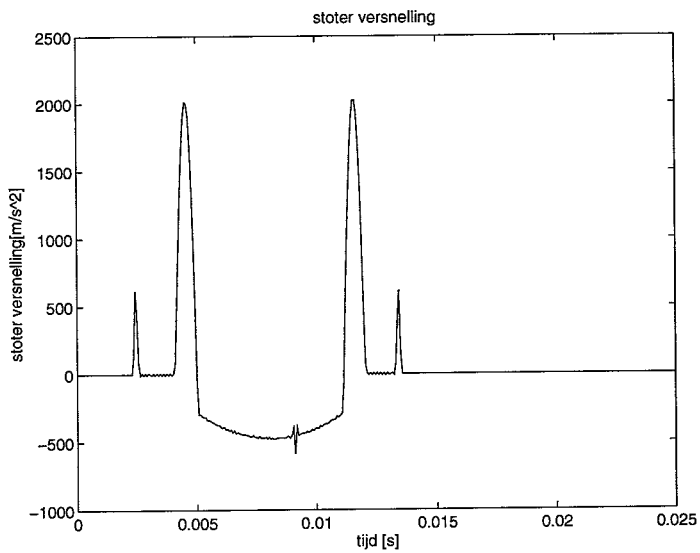
Een zorgvuldige afweging van al deze aspecten leidt vervolgens tot een 'optimaal' ontwerp van de nokfunctie.



Figuur 2.4: Nominale beweging van de stoter bij 1400 rpm van de nokkenas



Figuur 2.5: Nominale snelheid van de stoter bij 1400 rpm van de nokkenas



Figuur 2.6: Nominale versnelling van de stoter bij 1400 rpm van de nokkenas



## 3 Simulatie modellen in de literatuur

---

In het verleden is er al redelijk wat onderzoek gedaan naar nokmechanismen in machines [Kost 73] en in verbrandingsmotoren [Mews 94]. In het artikel [Mews 94] wordt een beschrijving gegeven van de simulatie van nokmechanismen in verbrandingsmotoren met hydraulische klepstoters zoals deze veel in de automobielindustrie worden toegepast. Voordat er aan de modellering in DADS wordt begonnen zal kort naar deze simulaties worden gekeken.

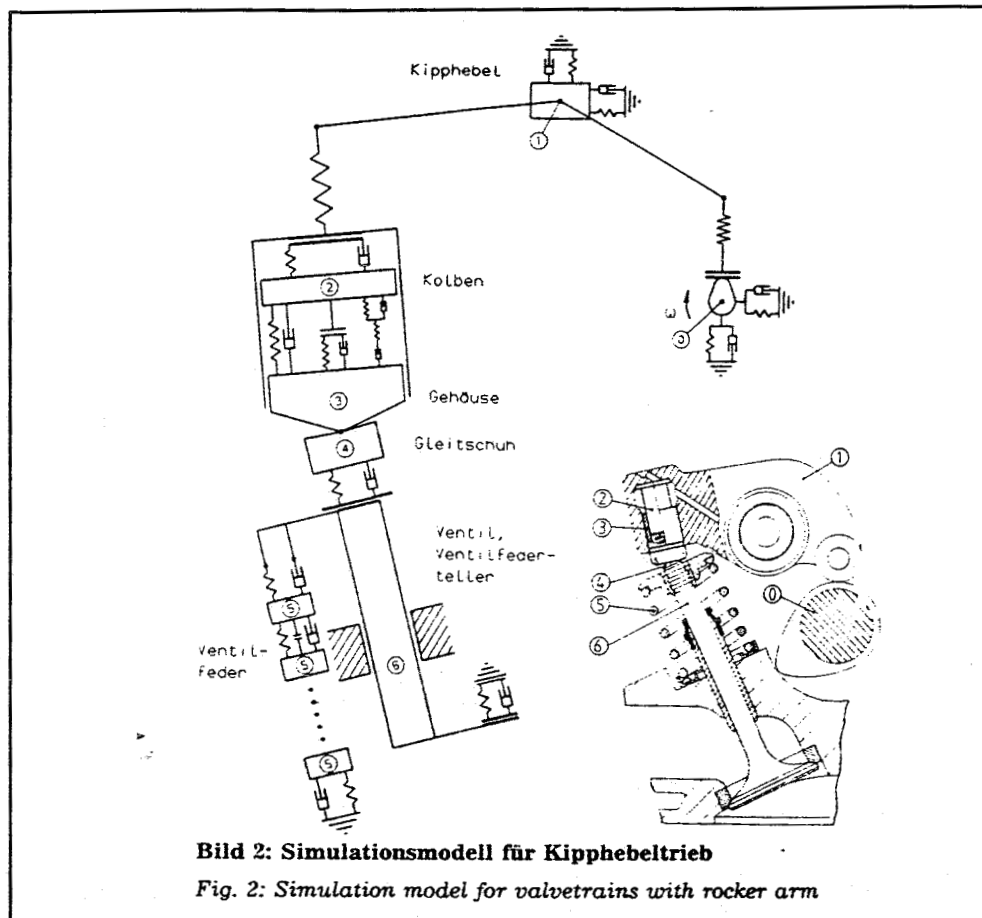
### 3.1 Het dynamische model

In figuur 3.1 staat een voorbeeld van een simulatiemodel voor een nokmechanisme met bovenliggende nokkenas zoals dat in [Mews 94] wordt gebruikt. In dit simulatiemodel is geprobeerd om een aantal parameters welke van belang zijn voor het dynamische gedrag van het mechanisme mee te nemen. Een aantal hiervan zijn:

- Om buigtrillingen en torsietrillingen in de nokkenas mee te kunnen nemen kan bovenop de nominale nokfunctie deze trilling worden gesuperponeerd.
- Het contactpunt tussen nok en tuimelaar wordt berekend aan de hand van de verplaatsingen van zowel de nok als de tuimelaar ten gevolge van de optredende belastingen.
- Om de wrijving tussen nok en tuimelaar mee te nemen is het mogelijk om de wrijvingscoëfficiënt als functie van de nokhoek op te geven.
- De wrijving tussen klep en klepgeleiding wordt door droge wrijving gemodelleerd.
- De stijfheid van de zitting, samen met de stijfheid van de klepschotel wordt, om meetresultaten mee te kunnen nemen, met een 7<sup>e</sup> graads polynoom opgegeven.

Het is duidelijk dat [Mews 94] een zeer complex simulatiemodel heeft opgesteld, waarin veel aspecten van het dynamisch gedrag van het mechanisme naar voren komen. Bij dit complexe model horen echter ook een groot aantal parameters. In het artikel wordt helaas weinig verteld over methodes om deze





Figuur 3.1: Dynamische model

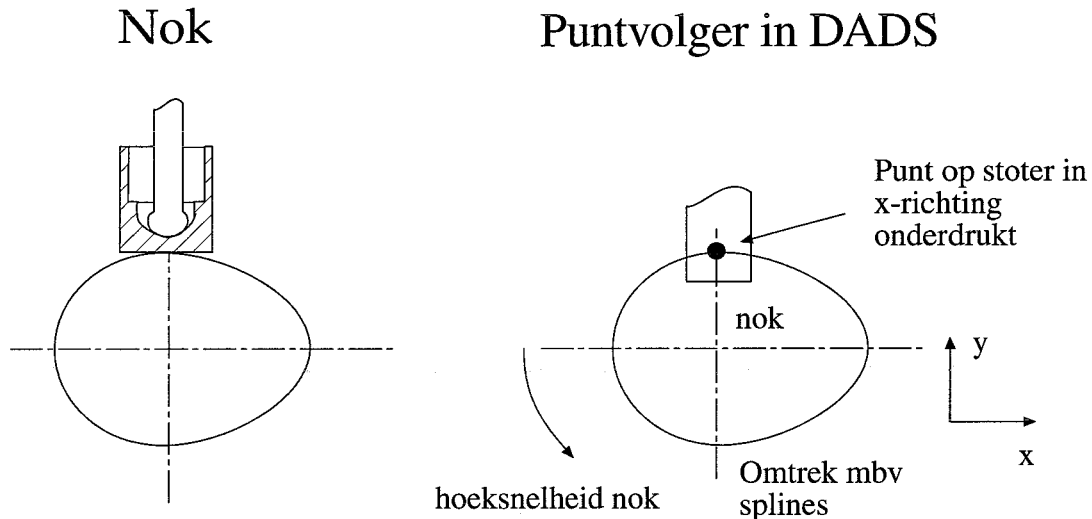
parameters te bepalen. Toch geeft het artikel duidelijk de complexiteit van het probleem en de belangrijke invloedsfactoren aan.

In dit hoofdstuk zal worden beschreven hoe met behulp van DADS een simulatiemodel is verkregen van het klepmechanisme van de inlaat van de zogenaamde RS-motor. In eerste instantie zal het model zeer eenvoudig van opzet zijn, mede omdat er maar weinig van het systeem bekend is, en in geen vergelijking staan tot het model van [Mews 94]. Er wordt begonnen met een afweging tussen de modellering in 2D en 3D. Vervolgens zal stap voor stap het DADS model worden toegelicht. Tot slot zullen de simulatie resultaten van dit model met starre lichamen worden bekeken. In bijlage A staat een overzicht van de gebruikte massa's, massatraagheden, veerstijfheden etc.

## 4.1 Afweging tussen modelleren in 2D of 3D

In wezen bestaat DADS uit 2 pakketten, namelijk DADS 2D en DADS 3D. Voor dat er ook maar iets ingevoerd kan worden moet er al duidelijk zijn welke van deze 2 pakketten gebruikt gaat worden. Een simpele uitbreiding van 2D naar 3D is in de loop van de invoer niet mogelijk. Een zorgvuldige afweging tussen de voordelen en nadelen van de twee mogelijkheden is dan ook een vereiste. Essentieel daarbij is of er bewegingen in de 3-richting kunnen plaatsvinden. Uiteindelijk is er gekozen voor DADS 2D omdat het probleem in wezen 2D is. Er is niets dat er op duidt dat DADS 3D meer waarde aan het model kan toevoegen. Verder kan het modelleren in 3D allerlei complicaties met zich mee brengen.

## 4.2 De nok



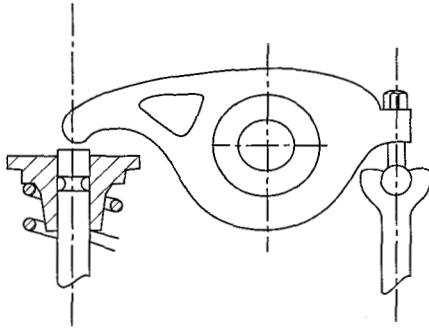
Figuur 4.1: De nok

Binnen DADS bestaat er de mogelijkheid om een zgn. *cam flat follower* te modelleren. De buitenomtrek van de nok moet hiertoe worden opgegeven. De platte volger blijft bij dit element altijd loodrecht op de buitenkant van de nok staan. Het probleem bij de modellering van de nok is dat de geometrie van de nok niet bekend is. De nokfunctie wordt bij DAF namelijk in de vorm van de stoterheffing opgegeven. De straal en de stoterheffing zijn niet dezelfde functie van de hoekverdraaiing van de nok. Om dit probleem te omzeilen is er niet gebruik gemaakt van de *cam flat follower* maar van de *point follower* binnen DADS (zie figuur 4.1). Met behulp van deze *point follower* kan er toch een idee worden gekregen over de krachten in het nok-contact. De geometrie van de nok is ingegeven door over 360 graden de stoterheffing in te geven in de vorm van numerieke waarden. Voor tussenliggende punten wordt vervolgens gebruik gemaakt van een multiple-spline die tussen deze punten interpolleert.

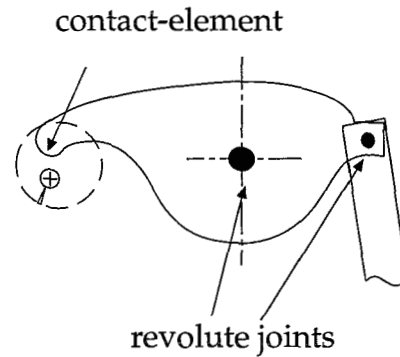
## 4.3 De tuimelaar

De tuimelaar is m.b.v. *revolute joints* aan de stoter en het blok bevestigd. Om de klepspeling te modelleren wordt er gebruik gemaakt van een contact element zoals in figuur 4.2 is getekend. Bij voorbaat kan worden gesteld dat het modelleren van een contact erg moeilijk is. Wat voor fysische processen er

## Tuimelaar



## Tuimelaar in DADS



Figuur 4.2: De tuimelaar

plaats vinden tijdens een botsing en hoe deze gemodelleerd kunnen worden zonder daarbij numerieke problemen te veroorzaken blijft een groot probleem. Om wat meer duidelijkheid te verschaffen over de manier waarop DADS tegen een contact aankijkt, zal er in paragraaf 4.3.1 wat dieper op dit onderwerp worden ingegaan.

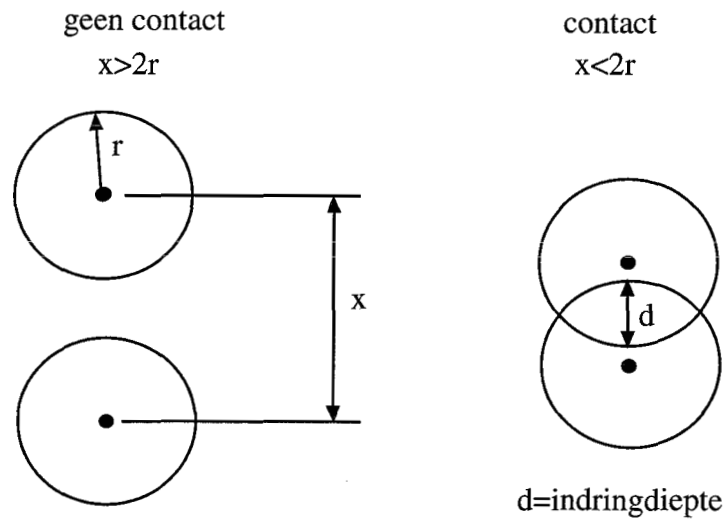
### 4.3.1 Het contactelement

Binnen DADS bestaat er in de nieuwe versie 7.0 en 7.5 de mogelijkheid om met contacten te werken. De fysische beschrijving van deze contacten is echter zeer eenvoudig. DADS kijkt namelijk of twee punten elkaar zo ver naderen dat er contact optreedt,  $x < 2r$  (zie figuur 4.3). Wordt aan deze voorwaarde voldaan dan oefenen de twee naderende punten een kracht op elkaar uit welke beschreven wordt door de volgende relatie.

$$F_{contact} = k_{contact}d + b_{contact} \dot{d} \quad (4.1)$$

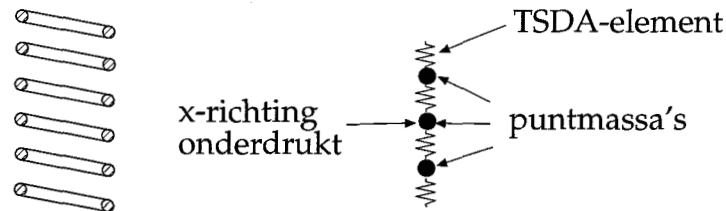
Met  $k_{contact}$  De contactstijfheid in N/m,  $d$  de indringdiepte in m,  $b_{contact}$  de contactdemping in Ns/m.

