Eindhoven University of Technology

MASTER

Development of a helicopter landing gear simulation model in DADS

den Engelsman, W.P.

Award date:
1992

Link to publication

DEVELOPMENT OF A HELICOPTER LANDING GEAR SIMULATION MODEL IN DADS

W.P. den Engelsman
Eindhoven
December 1992
2199-5900/92-1290

WFW 92-131

ABSTRACT

This report describes the modelling of the NH90 helicopter landing gear which is being developed at DAF Special Products. The model is generated in the multi body dynamics program DADS. The objective of this dynamic model is to optimize the design of the landing gear in order to meet the performance requirements.

In chapter 2 a description of the geometry of the landing gear is given. The landing gear is a tricycle type and is fully retractable. The shock absorbers installed are hydro pneumatic elements. To meet the higher energy absorbtion requirements for crashworthiness of one of the two versions of the NH90 a crash tube is added to the nose gear.

Modelling of the landing gear is discussed in chapter 3. The geometry is represented by a 2 dimensional model defined by DADS-3D. The behaviour of the shock absorbers and tires is defined in user defined elements. The crash tube is modelled with the use of a friction element, respectively a damper element. Both models are compared and have their disadvantages. In the case when a damper element is used an undesirable relaxation displacement occurs and a large CPU time is required. The performance of the friction element is highly sensitive for changes in the geometry of the model. However, due to the lack of time to generate an alternative model, a friction element is preferred to model the crash tube. To model the different landing cases, the model has to be positioned in different positions at the beginning of the simulations. Within DADS the initial positions and velocities can be defined with the use of initial condition elements. However, problems in the assembly process arise in case large initial displacements are defined. Therefore, a restart procedure is used to position the model. To define the initial velocities with the use of this procedure the restart file has to be edited manualy.

The generated model of the landing gear is adequate for use as a design tool and parameter studies can be employed if regard is paid to the restrictions in the performance of the friction element. To illustrate the performance of the landing gear model two different landing cases are discussed in chapter 4.

CONTENTS

APPENDICES

1 INTRODUCTION

Four European countries (Germany, Italy, France and the Netherlands) are working together to develop a new helicopter, named NH90. The objective of the NH90 programme is to build a new helicopter for the 90's with greatly improved operational performance compared to current helicopters. The organisations involved have proposed the use of advanced technologies to meet the high performance requirements. DAF Special Products is the company who is responsible for the development of the landing gear.

In order to develop a high performance landing gear, a dynamic simulation model of the landing gear is required in the design stage. This report documents the generation of a simulation model of the NH90 landing gear system. The objective of modelling the landing gears in this study is to optimize their design in order to meet the performance requirements, while conforming with the geometric, static and other constraints imposed upon them. The main performance requirements for the landing gear that have to be met consists of:

- Supporting the helicopter on the ground and minimizing the loads acting on the fuselage and landing gear during landing and ground manoeuvres.
- Assuring sufficient clearance between the ground and the fuselage during these manoeuvres.
- Crashworthiness of the landing gear.

To accomplish the crashworthiness particularly, advanced technologies are required.

To model the NH90 landing gear system the multi-body dynamic program DADS (Dynamic Analysis and Design System [3]) was employed. Van Slagmaat [13] successfully used this program to model an existing landing gear, and the model was verified by test results.

2. LANDING GEAR DESIGN

This chapter gives a global description of the geometry of the NH90 landing gear (paragraph 2.1). The form and behaviour of the shock absorbers is described in paragraph 2.2. Paragraph 2.3 describes the behaviour of the tire. Finally in paragraph 2.4 the behaviour of the crash tube employed on one of the two models of the NH90 is discussed.

2.1 Geometry

Figure 2.1a shows a general arrangement of the NH90 helicopter and its main dimensions (in millimeters).