De demping treedt hierbij alleen op tijdens het naderen van twee punten, de twee punten plakken dus niet aan elkaar vast. Het grootste probleem zal worden om de juiste contact-stijfheid en demping te vinden.



Figuur 4.3: Het contactelement

#### 4.4 De veer



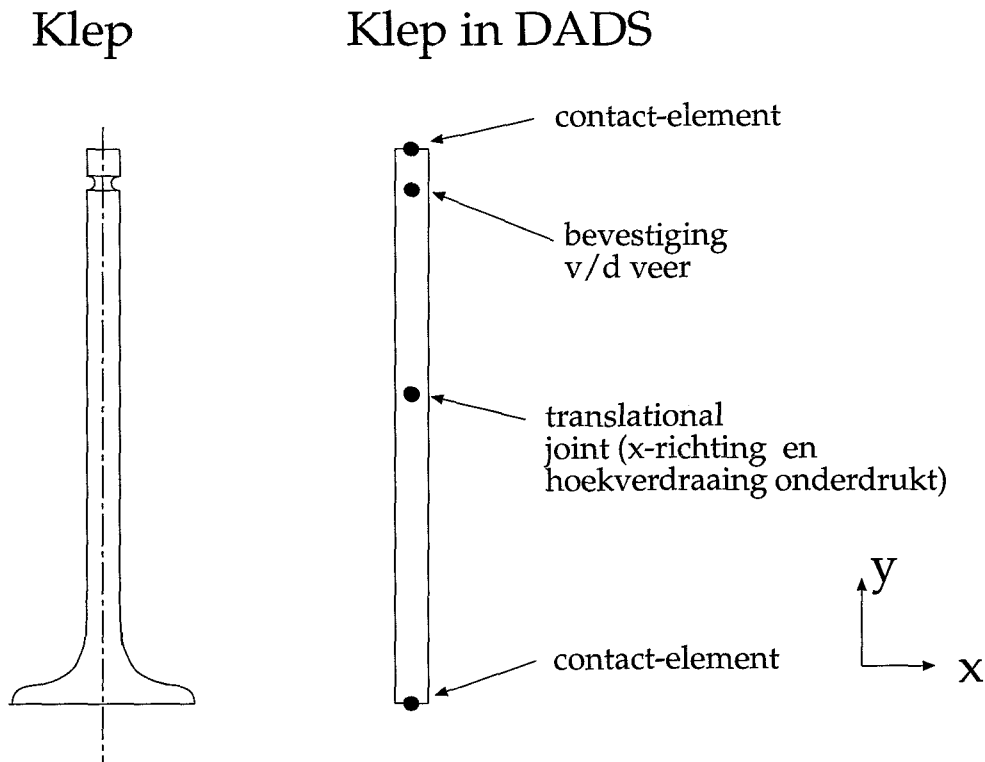
Figuur 4.4: Discretisering van de veer

Ook hier bezit DADS weer een standaard element waarmee een veer gemodelleerd kan worden: Het *TSDA* (*Translational Spring Damper Actuator*) element. Met dit element wordt er echter geen massa aan de veer toegekend. Het beschrijft alleen maar de relatie tussen kracht en verplaatsing volgens de volgende welbekende relatie.

$$F_{veer} = k_{veer}(x - x_0) + b_{veer} \dot{x} \quad (4.2)$$

Om nu toch de massa van de veer mee te nemen is deze in een aantal punten geconcentreerd (zie figuur 4.4). De geconcentreerde massa's zijn vervolgens met *TSDA* elementen verbonden. Op deze manier kan er toch iets gezegd worden over het dynamisch gedrag van de veer.

## 4.5 De klep



Figuur 4.5: De klep

De klep wordt hier als een star lichaam gemodelleerd. Het contact tussen klep en tuimelaar wordt gemodelleerd door het contactelement zoals beschreven staat bij de tuimelaar. De veer grijpt boven aan de klep aan en onder aan de klep wordt het contact tussen de klep en de zitting door een zelfde contactelement als bij de tuimelaar gemodelleerd.

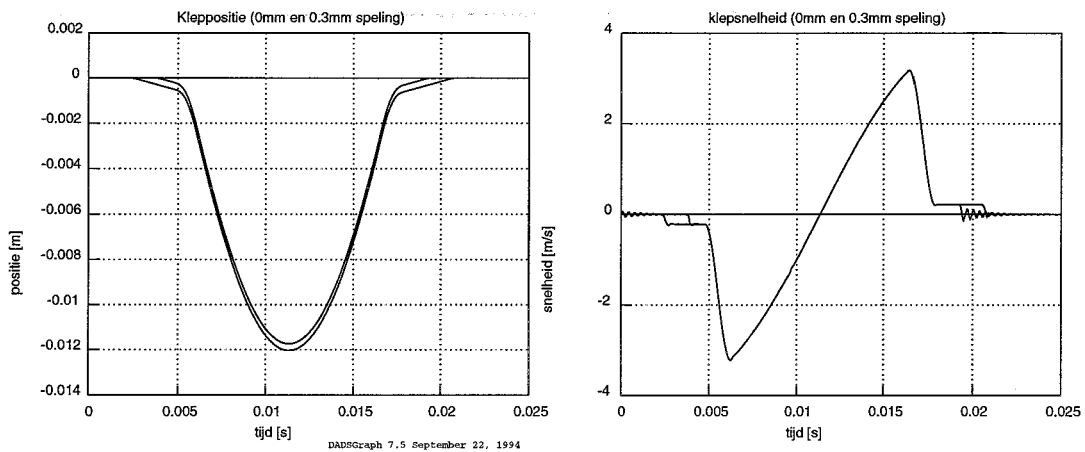
## 4.6 Simulatie

In grote lijn zijn de resultaten van de dynamische analyse met DADS gelijk aan die welke met de inverse dynamische analyse verkregen zijn. Tijdens de simulaties kwam echter al 'snel' naar voren dat de parameters welke het contact beschrijven van essentieel belang zijn. Variatie van deze parameters hebben een grote invloed op onder andere de versnellingen van de klep. Om hiervan een idee te geven zijn er bij verschillende waarden van deze parameters simulaties

uitgevoerd. Het blijft echter een groot probleem om de juiste waarden van deze parameters te bepalen. Het blijft zelfs de vraag of de beschrijving die DADS van een contact geeft überhaupt bruikbaar is ([Mews 94] gebruikt een 7<sup>e</sup> graads polynoom). Een volledig overzicht van de simulaties staat in bijlage B.

#### 4.6.1 Kleppositie en snelheid bij variabele speling

Er zijn 2 simulaties uitgevoerd bij een speling van 0 mm en 0.3 mm speling. De resultaten hiervan staan in figuur 4.6. Het enige dat dit figuur toont is is dat de nominale nokfunctie goed gevolgd wordt. Bij 0.3 mm speling gaat de klep wat later open, maar dit is natuurlijk logisch.



Figuur 4.6: Kleppositie en klepsnelheid bij variabele speling

#### 4.6.2 Variatie van contactdemping

Er zal nu naar de invloed van de demping worden gekeken. Voor de demping bij de klep/tuimelaar als ook voor de demping bij de klepzitting worden verschillende waarden genomen. De invloed is het sterkst aanwezig bij de versnellingen van de klep. De stijfheid van beide contacten zal constant op  $10^8$  N/m worden gehouden. Deze grote (maar niet overdreven) stijfheid is bij benadering gekozen.

Uit de simulaties blijkt dat de aangenomen veerstijfheid bij de contactelementen in combinatie met de klepmassa eigentrillingen initiëren met relatief hoge eigenfrequenties. Deze eigenfrequenties worden met name aangeslagen door de botsing na het doorlopen van de speling en de botsing van de klep op de klepzitting. Een wezenlijke vraag is nu of dit in werkelijkheid ook

gebeurt. Bijvoorbeeld gaat de klep na het sluiten nog even trillen op zijn zitting. Metingen aan het werkelijke systeem moeten hierover meer duidelijkheid verschaffen.

Overzicht van figuren:

figuur nummer	stijheid contacten in N/m	demping b1 klepzitting in Ns/m	demping b2 klep-tuimelaar in Ns/m	speling tuimelaar-klep in mm
4.7	$10^8$	$10^3$	$10^2$	0
4.8	$10^8$	$10^4$	$10^3$	0
4.9	$10^8$	$10^5$	$10^4$	0
4.10	$10^8$	$10^4$	$10^3$	0.3

Opmerkingen bij de figuren:

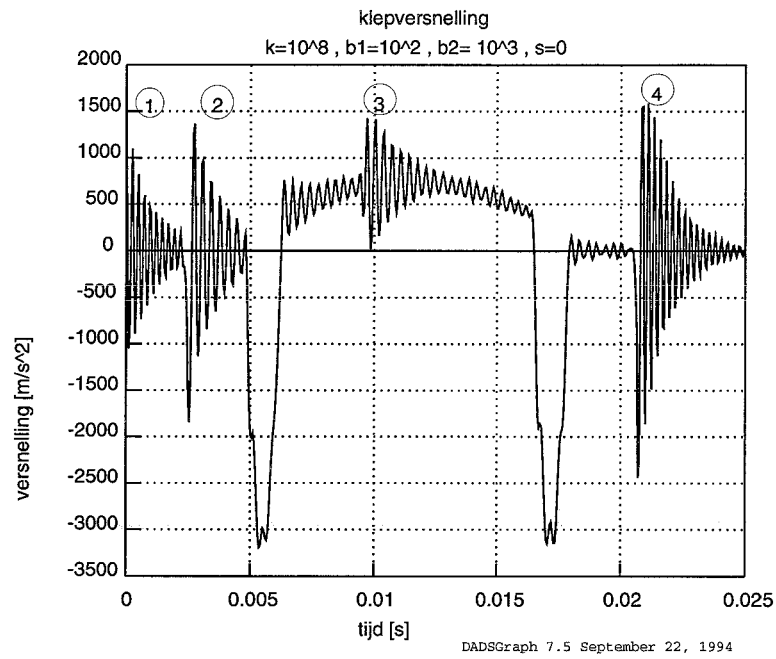
In figuur 4.7 worden een aantal keren trillingen geïnitieerd.

1. Bij aanvang van de simulatie moet het systeem eerst in evenwicht komen.
2. Trillingen welke ontstaan na de botsing van de tuimelaar met de klep.
3. Trillingen welke ontstaan door een oneffenheid in de nominale nokkromme.
4. Trillingen welke ontstaan door de botsing van de klep op de zitting.

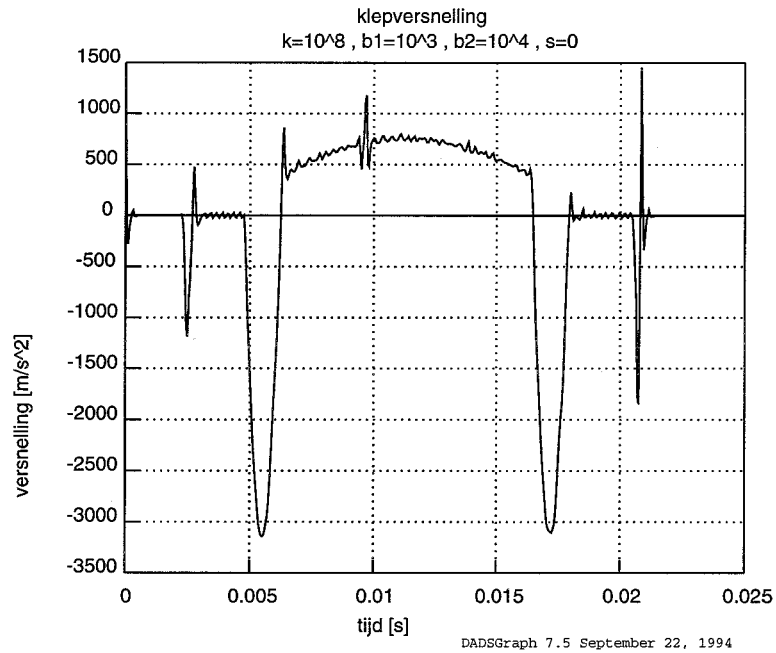
De verschillen tussen de simulatie zonder speling en met speling, respectievelijk figuur 4.8 en 4.10:

1. Door de speling speelt de demping in het contact tussen tuimelaar en klep bij een gesloten klep geen rol. Bij het systeem zonder speling is dit wel het geval. De geïnitieerde trilling zal daarom in het spelingsvrije geval sneller uitdempen.
2. Hogere versnelling t.g.v. de botsing na het doorlopen van de speling.

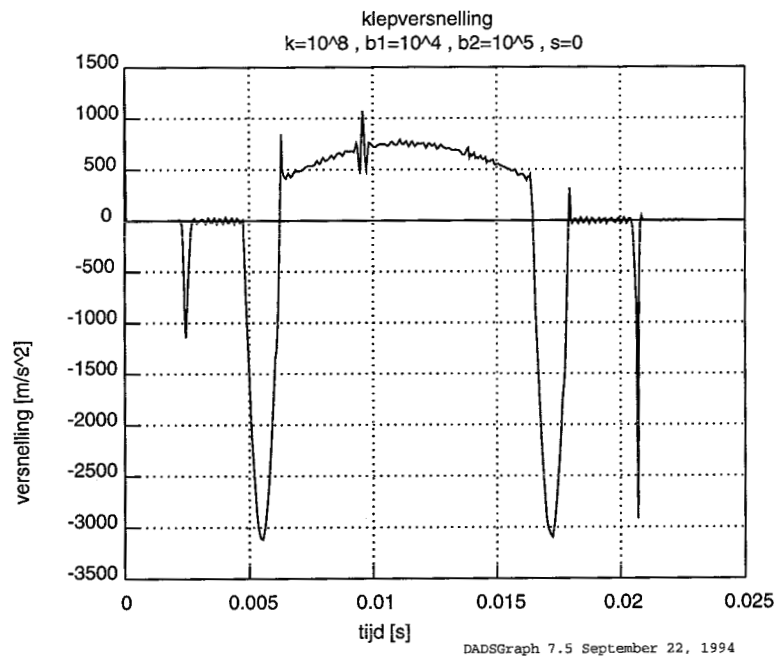




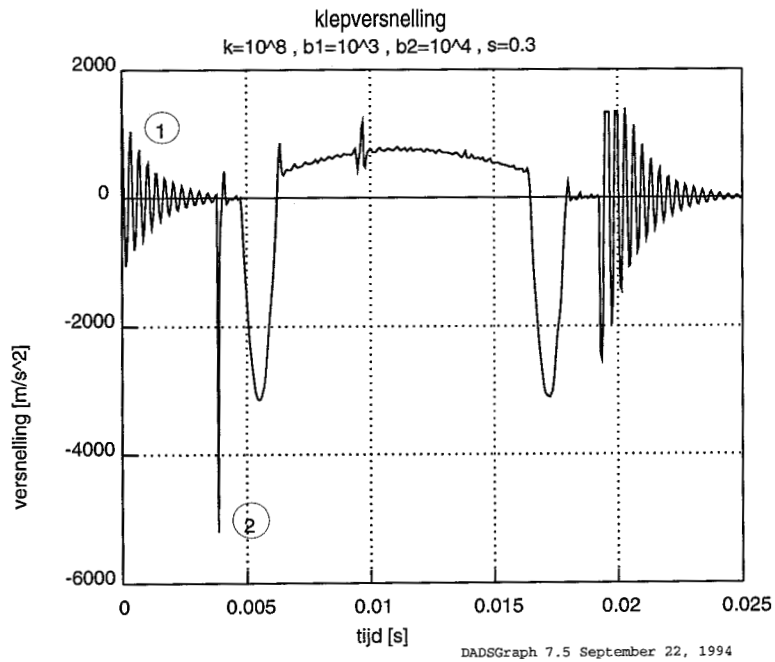
Figuur 4.7: Klepversnelling bij lage demping



Figuur 4.8: Klepversnelling bij gemiddelde demping



Figuur 4.9: Klepversnelling bij hoge demping



Figuur 4.10: Klepversnelling bij gemiddelde demping met speling



## 5

# DADS-model met flexibele klep

---

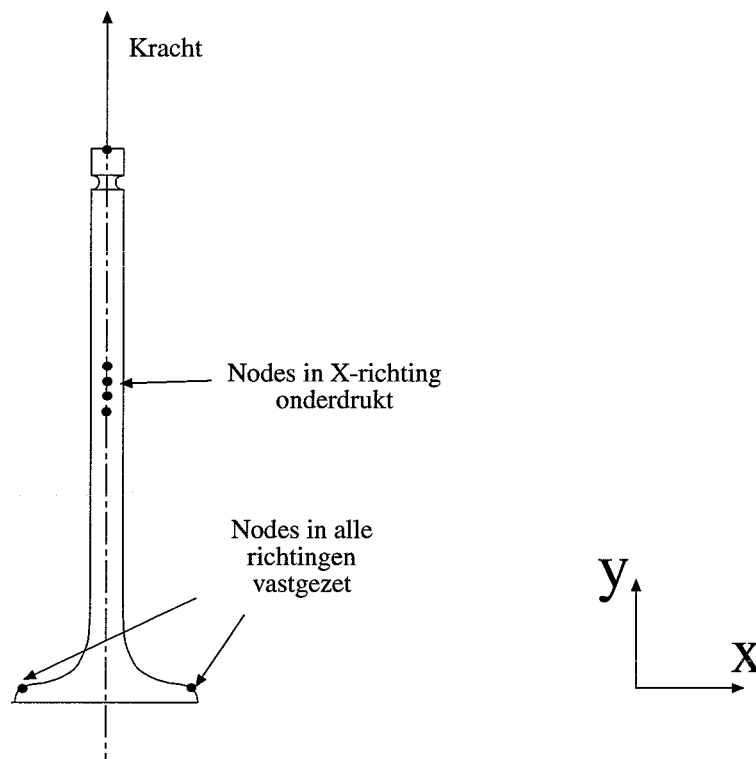
In dit hoofdstuk zal worden geprobeerd om flexibiliteit aan de klep toe te voegen. Hiervoor moet eerst een dynamische analyse m.b.v. een eindige elementen pakket worden uitgevoerd. De uitkomsten hiervan kunnen vervolgens met een speciale *intermediate processor (ip)* worden omgezet naar een geschikt formaat voor DADS. In de zogenaamde flex. file staan de knooppuntscoördinaten, de eigenmodes, de modale massamatrix, de modale stijfheidsmatrix en de lumped massamatrix. Deze flex. file kan in DADS worden ingelezen. Na deze stap is er een model verkregen waar bovenop flexibiliteit is gezet. Voor details zie [Meeuwse 93].

### 5.1 De NASTRAN som

DADS beschikt over de mogelijkheid om zowel statische modes als dynamische modes in een model mee te nemen. In eerste instantie was dan ook het idee om hiervan gebruik te maken. Het komt er kort op neer dat er een som moet worden gedraaid waarin het vrije trillingsgedrag van de klep bepaald wordt, de eigenmodes, en een tweede som waarin de klep op een bepaald punt wordt ingeklemd en in een bepaalde vorm wordt gedrukt door een belasting, de statische modes (zie [Meeuwse 93]). De uiteindelijke resultaten van deze 2 sommen worden vervolgens door *ip* vertaald naar een voor DADS geschikt formaat. De inklemming en belasting voor de berekening van de statische mode staat in figuur 5.1. De onderkant van de klep is hierbij vasgezet (als het ware aan de zitting vast gemaakt), in het midden van de klep zijn een aantal nodes in x-richting onderdrukt (de geleiding van de klep) en aan de top is een belasting aangebracht in y-richting.

### 5.2 IP en de problemen bij DAF

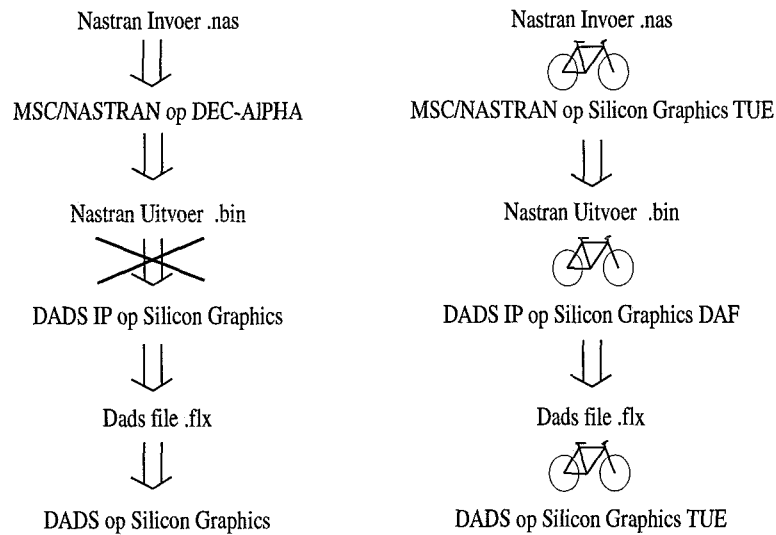
De uitkomsten van de twee NASTRAN sommen moet vervolgens met *IP* worden omgezet. Hier begonnen de echte problemen te ontstaan. NASTRAN levert namelijk de resultaten in zogenaamde bin-files welke machine afhankelijk zijn. NASTRAN draait bij DAF op een DEC-alpha en *IP* en DADS draaien beide op een Silicon Graphics machine. In wezen was dit het einde van het



Figuur 5.1: De statische mode van de klep

verhaal. Echter op het rekencentrum van de TUE staat een Silicon Graphics waarop NASTRAN draait. De bin-files die NASTRAN op de Silicon Graphics aanmaakte waren nu in te lezen met behulp van *IP*. Dit hield echter wel in dat er flink wat data transport tussen de TUE en DAF moest plaats vinden. Aangezien er geen goede directe data communicatie tussen de TUE en DAF mogelijk is betekende dit dat er flink wat gefietst moest worden met een datacardridge (zie figuur 5.2).

Vervolgens bleken de resultaten die *IP* genereerde niet correct te zijn. Als in *IP* alleen de eigenmodes werden omgezet bleken de eigenfrequenties die NASTRAN uitgerekend had niet meer in de door *IP* gegenereerde flex-file voor te komen. Na ongeveer alle mogelijke veranderingen in de NASTRAN invoerfiles te hebben doorgevoerd bleek dit probleem onoplosbaar. Uiteindelijk is er voor gekozen om de statische modes over boord te zetten en alleen met de relevante eigenmodes verder te gaan. Hierbij hoeft slechts één som in NASTRAN te worden gedraaid, waarbij dit probleem niet naar voren kwam. In het vervolg zal de klep flexibel worden gemaakt door slechts de eerste relevante eigenmode in longitudinale richting mee te nemen (6 kHz). Buigtrillingen zullen door de aanwezigheid van de klepleiding waarschijnlijk niet voorkomen.



Figuur 5.2: Stroomschema van de vertaalproblemen

Aangezien MSC/Nastran alleen met 3D modellen kan werken zijn, moeten na de *IP* vertaling de vrijheidsgraden in z-richting worden onderdrukt.

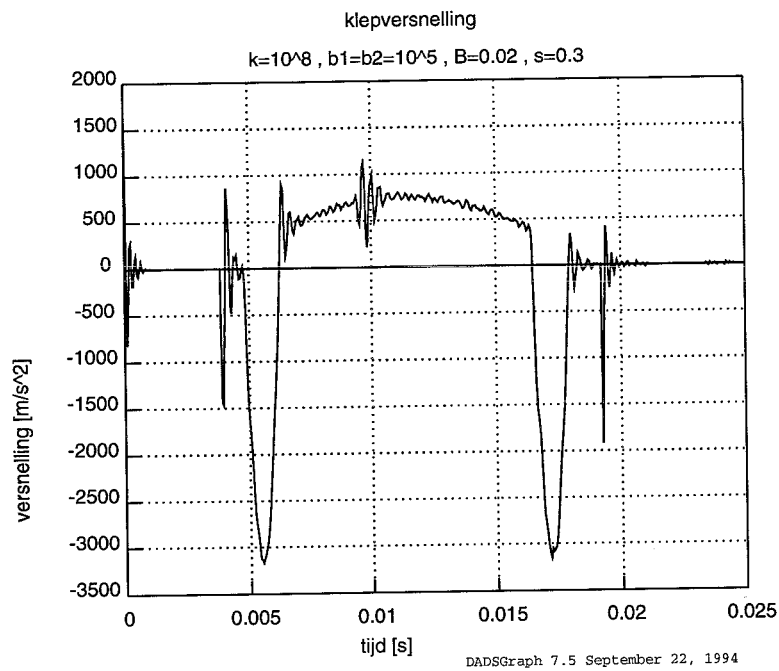
### 5.3 Resultaten van de simulaties

Ook hier was het kiezen van de juiste contactstijfheid en demping een groot probleem. In wezen heeft de flexibiliteit van de klep een deel van de contactstijfheid in zich. Het flexibel zijn van de klep is er mede de oorzaak van dat het contact niet oneindig stijf is. Er zijn dan ook een aantal simulaties uitgevoerd met verschillende relatieve demping van de eigentrillingen van de klep en verschillende stijfheden in de contacten. Er is gekozen voor een relatief hoge demping in de contacten zodat de eigentrilling van de klep zichtbaar wordt en niet de trilling van de klep op de contacten. Verder is de speling tussen de klep en de tuimelaar gevarieerd (zie bijlage C).

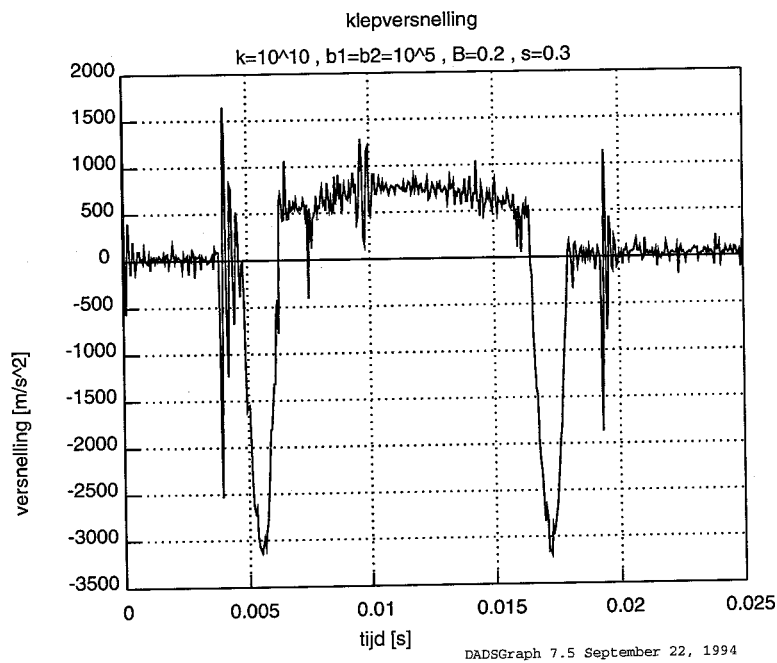
Overzicht van figuren:

figuur nummer	contactstijfheid in N/m	demping contacten in Ns/m	relatieve demping eigen trilling	speling in mm
5.3	$10^8$	$10^5$	0.02	0.3
5.4	$10^{10}$	$10^5$	0.2	0.3
5.5	$10^{10}$	$10^5$	0.02	0.3
5.6	$10^{11}$	$10^5$	0.02	0.3
5.7	$10^{10}$	$10^5$	0.05	0.3

In bijlage C staan de resultaten uitgebreid weergegeven, waarbij nog meer variaties in parameters zijn uitgevoerd. Bij alle simulaties blijkt dat de invloed van de contactstijfheid en de relatieve demping van de eigen mode van groot belang zijn voor de uiteindelijke resultaten. Bij een te lage contactstijfheid  $k = 10^8$  N/m wordt de eigen trilling van de klep bijna niet aangeslagen. Door het opvoeren van de contactstijfheid is dit wel het geval. De eigen trillingen worden dan voornamelijk aangeslagen tijdens het openen en sluiten van de klep, dus elke keer als er een botsing plaats vindt. In de figuren 5.3 tot en met 5.7 valt ook een hoop 'gras' te zien. Er moeten waarschijnlijk ook nog enkele vraagtekens worden geplaatst bij de numerieke stabiliteit van de simulaties.