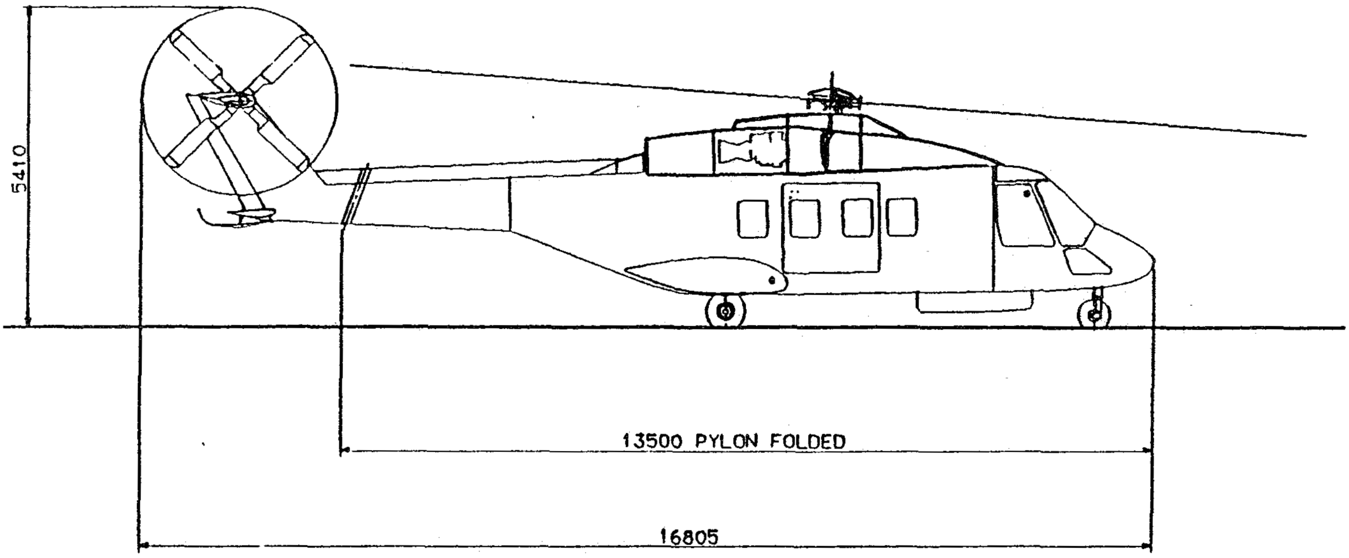


Figure 2.1a NH90 helicopter

To fullfill the demands of the armies, navies and airforces of the four corporating countries, the NH90 helicopter has to be built in two versions:

- a land based version, the Tactical Transport Helicopter (TTH)
- a naval version, the Nato Frigate Helicopter (NFH).

Both versions will feature a retractable tricycle landing gear. The nose gear is the twin wheel type and retracts rearwards. The main landing gear is the trailing arm type with independent shock absorbers and a single wheel at each leg. The retract actuator of the main gear is mounted above the shock absorber. The wheelbase between nose and main landing gear is 6.10 m. for the TTH and 6.18 m. for the NFH. The current design of the landing gear is shown in figure 2.1b.