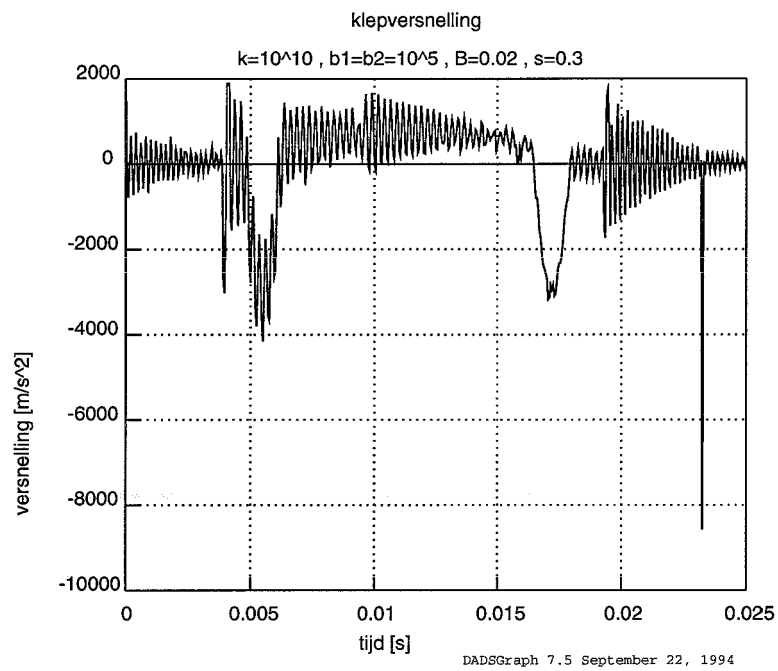


Figuur 5.3: Klepversnelling

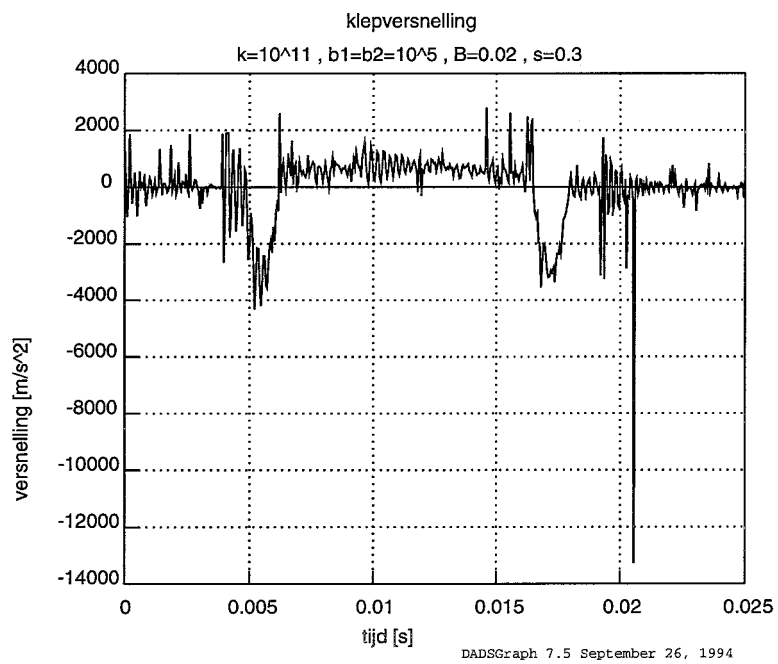


Figuur 5.4: Klepversnelling

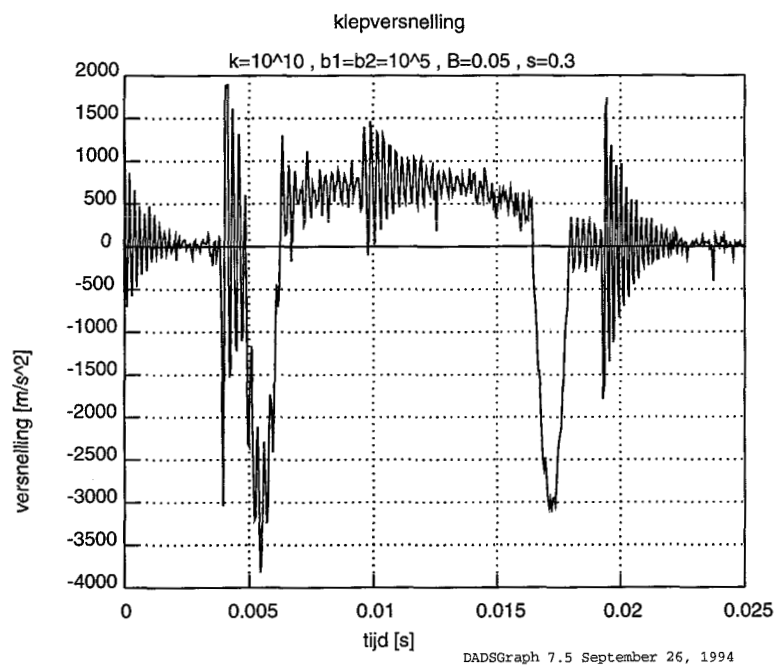




Figuur 5.5: Klepversnelling



Figuur 5.6: Klepversnelling



Figuur 5.7: Klepversnelling



Uit de uitgevoerde simulaties blijkt hoe gevoelig het model is voor parameter-varianties in het contact. De processen die zich afspelen tijdens het doorlopen van de speling en tijdens de botsing die daar op volgt zijn essentieel voor het functioneren van het mechanisme. Tijdens de botsing worden de eigentrillingen van alle flexibele onderdelen, in dit geval de klep, aangeslagen. De mate waarin dit gebeurt is afhankelijk van de onbekende parameters in dit contact en de te doorlopen speling. Om meer te weten te komen over het dynamisch gedrag van het systeem zal vooral moeten worden gekeken naar de mate waarin de eigentrillingen worden aangestoten en op welke manier deze uitdempen. Een goede beschrijving van het contact is noodzakelijk om het meenemen van meer flexibele elementen in het model zinvol te maken.

In wezen wordt het eindig zijn van de contactstijfheid veroorzaakt door de flexibiliteit van de verschillende lichamen. Als er een 'perfecte' beschrijving zou zijn van de flexibiliteit van de lichamen, dan zou in het model de contactstijfheid erg hoog kunnen worden genomen, zodat de werkelijke contactstijfheid wordt gemodelleerd door de flexibiliteit van de lichamen. In werkelijkheid is er geen perfecte beschrijving van de flexibiliteit van de lichamen en moet het flexibele gedrag van de 'contactzone' worden beschreven met een contactelement.

Om deze manier van modelleren in DADS te valideren zijn goede metingen aan het systeem onontbeerlijk. De complexiteit van het mechanisme en de problemen om er goed aan te meten maken een goede validatie echter onmogelijk. Het is dan ook aan te raden om eerst een eenvoudig mechanisme te maken waaraan wel goed kan worden gemeten en welke de zelfde problematiek heeft. De modellering van flexibele elementen samen met contactelementen in DADS zou dan wel goed kunnen worden gevalideerd.

In het uiteindelijke model, waarin het contact 'goed' beschreven wordt, zal naar voren komen dat het doorlopen van de speling in meer of mindere mate ongewenste trillingen veroorzaakt. Om dit probleem te voorkomen zal de speling uit het mechanisme moeten verdwijnen, zonder dat daarbij de klep niet sluit. Een oplossing hiervan is bijvoorbeeld het toepassen van hydraulische klepstoters. Binnen de automobiël-industrie wordt deze constructie al veel toegepast samen met de bovenliggende nokkenas. Waarschijnlijk is de toepassing hiervan ook mogelijk in samenhang met de onderliggende nokkenas. Bijkomend voordeel is dat in de nokfunctie geen deel meer voorkomt om de

speling te doorlopen. Dit biedt extra tijd om met een tammere nokfunctie (minder grote versnellingen) toch de klep voldoende snel te laten openen. Verder hoeft de klepspeling niet meer nagesteld te worden.

## Bibliografie

---

- [Meeuwse 93] J.P. Meeuwse, *Flexibele elementen in een multibody omgeving*  
WFW rapport 93.186, TUE 1993.
- [Latex 84] R.de Bruin ea.  
CWI syllabus Publiceren met LaTeX  
1988 stichting mathematisch centrum, Amsterdam.
- [Mews 94] *Dynamische Simulation von Ventiltrieben mit hydraulischem  
Spielausgleich*  
Horst Mews, Hendrik Brehler, ua.  
MTZ Motortechnische Zeitschrift 55 1994.
- [Kost 73] M.P. Koster, *Vibrations of cam mechanisms and their consequences  
on the design*, dissertatie TH-Eindhoven 1973.
- [Dads 94] Reference Manual DADS 7.5  
CADSI 1994.



## A Gebruikte parameters en instellingen

---

parameter	waarde
toerental nokkenas	1400 rpm
massa stoter+stoterbus	0.432 kg
massatraagheid tuimelaar	$0.21 \cdot 10^{-3} \text{ kgm}^2$
massa tuimelaar + stelschroef	0.237 kg
massa klep + veerschotel + deel van de veer	0.300 kg
massa veer	0.117 kg
veerstijfheid	34400 N/m
veerdemping	5 Ns/m
voorspanning veer	435 N
maximale integratiestap	$1 \cdot 10^{-6} \text{ s}$
printinterval	$5 \cdot 10^{-5} \text{ s}$
simulatie tijd	0.025 s
integratie methode	variabel

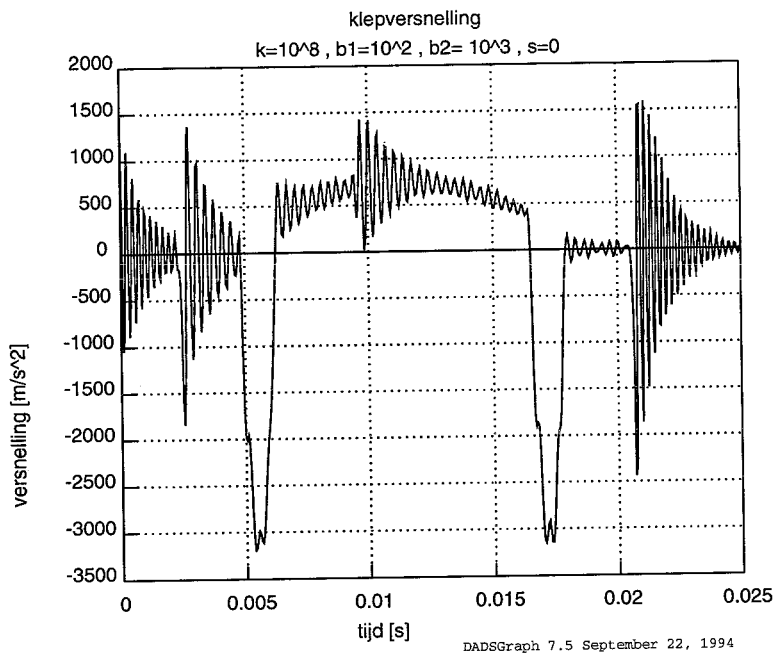


## **B** Resultaten simulaties met starre klep

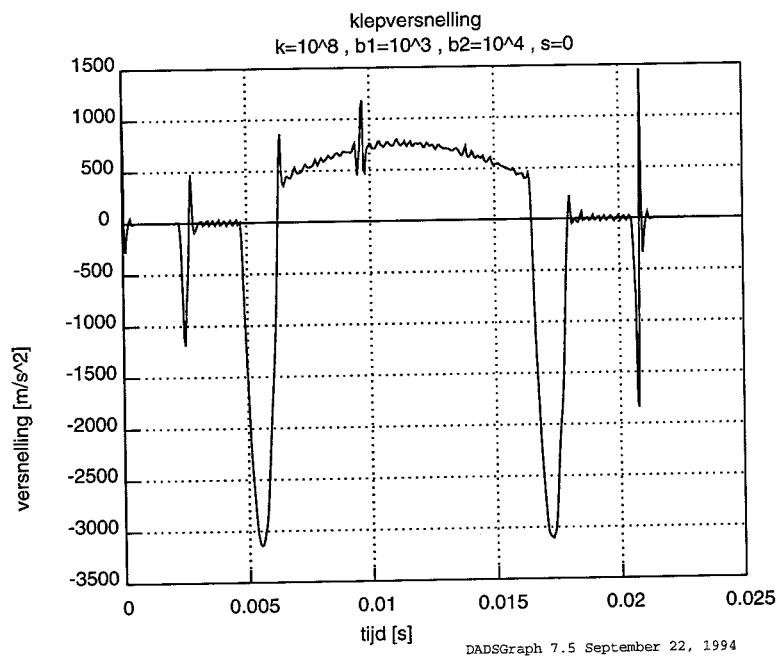
---

Parameters:

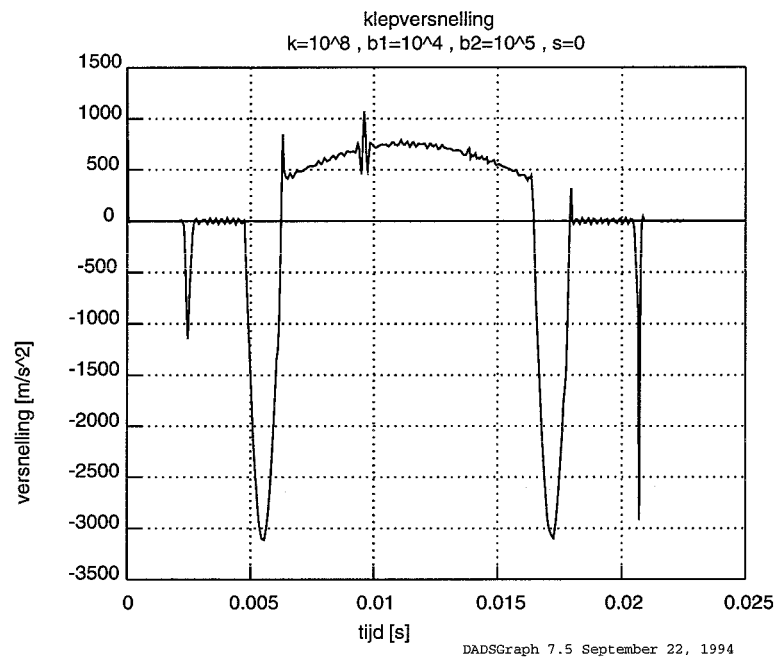
- $k$ :stijfheid in beide contactelementen in N/m
- $b_1$ :damping in het klepzitting-contact in Ns/m
- $b_2$ :damping in het klep-tuimelaar-contact in Ns/m
- $s$ :speling tussen tuimelaar en klep in mm



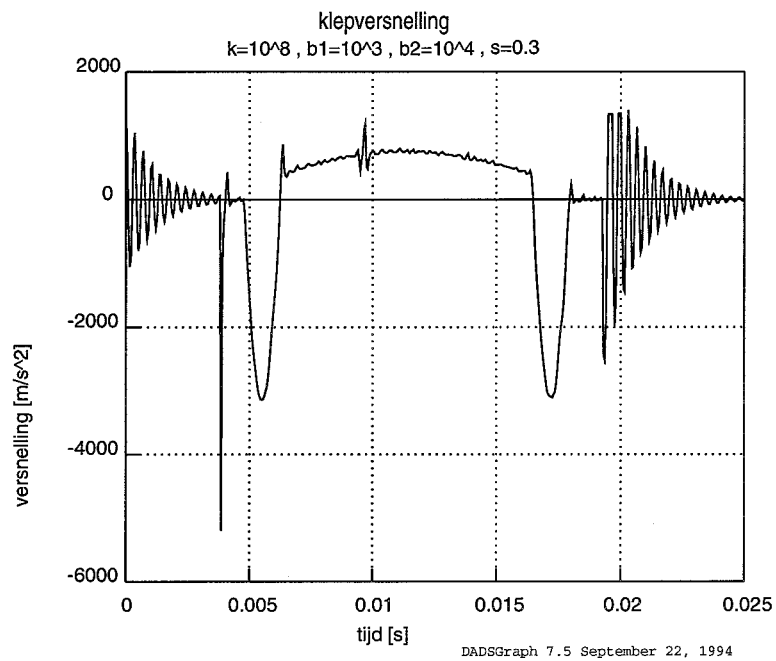
Figuur B.1:



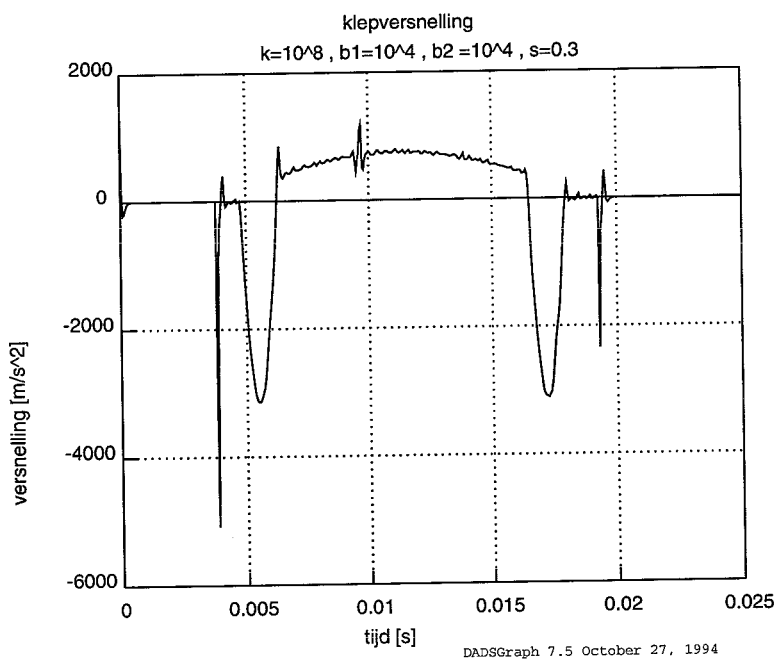
Figuur B.2:



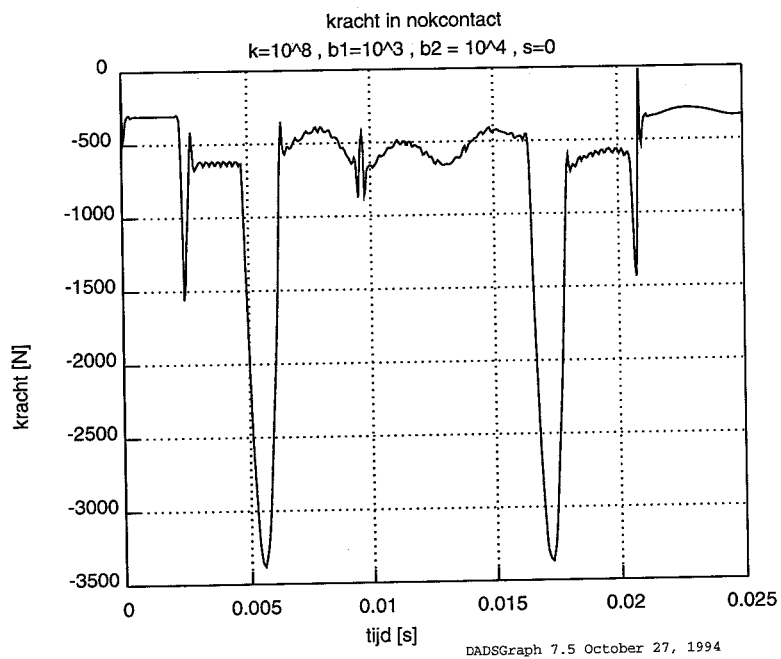
Figuur B.3:



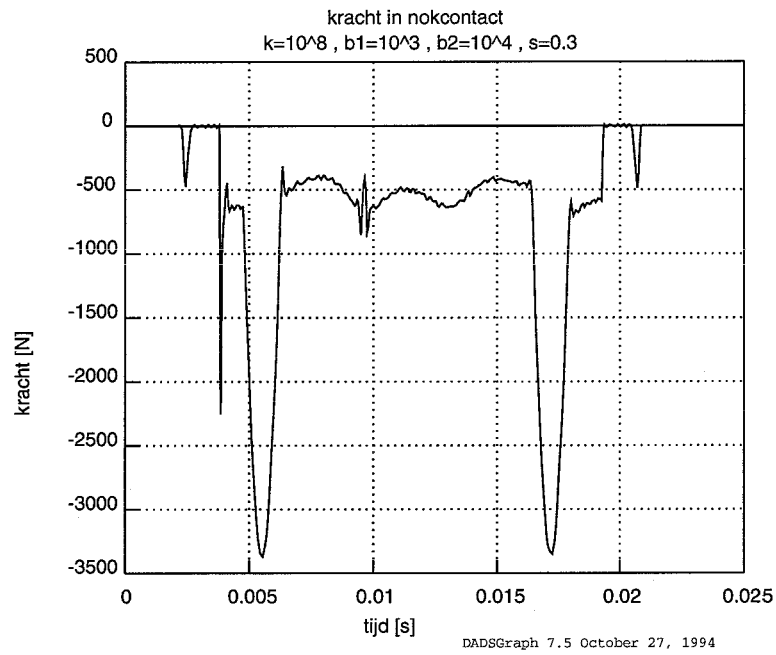
Figuur B.4:



Figuur B.5:



Figuur B.6:



Figuur B.7:



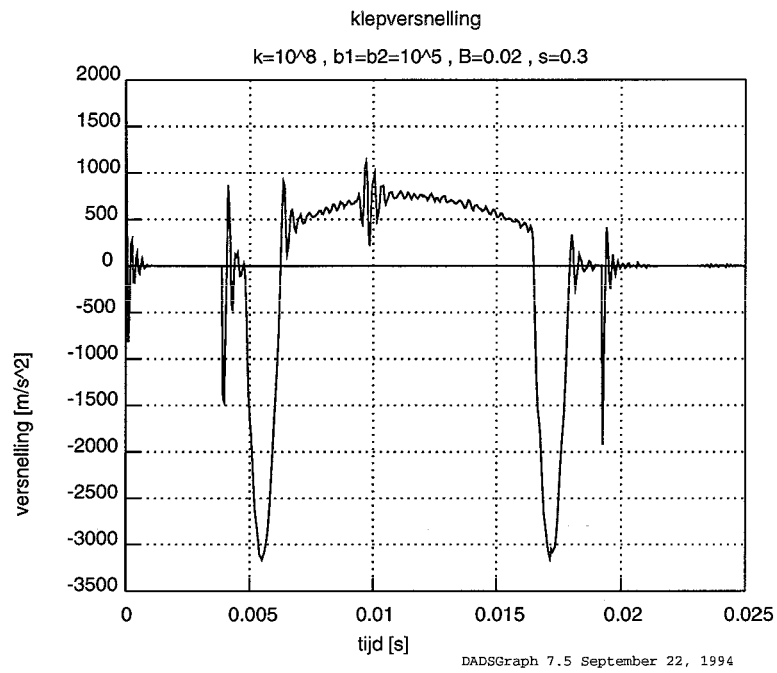
Figuur B.8:

## C Resultaten simulaties met flexibele klep

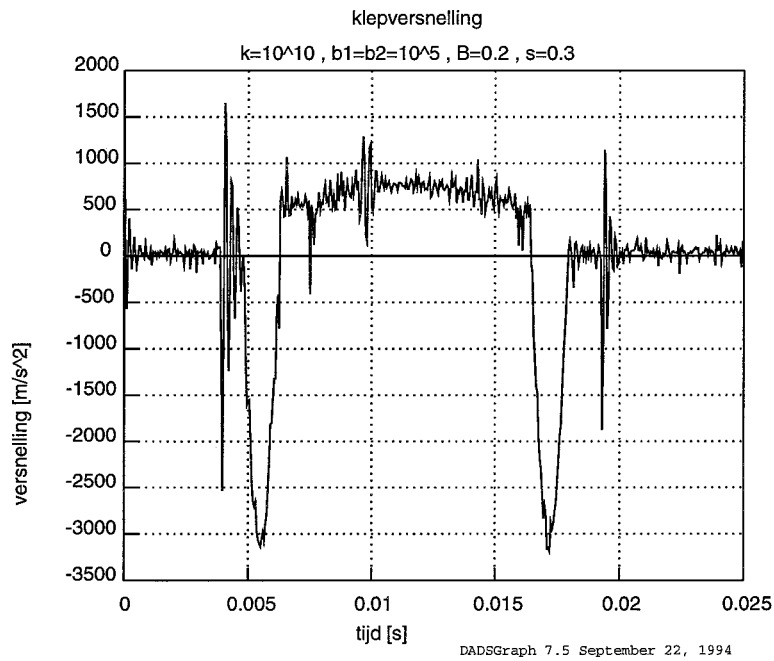
---

Parameters:

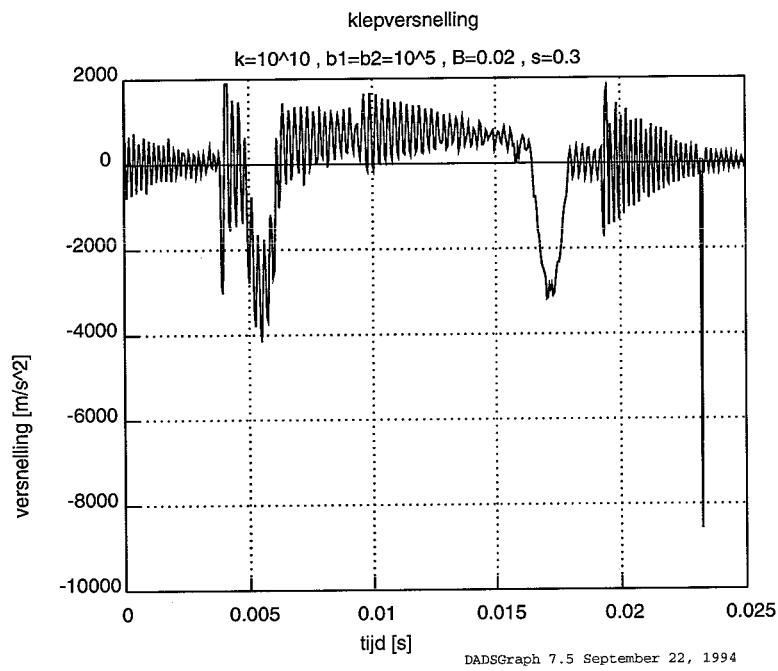
- k:stijfheid in beide contactelementen in N/m
- b1:demping in het klepzitting-contact in Ns/m
- b2:demping in het klep-tuimelaar-contact in Ns/m
- B:relatieve demping van de eigentrilling van de klep
- s:speling tussen tuimelaar en klep in mm



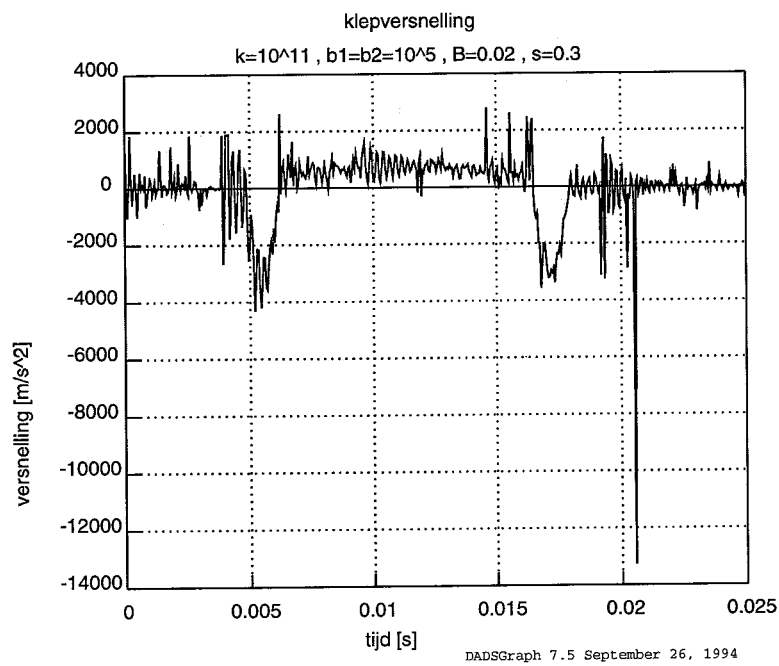
Figuur C.1:



Figuur C.2:

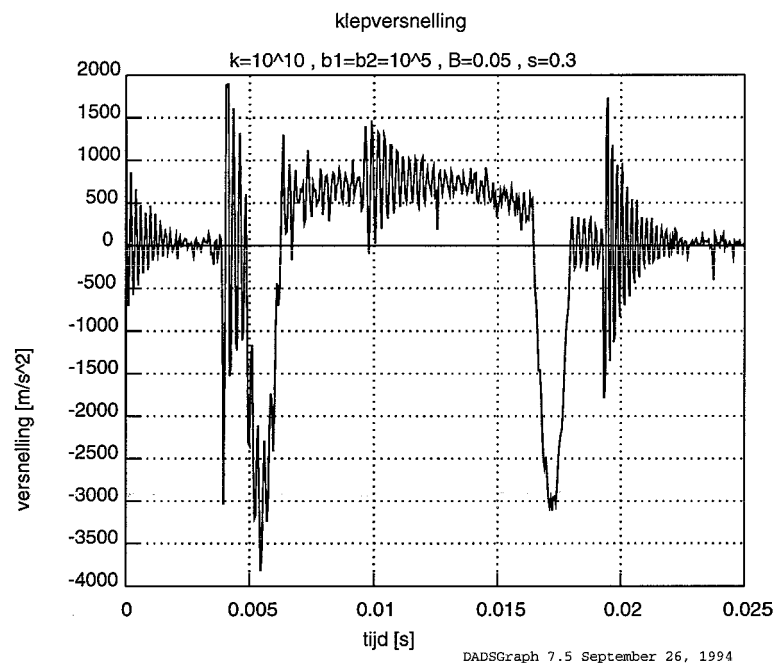


Figuur C.3:

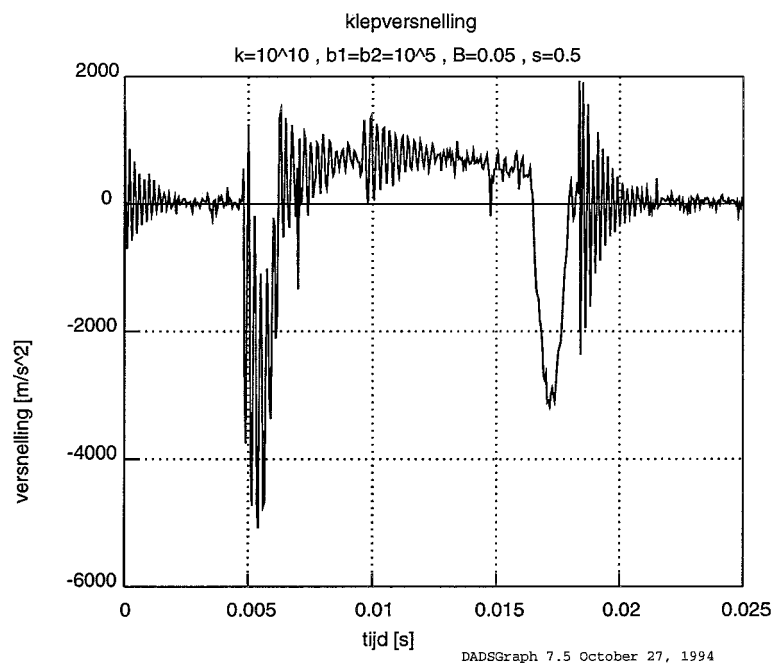


Figuur C.4:

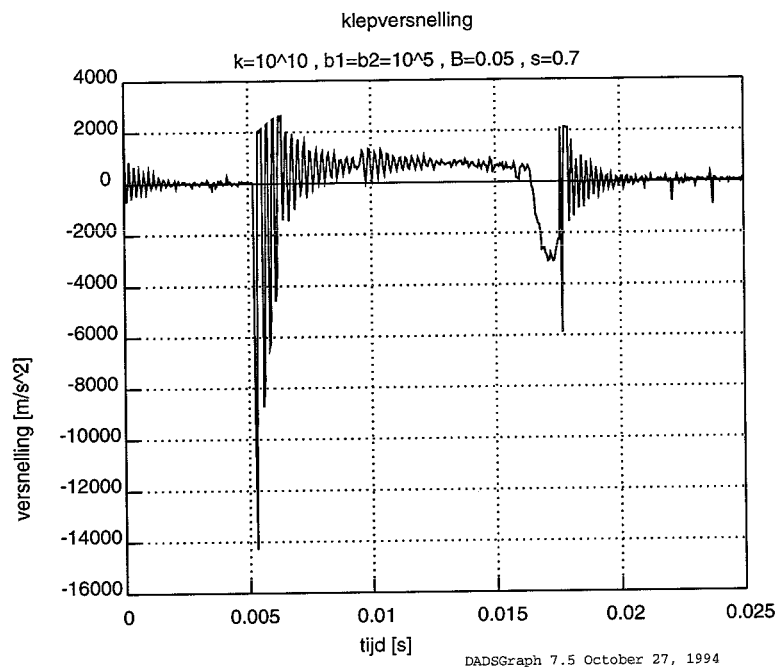




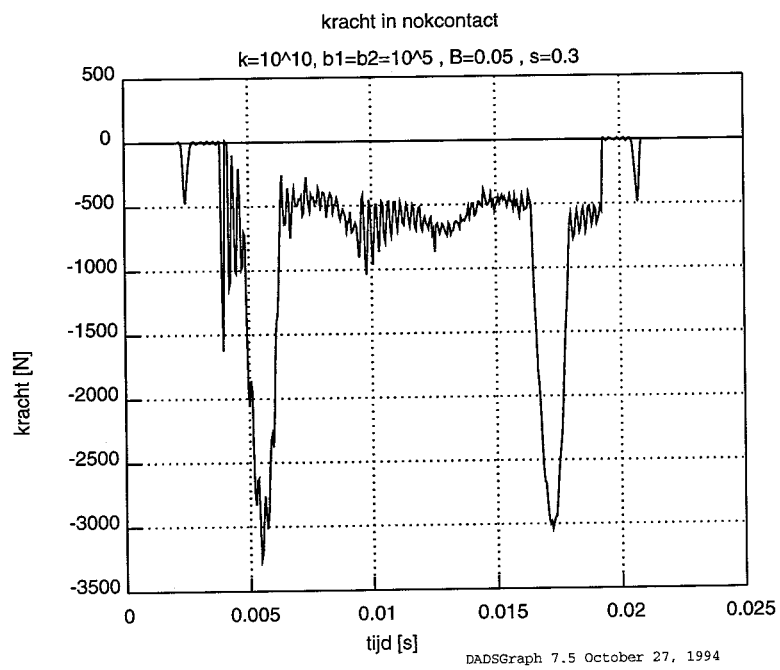
Figuur C.5:



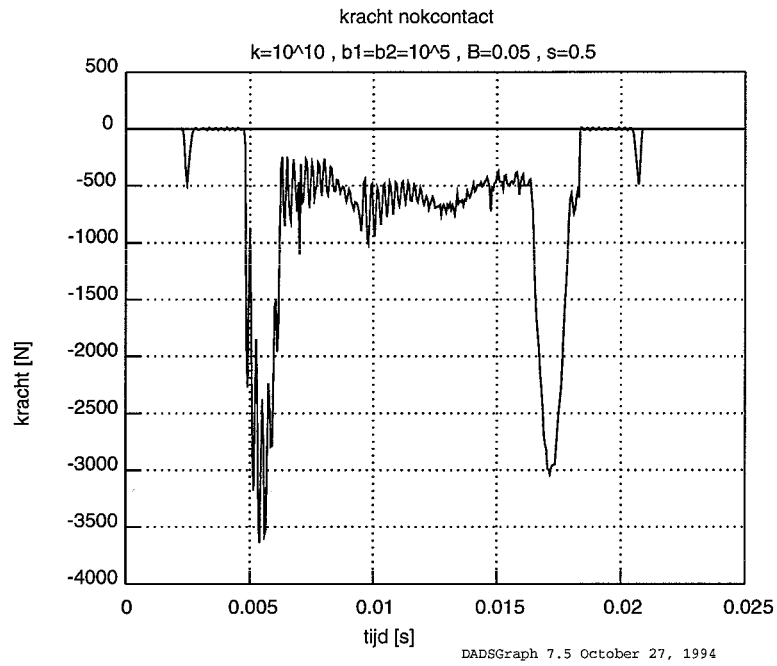
Figuur C.6:



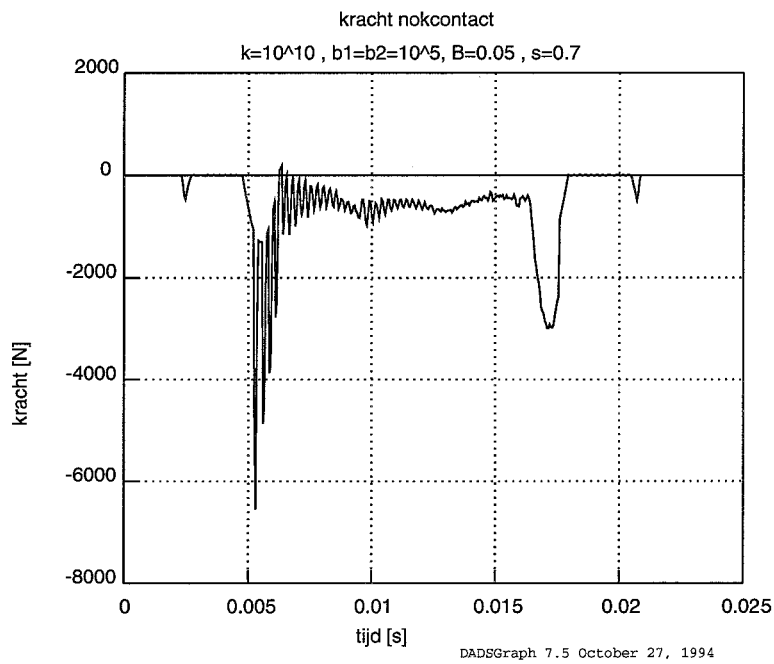
Figuur C.7:



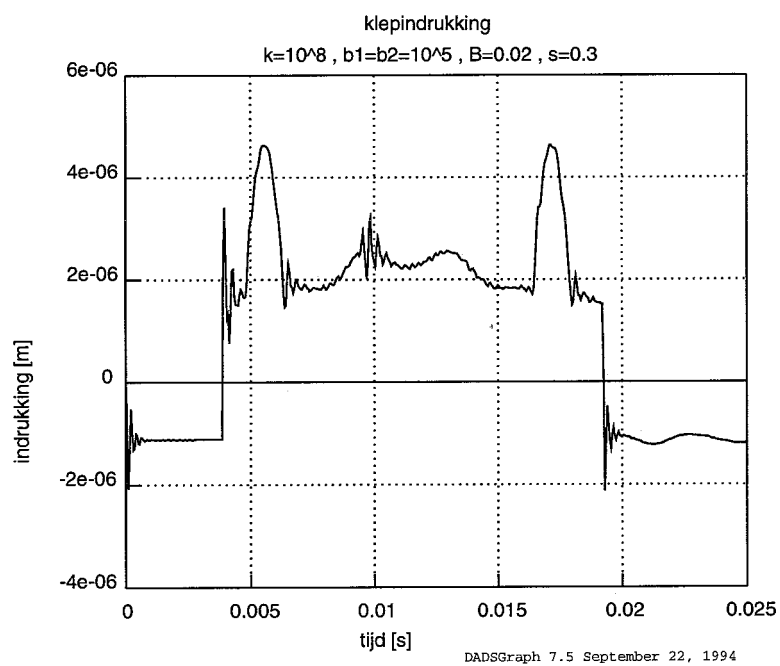
Figuur C.8:



Figuur C.9:



Figuur C.10:



Figuur C.11:

## D.1 Headers bij combinatie statische en normal modes

### D.1.1 Normal modes

```

$ Nastran invoerdeck om de normal modes van een RS-klep te berekenen;
$   Klepgeometrie in m ipv. mm (10 aug 1994)
$-----
$
$ De volgende regel toevoegen om de juiste output files te krijgen als
$ je via xnas draait.
ASSIGN OUTPUT2='RSKLEP1_NM_EIG.TAB' UNIT=91
ASSIGN OUTPUT4='RSKLEP1_NM_MAS.MAT' UNIT=92
$
ID DAF TRUCKS EINDHOVEN
TIME 1000.
DIAG=8,13,16,19,44,49
$
SOL 3
COMPILE DMAP=SOL3,SOUIN=MSCSOU,NOLIST,NOREF,NODECK
ALTER,24
OUTPUT2 GPL,BGPDT,,,/-1/91 $
ALTER,26
OUTPUT2 ECT,,,/0/91 $
ALTER,73
OUTPUT4 MGG,,,/-1/92 $
OUTPUT4 ,,,,/-2/92 $
OUTPUT2 OGPWG,,,/0/91 $
ALTER,405
OUTPUT2 LAMA,,,/0/91 $
ALTER,439
OUTPUT2 OUGV1,,,/0/91 $
OUTPUT2 ,,,,/-9/91 $
$ MATPRT OUGV1// $
CEND
$
$-----CASE CONTROL DECK
TITLE=NORMAL MODES DYNRSKLEP3D
SUBTITLE=1 SEPT 1994
$-----
$ Oplossing dynamische vergelijkingen mbv. Lanczos.
$-----
METHOD=10
$DYNRED=25
$
SUBCASE=1
  LABEL=RS-KLEP 3D
$-----
$ Output plot-file (OUT2) met modes
$-----
DISP(SORT1)=ALL
$

```

```

BEGIN BULK
PARAM,COUPMASS,-1          $ NODIG TBV. DADS!
$PARAM,K6ROT,1.0
PARAM,AUTOSPC,YES
PARAM,GRDPNT,0           $ NODIG TBV. DADS!
$PARAM,POST,-1
$-----
$ Default-hernummering aangehouden (optie 3), eerder per ongeluk geen her-
$ nummering (optie -1).
$-----
PARAM,NEWSEQ,3
$-----
$ Efficiëntere omgang met scratch-ruimte.
$-----
$PARAM,DBALL,SCRATCH
$
$-----
$ Definitie frequentiebereik voor eigentrillingsberekening.
$-----
$23456781234567812345678123456781234567812345678123456781234567812345678
EIGR      10      MGIV      0.          25          +E      1X
+E      1X      MAX
$EIGRL,10,1.,2500.,10
$
$ASET1  0      9001      THRU      9100
$QSET1  0      9001      THRU      9100
$$POINT 9001      THRU      9100
$DYNRED 25      25000.
$
$*****
$ Bulk data file
$*****

```

## D.1.2 statische modes

\$ De volgende regel toevoegen om de juiste output files te krijgen als je via xnas draait.

```

ASSIGN OUTPUT2='RSKLEP_ST_STAT.TAB' UNIT=93
ASSIGN OUTPUT4='RSKLEP_ST_STIF.MAT' UNIT=95
$
ID DAF TRUCKS EINDHOVEN
TIME 1000.
DIAG=8,13,16,19,49
$
SOL 24
COMPILE DMAP=SOL24,SOUIN=MSCSOU,NOLIST,NOREF,NODECK
ALTER,99
OUTPUT4 KGG,,,,//-1/95 $
OUTPUT4 ,,,,,//-2/95 $
ALTER,176
OUTPUT2 OUGV1,,,,//-1/93 $
OUTPUT2 ,,,,,//-9/93 $
CEND
$
$-----CASE CONTROL DECK
TITLE=STATIC MODES DYNRSKLEP3D
SUBTITLE=1 SEPT 1994
$-----
$ Oplossing dynamische vergelijkingen mbv. Lanczos.
$-----
$METHOD=10
$-----
$ Output plot-file (OUT2) met modes

```

```

$-----
DISP (SORT1) =ALL
$
SUBCASE=1
  SUBTITLE=STATIC MODES
  LABEL=RS-KLEP 3D
  LOAD=101
  SPC=502
$
BEGIN BULK
PARAM, COUPMASS, -1
$PARAM, K6ROT, 1.0
PARAM, AUTOSPC, YES
PARAM, GRDPNT, 0
$-----
$ Default-hernummering aangehouden (optie 3), eerder per ongeluk geen her-
$ nummering (optie -1).
$-----
PARAM, NEWSEQ, 3
$-----
$ Efficiëntere omgang met scratch-ruimte.
$-----
$PARAM, DBALL, SCRATCH
$
$-----
$ Definitie frequentiebereik voor eigentrillingsberekening.
$-----
$EIGRL, 10, 1., 2500., 25
$
$*****
$ Bulk data file
$*****
$
$
$
$
$Einde Nastran File
$Aangebrachte static load en
$onderdrukking vrijheidsgraden van een aantal nodes
FORCE      101      1679              1.0      0.0      1.0      0.0
$
SPC1       502        2      1222      1223      1249      1167      1168      1169+D      2A
+D      2A      1195      1196
SPC1       502        13      1589      1603      1738      1752      1591      1605
$
PSOLID     20        1
MAT1       1 21.E+10              0.3000 7200.00      0.
ENDDATA