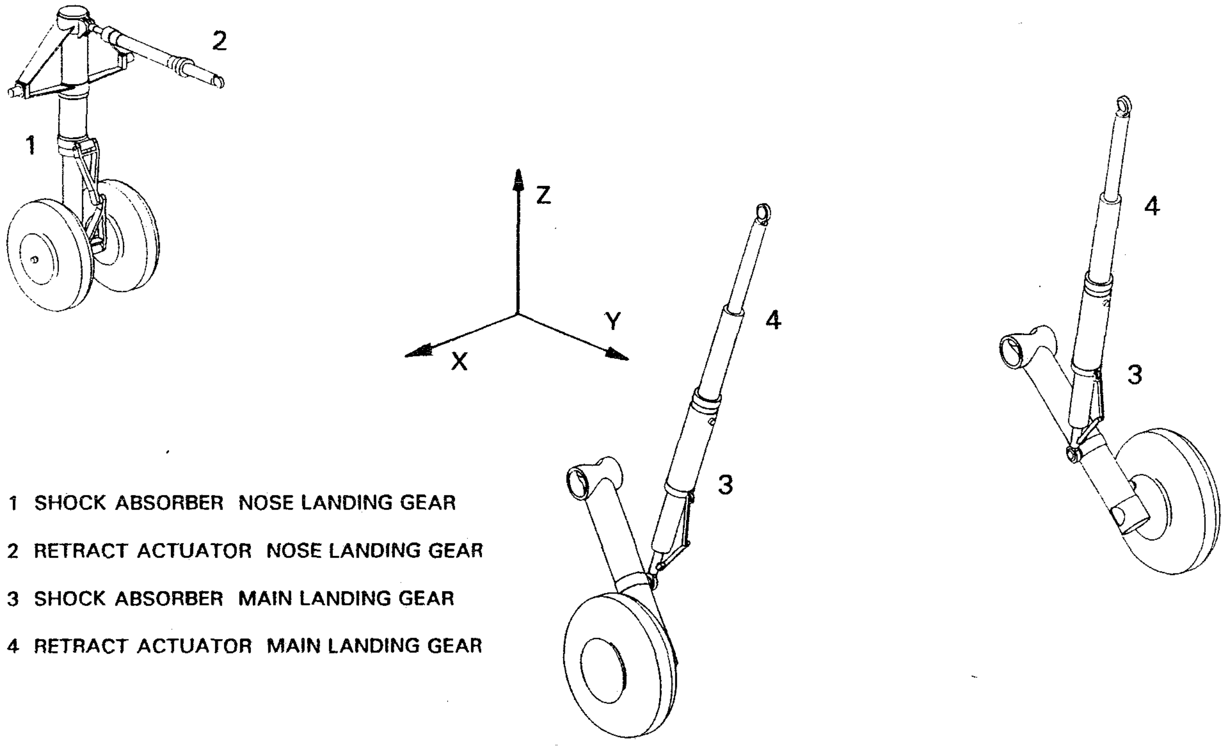


Figure 2.1b NH90 landing gear system

The design has been developed to achieve substantial commonality between the NFH and the TTH versions. Although these two versions have different requirements, only small differences exists.

The differences between the TTH and NFH landing gear are:

- NFH version has a steering system on the nose landing gear and no trail.
- TTH version has a free swiveling nose landing gear with a trail of 0.0762 meter to permit steering of the helicopter by differential braking of the main landing gear wheels and tail rotor thrust.
- the nose landing gear of the TTH version has a crash tube in order to make this landing gear crashworthy.

2.2 Shock absorber

The shock absorber is the most significant part of the main and nose landing gears. The shock absorber is a hydro-pneumatic spring damper element often called an oleo. Many types of oleos with diverse configurations exist. For a detailed survey of these configurations refer to Sijpkes [14] and Curry [2]. All these complex forms may be reduced however to a basic functional representation.

The basic shock absorber

Figure 2.2 explains the operation of the basic shock absorber.

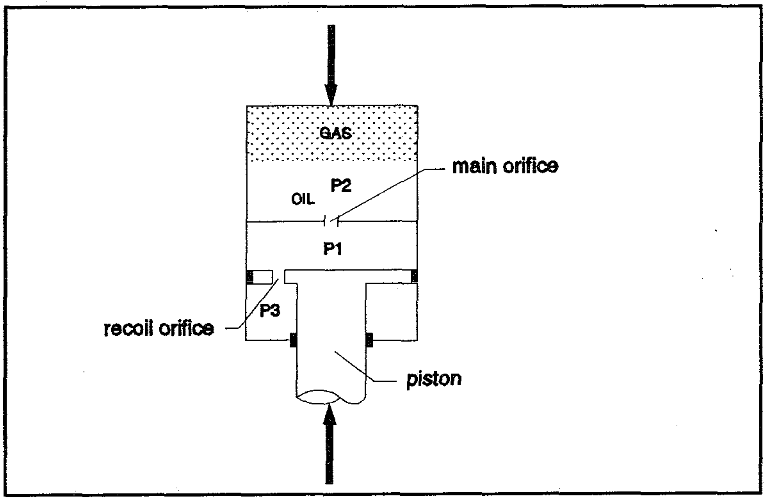


Figure 2.2 basic shock absorber

As the shock absorber is compressed the piston moves inwards and the oil is forced to flow through the main orifice to the upper chamber (P2) and through the recoil orifice to the recoil chamber (P3). Hydraulic damping is caused by the flow of the oil through the orifices and can be described by Bernoulli's law.