```

## D.2 Alleen de normal modes meegenomen

```

$ Nastran invoerdeck om de normal modes van een RS-klep te berekenen;
$   Klepgeometrie in m ipv. mm (10 aug 1994)
$-----
$
$ De volgende regel toevoegen om de juiste output files te krijgen als
$ je via xnas draait.
ASSIGN OUTPUT2='RSKLEP1_NM_EIG.TAB' UNIT=91
ASSIGN OUTPUT4='RSKLEP1_NM_MAS.MAT' UNIT=92
ASSIGN OUTPUT4='RSKLEP1_NM_STIF.MAT' UNIT=93
$

```

```

ID DAF TRUCKS EINDHOVEN
TIME 1000.
DIAG=8,13,16,19,44,49
$
SOL 3
COMPILE DMAP=SOL3,SOUIN=MSCSOU,NOLIST,NOREF,NODECK
ALTER,24
OUTPUT2 GPL,BGPDT,,,// -1/91 $
ALTER,26
OUTPUT2 ECT,,,//0/91 $
ALTER,73
OUTPUT4 MGG,,,// -1/92 $
OUTPUT4,,,// -2/92 $
OUTPUT2 OGPWG,,,//0/91 $
ALTER,83
OUTPUT4 KGG,,,// -1/93 $
OUTPUT4,,,// -2/93 $
ALTER,405
OUTPUT2 LAMA,,,//0/91 $
ALTER,439
OUTPUT2 OUGV1,,,//0/91 $
OUTPUT2,,,// -9/91 $
$ MATPRT OUGV1// $
CEND
$
$-----CASE CONTROL DECK
TITLE=NORMAL MODES DYNRSKLEP3D
SUBTITLE=1 SEPT 1994
$-----
$ Oplossing dynamische vergelijkingen mbv. Lanczos.
$-----
METHOD=10
$DYNRED=25
$
SUBCASE=1
  LABEL=RS-KLEP 3D
$-----
$ Output plot-file (OUT2) met modes
$-----
DISP(SORT1)=ALL
$
BEGIN BULK
PARAM,COUPMASS,-1          $ NODIG TBV. DADS!
$PARAM,K6ROT,1.0
PARAM,AUTOSPC,YES
PARAM,GRDPNT,0            $ NODIG TBV. DADS!
$PARAM,POST,-1
$-----
$ Default-hernummering aangehouden (optie 3), eerder per ongeluk geen her-
$ nummering (optie -1).
$-----
PARAM,NEWSEQ,3
$-----
$ Efficiëntere omgang met scratch-ruimte.
$-----
$PARAM,DBALL,SCRATCH
$
$-----
$ Definitie frequentiebereik voor eigentrillingsberekening.
$-----
$2345678123456781234567812345678123456781234567812345678123456781234567812345678
EIGR      10      MGIV      0.          25          +E      1X
+E      1X      MAX
$EIGRL,10,1.,2500.,10
$

```



```
$ASET1  0      9001  THRU  9100
$QSET1  0      9001  THRU  9100
$SPOINT 9001   THRU  9100
$DYNRED 25     25000.
```

```
$
```

```
$*****
$ Bulk data file
$*****
```

Relevante gedeeltes uit de Dads definition file van het model met starre klep.

```

SPATIALDIMENSION = 2;
system.data "system.data" {
  ANALYSIS.TYPE           := "DYNAMIC";
  UNITS                   := "SI";
  OUTPUT.FILE             := "BINARY_ONLY";
  STARTING.TIME           := "0.0";
  ENDING.TIME             := "0.025";
  PRINT.INTERVAL          := "0.00005";
  REFERENCE.FRAME        := "NCBF";
  GRAVITY.SEA.LEVEL      := "9.80665";
  SCALE.GRAVITY.COEF     := "0.0";
  X.GRAVITY               := "0.0";
  Y.GRAVITY               := "1.0";
  MATRIX.OPERATIONS      := "SPARSE";
  REDUNDANCY.CHECK       := "TRUE";
  LU.TOL                  := "1.0E-12";
  ASSEMBLY.TOL           := "1.0E-5";
  BYPASS.ASSEMBLY        := "FALSE";
  DEBUG.FLAG              := "TRUE";
  WRITE.REDUNDANT        := "FALSE";
};
dynamic.data "dynamic.data" {
  REACTION.FORCES        := "TRUE";
  FORCE.COORDINATES       := "GLOBAL";
  PRINT.METHOD          := "INTERPOLATED";
  MAX.INT.STEP           := "0.000001";
  SOLUTION.TOL           := "0.00001";
  INTEGRATION.TOL        := "0.00001";
  METHOD.INTEGRATION      := "VARIABLE";
  PRINT.FREQ              := "0";
};
.....
.....
};
position.constraint "veerconstraint1" {
  NAME                   := "veerconstraint1";
  BODY.NAME              := "veer1";
  TRIAD.ON.BODY          := "veertriad1";
  TYPE.CONSTRAINT        := "X";
  CONSTRAINT.VALUE       := "50e-3";
  ANGULAR.UNITS          := "DEGREES";
};
position.constraint "veerconstraint2" {
  NAME                   := "veerconstraint2";
  BODY.NAME              := "veer2";
  TRIAD.ON.BODY          := "veertriad2";
  TYPE.CONSTRAINT        := "X";
  CONSTRAINT.VALUE       := "50e-3";
  ANGULAR.UNITS          := "DEGREES";
};
position.constraint "veerconstraint3" {

```

```

NAME                := "veerconstraint3";
BODY.NAME           := "veer3";
TRIAD.ON.BODY       := "veertriad3";
TYPE.CONSTRAINT     := "X";
CONSTRAINT.VALUE    := "50e-3";
ANGULAR.UNITS       := "DEGREES";
};
position.constraint "stoterconstr" {
NAME                := "stoterconstr";
BODY.NAME           := "stoter";
TRIAD.ON.BODY       := "nokstoter";
TYPE.CONSTRAINT     := "X";
CONSTRAINT.VALUE    := "135e-3";
ANGULAR.UNITS       := "DEGREES";
};
tsda "veertje1" {
NAME                := "veertje1";
TYPE                := "BIDIRECTIONAL";
BODY.1.NAME         := "blok";
BODY.2.NAME         := "veer1";
TRIAD.ON.BODY.1     := "veerblok";
TRIAD.ON.BODY.2     := "veertriad1";
SPRING.CONSTANT     := "138e3";
DAMPING.COEFFICIENT := "5";
ACTUATOR.FORCE      := "0.0";
CURVE.SPRING        := "NONE";
CURVE.DAMPER        := "NONE";
CURVE.ACTUATOR      := "NONE";
NODE.1              := "0";
NODE.2              := "0";
FREE.LENGTH.SPRING  := "15.6e-3";
};
tsda "veertje2" {
NAME                := "veertje2";
TYPE                := "BIDIRECTIONAL";
BODY.1.NAME         := "veer1";
BODY.2.NAME         := "veer2";
TRIAD.ON.BODY.1     := "veertriad1";
TRIAD.ON.BODY.2     := "veertriad2";
SPRING.CONSTANT     := "138e3";
DAMPING.COEFFICIENT := "5";
ACTUATOR.FORCE      := "0.0";
CURVE.SPRING        := "NONE";
CURVE.DAMPER        := "NONE";
CURVE.ACTUATOR      := "NONE";
NODE.1              := "0";
NODE.2              := "0";
FREE.LENGTH.SPRING  := "15.6e-3";
};
tsda "veertje3" {
NAME                := "veertje3";
TYPE                := "BIDIRECTIONAL";
BODY.1.NAME         := "veer2";
BODY.2.NAME         := "veer3";
TRIAD.ON.BODY.1     := "veertriad2";
TRIAD.ON.BODY.2     := "veertriad3";
SPRING.CONSTANT     := "138e3";
DAMPING.COEFFICIENT := "5";
ACTUATOR.FORCE      := "0.0";
CURVE.SPRING        := "NONE";
CURVE.DAMPER        := "NONE";
CURVE.ACTUATOR      := "NONE";
NODE.1              := "0";
NODE.2              := "0";
FREE.LENGTH.SPRING  := "15.6e-3";
};

```

```

};
tsda "veertje4" {
  NAME                := "veertje4";
  TYPE                := "BIDIRECTIONAL";
  BODY.1.NAME         := "veer3";
  BODY.2.NAME         := "klep";
  TRIAD.ON.BODY.1     := "veertriad3";
  TRIAD.ON.BODY.2     := "veerklep";
  SPRING.CONSTANT     := "138e3";
  DAMPING.COEFFICIENT := "5";
  ACTUATOR.FORCE      := "0.0";
  CURVE.SPRING        := "NONE";
  CURVE.DAMPER        := "NONE";
  CURVE.ACTUATOR      := "NONE";
  NODE.1              := "0";
  NODE.2              := "0";
  FREE.LENGTH.SPRING := "15.6e-3";
};
point.follower "nokmechanisme" {
  NAME                := "nokmechanisme";
  TRACKED.BODY.NAME   := "nok";
  PINNED.BODY.NAME    := "stoter";
  TRIAD.ON.TRACK.BODY := "bloknok";
  TRIAD.ON.PINNED.BODY := "nokstoter";
  CURVE.TRACK         := "nokfunctie2";
  INIT.CONTACT.ANGLE := "pi";
  ANGULAR.UNITS       := "RADIANS";
};
revolute.joint "tuimelaaras" {
  NAME                := "tuimelaaras";
  BODY.1.NAME         := "blok";
  BODY.2.NAME         := "tuimelaar";
  TRIAD.ON.BODY.1     := "tuimelaarblok";
  TRIAD.ON.BODY.2     := "bloktuimelaar";
  NODE.1              := "0";
  NODE.2              := "0";
};
revolute.joint "nokkenas" {
  NAME                := "nokkenas";
  BODY.1.NAME         := "blok";
  BODY.2.NAME         := "nok";
  TRIAD.ON.BODY.1     := "nokkenasblok";
  TRIAD.ON.BODY.2     := "nokaansturing";
  NODE.1              := "0";
  NODE.2              := "0";
};
revolute.joint "stotertuimelrev" {
  NAME                := "stotertuimelrev";
  BODY.1.NAME         := "tuimelaar";
  BODY.2.NAME         := "stoter";
  TRIAD.ON.BODY.1     := "stotertuimelaar";
  TRIAD.ON.BODY.2     := "tuimelaarstoter";
  NODE.1              := "0";
  NODE.2              := "0";
};
translational.joint "klepgeleiding" {
  NAME                := "klepgeleiding";
  BODY.1.NAME         := "blok";
  BODY.2.NAME         := "klep";
  TRIAD.ON.BODY.1     := "geleidingblok";
  TRIAD.ON.BODY.2     := "geleidingklep";
  NODE.1              := "0";
  NODE.2              := "0";
};
contact "klepzitting" {

```

```

OPTION                := "COMPRESSION";
CURVE.PROFILE2       := "NONE";
NAME                  := "klepzitting";
BODY.1.NAME           := "blok";
BODY.2.NAME           := "klep";
TRIAD.ON.BODY.1      := "zittingblok";
TRIAD.ON.BODY.2      := "zittingklep";
TYPE                  := "POINT-POINT";
CURVE.PROFILE1       := "NONE";
SPRING.CONSTANT      := "1e8";
DAMPING.CONSTANT     := "1e3";
MU1.COEFF             := "0.0";
CURVE.SPRING         := "NONE";
CURVE.DAMPER         := "NONE";
MU2.COEFF             := "1.0";
PARAMETER.1          := "10e-3";
PARAMETER.2          := "10e-3";
DPARAMETER.1         := "0";
DPARAMETER.2         := "0";
ANGULAR.UNITS        := "DEGREES";
};
contact "kleptuimelaarcont" {
  OPTION                := "COMPRESSION";
  CURVE.PROFILE2       := "NONE";
  NAME                  := "kleptuimelaarcont";
  BODY.1.NAME           := "klep";
  BODY.2.NAME           := "tuimelaar";
  TRIAD.ON.BODY.1      := "kleptop";
  TRIAD.ON.BODY.2      := "kleptuimelaar";
  TYPE                  := "POINT-POINT";
  CURVE.PROFILE1       := "NONE";
  SPRING.CONSTANT      := "1e8";
  DAMPING.CONSTANT     := "1e4";
  MU1.COEFF             := "0.0";
  CURVE.SPRING         := "NONE";
  CURVE.DAMPER         := "NONE";
  MU2.COEFF             := "1.0";
  PARAMETER.1          := "10e-3";
  PARAMETER.2          := "10e-3";
  DPARAMETER.1         := "0";
  DPARAMETER.2         := "0";
  ANGULAR.UNITS        := "DEGREES";
};
initial.condition "start" {
  NAME                  := "start";
  TYPE.INITIAL.COND    := "Y";
  INITIAL.VALUE        := "0";
  TIME.DERIVATIVE      := "0.0";
  BODY.1.NAME           := "stoter";
  BODY.2.NAME           := "NONE";
  TRIAD.ON.BODY.1      := "geleidingstoter";
  TRIAD.ON.BODY.2      := "NONE";
  EXTRA.COORD          := "0";
  ANGULAR.UNITS        := "DEGREES";
};
driver "aansturing" {
  NAME                  := "aansturing";
  TYPE.DRIVER           := "PHI.DIFF";
  BODY.1.NAME           := "blok";
  BODY.2.NAME           := "nok";
  TRIAD.ON.BODY.1      := "nokkenashblok";
  TRIAD.ON.BODY.2      := "nokaansturing";
  DRIVING.FUNCTION      := "POLYNOMIAL";
  PARAMETER.1          := "-pi/2";
  PARAMETER.2          := "24*2*pi";
};

```

```

PARAMETER.3           := "0.0";
PARAMETER.4           := "0.0";
CURVE.DRIVER          := "nokfunctie";
JOINT.NAME            := "NONE";
NODE.NAME             := "NONE";
ANGULAR.UNITS         := "RADIANS";
};
curve "nokfunctie2" {
  NAME                 := "nokfunctie2";
  TYPE.DATA            := "SPLINE.INCREMENTAL";
  DATA                := "@nokfunc-reversed.txt ";
  CYCLIC               := "TRUE";
  NUMBER.ROWS          := "1";
  GRID                 := "2";
  SLOPE.LEFT           := "0.0";
  SLOPE.RIGHT          := "0.0";
  INTERPOLATION        := "CUBIC";
  SCALE.X              := "pi/180";
  SCALE.Y              := "1.0";
  SCALE.Z              := "1e-3";
  START.X              := "0";
  START.Y              := "0.0";
  OFFSET.Z             := "20";
  INCREMENT.X          := "1";
  INCREMENT.Y          := "0.0";
  REVERSE.NORMALS     := "FALSE";
};
.....
.....
JOINTTRIAD "zittingblok" {
  /* parent: blok */
  .....
  method = EULER;
  Name                 := "zittingblok";
  X.P                  := "50e-3";
  Y.P                  := "10e-3";
  PHI                  := "0.0";
  ANGULAR.UNITS        := "DEGREES";
};

JOINTTRIAD "geleidingblok" {
  /* parent: blok */
  .....
  method = EULER;
  Name                 := "geleidingblok";
  X.P                  := "0.05";
  Y.P                  := "0.049563";
  PHI                  := "90.0";
  ANGULAR.UNITS        := "DEGREES";
};

JOINTTRIAD "veerblok" {
  /* parent: blok */
  .....
  method = EULER;
  Name                 := "veerblok";
  X.P                  := "50e-3";
  Y.P                  := "100e-3";
  PHI                  := "0.0";
  ANGULAR.UNITS        := "DEGREES";
};

JOINTTRIAD "tuimelaarblok" {
  /* parent: blok */

```

```

.....
method = EULER;
Name           := "tuimelaarblok";
X.P           := "100e-3";
Y.P           := "155e-3";
PHI           := "0.0";
ANGULAR.UNITS := "DEGREES";
};

JOINTTRIAD "geleidingblok2" {
  /* parent: blok */
  .....
  method = EULER;
  Name           := "geleidingblok2";
  X.P           := "135e-3";
  Y.P           := "0";
  PHI           := "90";
  ANGULAR.UNITS := "DEGREES";
};

JOINTTRIAD "nokkenasblok" {
  /* parent: blok */
  .....
  method = EULER;
  Name           := "nokkenasblok";
  X.P           := "0.135";
  Y.P           := "-0.25";
  PHI           := "0.0";
  ANGULAR.UNITS := "DEGREES";
};

BODY "klep" {
  /* parent: world */
  .....
  method = EULER;
  X.P           := "50e-3";
  Y.P           := "0.0";
  PHI           := "0.0";
  ANGULAR.UNITS := "DEGREES";
  MASS          := "0.3";
  INERTIA       := "4.84e-3";
  XG.FORCE     := "0.0";
  YG.FORCE     := "0.0";
  TORQUE.CONSTANT := "0.0";
  CURVE.XGF    := "NONE";
  CURVE.YGF    := "NONE";
  CURVE.TORQUE := "NONE";
  OUTLINE.SHAPE := "NONE";
  X.S          := "0.0";
  Y.S          := "0.0";
  FIXED.TO.GROUND := "FALSE";
  SUPERELEMENT := "FALSE";
  FLEXIBLE     := "FALSE";
}; /* klep */

CGTRIAD "klep_cg" {
  /* parent: klep */
  .....
  method = EULER;
  X.P           := "0";
  Y.P           := "72.5e-3";
  PHI           := "0.0";
  ANGULAR.UNITS := "DEGREES";
};

```

```

JOINTTRIAD "zittingklep" {
  /* parent: klep */
  VISIBILITY = null;
  LABEL = disable;
  PRELABEL = ;
  LABELVAR = ;
  UNITS = ;
  LABELPOS = 0.0 0.0 0.0;
  LABELCOLOR = 0.0 1.0 1.0;
  TRIAD = disable;
  TRIADCOLOR = 0 1 1;
  COLORTABLE = "";
  CHILD = ;
  method = EULER;
  Name                := "zittingklep";
  X.P                 := "0";
  Y.P                 := "-10e-3";
  PHI                 := "90";
  ANGULAR.UNITS      := "DEGREES";
};
.....
JOINTTRIAD "geleidingklep" {
  /* parent: klep */
  .....
  method = EULER;
  Name                := "geleidingklep";
  X.P                 := "0.0";
  Y.P                 := "0.049563";
  PHI                 := "90.0";
  ANGULAR.UNITS      := "DEGREES";
};

JOINTTRIAD "veerklep" {
  /* parent: klep */
  .....
  method = EULER;
  Name                := "veerklep";
  X.P                 := "0.0";
  Y.P                 := "150e-3";
  PHI                 := "0.0";
  ANGULAR.UNITS      := "DEGREES";
};

JOINTTRIAD "kleptop" {
  /* parent: world */
  .....
  method = EULER;
  Name                := "kleptop";
  X.P                 := "0.0";
  Y.P                 := "145e-3";
  PHI                 := "0";
  ANGULAR.UNITS      := "DEGREES";
};

BODY "tuimelaar" {
  .....
  method = EULER;
  X.P                 := "100e-3";
  Y.P                 := "155e-3";
  PHI                 := "0.0";
  ANGULAR.UNITS      := "DEGREES";
  MASS                := "0.238";
  INERTIA              := "2.1e-4";
  XG.FORCE            := "0.0";
};

```



```

YG.FORCE                := "0.0";
TORQUE.CONSTANT         := "0.0";
CURVE.XGF               := "NONE";
CURVE.YGF               := "NONE";
CURVE.TORQUE           := "NONE";
OUTLINE.SHAPE          := "NONE";
X.S                     := "0.0";
Y.S                     := "0.0";
FIXED.TO.GROUND        := "FALSE";
SUPERELEMENT           := "FALSE";
FLEXIBLE                := "FALSE";
}; /* tuimelaar */

CGTRIAD "tuimelaar_cg" {
  /* parent: tuimelaar */
  .....
  method = EULER;
  X.P                := "0";
  Y.P                := "0";
  PHI                := "0.0";
  ANGULAR.UNITS     := "DEGREES";
};

JOINTTRIAD "bloktuimelaar" {
  /* parent: tuimelaar */
  .....
  method = EULER;
  Name                := "bloktuimelaar";
  X.P                := "0.0";
  Y.P                := "0.0";
  PHI                := "0.0";
  ANGULAR.UNITS     := "DEGREES";
};

JOINTTRIAD "kleptuimelaar" {
  /* parent: tuimelaar */
  .....
  method = EULER;
  Name                := "kleptuimelaar";
  X.P                := "-50e-3";
  Y.P                := "10.3e-3";
  PHI                := "90";
  ANGULAR.UNITS     := "DEGREES";
};

JOINTTRIAD "stotertuimelaar" {
  /* parent: tuimelaar */
  .....
  method = EULER;
  Name                := "stotertuimelaar";
  X.P                := "35e-3";
  Y.P                := "0";
  PHI                := "90";
  ANGULAR.UNITS     := "DEGREES";
};

BODY "stoter" {
  /* parent: world */
  .....
  method = EULER;
  X.P                := "135e-3";
  Y.P                := "0";
  PHI                := "0.0";
  ANGULAR.UNITS     := "DEGREES";
  MASS               := "0.432";
};

```

```

INERTIA                := "5.8e-4";
XG.FORCE               := "0.0";
YG.FORCE               := "0.0";
TORQUE.CONSTANT        := "0.0";
CURVE.XGF              := "NONE";
CURVE.YGF              := "NONE";
CURVE.TORQUE           := "NONE";
OUTLINE.SHAPE          := "NONE";
X.S                    := "0.0";
Y.S                    := "0.0";
FIXED.TO.GROUND        := "FALSE";
SUPERELEMENT           := "FALSE";
FLEXIBLE                := "FALSE";
}; /* stoter */

CGTRIAD "stoter_cg" {
  /* parent: stoter */
  .....
  method = EULER;
  X.P                := "0";
  Y.P                := "-37.5e-3";
  PHI                := "0.0";
  ANGULAR.UNITS      := "DEGREES";
};

JOINTTRIAD "geleidingstoter" {
  /* parent: stoter */
  VISIBILITY = null;
  LABEL = disable;
  PRELABEL = ;
  LABELVAR = ;
  UNITS = ;
  LABELPOS = 0.0 0.0 0.0;
  LABELCOLOR = 0.0 1.0 1.0;
  TRIAD = disable;
  TRIADCOLOR = 0 1 1;
  COLORTABLE = "";
  CHILD =;
  method = EULER;
  Name                := "geleidingstoter";
  X.P                := "0.0";
  Y.P                := "0.0";
  PHI                := "90";
  ANGULAR.UNITS      := "DEGREES";
};

JOINTTRIAD "nokstoter" {
  /* parent: stoter */
  .....
  method = EULER;
  Name                := "nokstoter";
  X.P                := "0.0";
  Y.P                := "-0.23";
  PHI                := "0.0";
  ANGULAR.UNITS      := "DEGREES";
};

JOINTTRIAD "tuimelaarstoter" {
  /* parent: stoter */
  .....
  method = EULER;
  Name                := "tuimelaarstoter";
  X.P                := "0";
  Y.P                := "155e-3";
};

```

```

    PHI                                     := "0";
    ANGULAR.UNITS                           := "DEGREES";
};

BODY "nok" {
  /* parent: world */
  .....
  method = EULER;
  X.P                                       := "135e-3";
  Y.P                                       := "-250e-3";
  PHI                                       := "0";
  ANGULAR.UNITS                             := "DEGREES";
  MASS                                      := "0.17";
  INERTIA                                   := "3.9e-5";
  XG.FORCE                                  := "0.0";
  YG.FORCE                                  := "0.0";
  TORQUE.CONSTANT                           := "0.0";
  CURVE.XGF                                 := "NONE";
  CURVE.YGF                                 := "NONE";
  CURVE.TORQUE                              := "NONE";
  OUTLINE.SHAPE                             := "nokfunctie2";
  X.S                                        := "0.0";
  Y.S                                        := "0.0";
  FIXED.TO.GROUND                           := "FALSE";
  SUPERELEMENT                              := "FALSE";
  FLEXIBLE                                   := "FALSE";
}; /* nok */

CGTRIAD "nok_cg" {
  /* parent: nok */
  .....
  method = EULER;
  X.P                                       := "0.0";
  Y.P                                       := "0.0";
  PHI                                       := "0.0";
  ANGULAR.UNITS                             := "DEGREES";
};

JOINTTRIAD "bloknok" {
  /* parent: nok */
  .....
  method = EULER;
  Name                                      := "bloknok";
  X.P                                       := "0.0";
  Y.P                                       := "0.0";
  PHI                                       := "0.0";
  ANGULAR.UNITS                             := "DEGREES";
};

JOINTTRIAD "nokaansturing" {
  /* parent: nok */
  .....
  method = EULER;
  Name                                      := "nokaansturing";
  X.P                                       := "0.0";
  Y.P                                       := "0.0";
  PHI                                       := "0";
  ANGULAR.UNITS                             := "DEGREES";
};

BODY "veer1" {
  /* parent: veer1 */
  .....

```

```

method = EULER;
X.P           := "50e-3";
Y.P           := "112.5e-3";
PHI           := "0.0";
ANGULAR.UNITS := "DEGREES";
MASS          := "23.4e-3";
INERTIA       := "1.0";
XG.FORCE      := "0.0";
YG.FORCE      := "0.0";
TORQUE.CONSTANT := "0.0";
CURVE.XGF     := "NONE";
CURVE.YGF     := "NONE";
CURVE.TORQUE  := "NONE";
OUTLINE.SHAPE := "NONE";
X.S           := "0.0";
Y.S           := "0.0";
FIXED.TO.GROUND := "FALSE";
SUPERELEMENT  := "FALSE";
FLEXIBLE      := "FALSE";
}; /* veer1 */

CGTRIAD "veer1_cg" {
  /* parent: veer1 */
  .....
  method = EULER;
  X.P           := "0.0";
  Y.P           := "0.0";
  PHI           := "0.0";
  ANGULAR.UNITS := "DEGREES";
};

BODY "veer2" {
  /* parent: world */
  .....
  method = EULER;
  X.P           := "50e-3";
  Y.P           := "125e-3";
  PHI           := "0.0";
  ANGULAR.UNITS := "DEGREES";
  MASS          := "23.4e-3";
  INERTIA       := "1.0";
  XG.FORCE      := "0.0";
  YG.FORCE      := "0.0";
  TORQUE.CONSTANT := "0.0";
  CURVE.XGF     := "NONE";
  CURVE.YGF     := "NONE";
  CURVE.TORQUE  := "NONE";
  OUTLINE.SHAPE := "NONE";
  X.S           := "0.0";
  Y.S           := "0.0";
  FIXED.TO.GROUND := "FALSE";
  SUPERELEMENT  := "FALSE";
  FLEXIBLE      := "FALSE";
}; /* veer2 */

CGTRIAD "veer2_cg" {
  /* parent: veer2 */
  .....
  method = EULER;
  X.P           := "0.0";
  Y.P           := "0.0";
  PHI           := "0.0";
  ANGULAR.UNITS := "DEGREES";
};

```

```

BODY "veer3" {
  /* parent: world */
  .....
  method = EULER;
  X.P           := "50e-3";
  Y.P           := "137.5e-3";
  PHI           := "0.0";
  ANGULAR.UNITS := "DEGREES";
  MASS          := "23.4e-3";
  INERTIA       := "1.0";
  XG.FORCE      := "0.0";
  YG.FORCE      := "0.0";
  TORQUE.CONSTANT := "0.0";
  CURVE.XGF     := "NONE";
  CURVE.YGF     := "NONE";
  CURVE.TORQUE  := "NONE";
  OUTLINE.SHAPE := "NONE";
  X.S           := "0.0";
  Y.S           := "0.0";
  FIXED.TO.GROUND := "FALSE";
  SUPERELEMENT  := "FALSE";
  FLEXIBLE      := "FALSE";
}; /* veer3 */

CGTRIAD "veer3_cg" {
  /* parent: veer3 */
  .....
  method = EULER;
  X.P           := "0.0";
  Y.P           := "0.0";
  PHI           := "0.0";
  ANGULAR.UNITS := "DEGREES";
};

JOINTTRIAD "veertriad1" {
  /* parent: veer1 */
  .....
  method = EULER;
  Name           := "veertriad1";
  X.P           := "0.0";
  Y.P           := "0.0";
  PHI           := "0.0";
  ANGULAR.UNITS := "DEGREES";
};

JOINTTRIAD "veertriad2" {
  /* parent: veer2 */
  .....
  method = EULER;
  Name           := "veertriad2";
  X.P           := "0.0";
  Y.P           := "0.0";
  PHI           := "0.0";
  ANGULAR.UNITS := "DEGREES";
};

JOINTTRIAD "veertriad3" {
  /* parent: veer3 */
  .....
  method = EULER;
  Name           := "veertriad3";
  X.P           := "0.0";

```

```
Y.P                := "0.0";  
PHI                := "0.0";  
ANGULAR.UNITS     := "DEGREES";  
};
```