$$\Delta p = \frac{1}{C_d^2} \cdot \frac{Q_v^2}{A_{orf}^2} \cdot \frac{\rho}{2}$$

with:

Δp : pressure difference over the orifice
C_d : coefficient of discharge
A_{orf} : orifice area

ρ : density of the oil

Q_v : the flow through the orifice

As a result of the flow through the main orifice, the gas volume in the upper chamber will be compressed. The compression of the gas can be described with polytropic gas law:

$$P_{ext} \cdot V_{ext}^{n} = P_2 \cdot V_2^{n}$$

Where P2 and V2 represent respectively the current gas pressure and volume and P_{ext} and V_{ext} represent respectively the shock absorber fully extended gas pressure and volume. The polytropic gas exponent is denoted by n. The theoretical value of the polytropic exponent is bounded by the isothermal (n = 1) and the adiabatic exponent (n = 1.4). According to Greer Maclain and Grenshaw [5], if a mixture of gas and oil occurs the value of the exponent tends towards the isothermal as a result of the improved cooling. If however, a separator between the oil and gas is used, the compression of the gas can be described as almost adiabatic.

These two physical laws are used to relate the compression of the shock absorber and the resulting pressures. The shock absorber force is calculated by multiplying the relevant. pressures and the corresponding areas. A more detailed description of the mathematical model is given by Sijpkes [14].

Nose landing gear shock absorber

Figure 2.3a shows a schematic representation of the shock absorber in the NH90 nose landing gear. It is a development of the basic shock absorber. To optimize the damping a pressure relief valve is added in parallel with the main orifice. The orifice area of this valve is dependent on the pressure between P1 and P2. Figure 2.3b shows the characteristic of this valve.

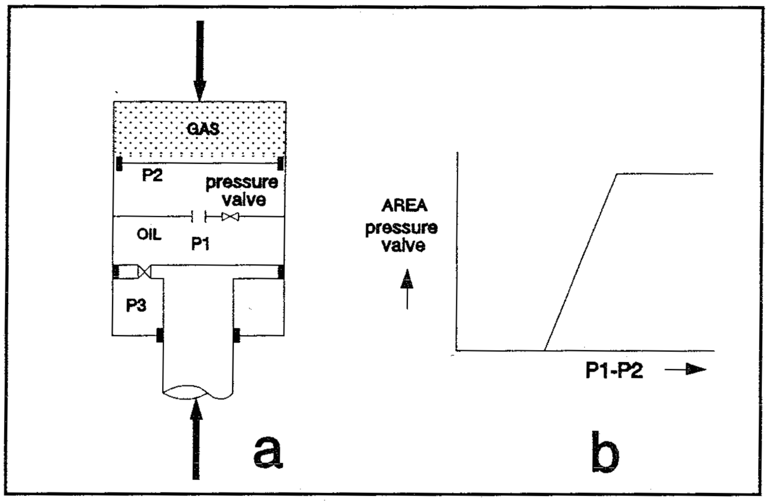


Figure 2.3 a/b nose landing gear shock absorber

A valve for which the effective area of the recoil orifice is dependent on the direction of the flow is fitted instead of the recoil orifice. In this shock absorber a separator is employed and so there is no direct contact between oil and gas.

Main landing gear shock absorber

The shock absorber in the main landing gear is simular to the one used for the nose landing gear, and may again be represented by a development of the basic shock absorber. However, in case of an emergency landing it is intended that the retract actuator of the main landing gear which is mounted directly on top of the main landing gear shock absorber also absorb part of the energy. For this purpose an emergency valve is installed in the retract actuator. This valve opens when the pressure (Pr) in the retract actuator exceeds a certain level and the oil flows through the emergency valve into an internal cavity in the retract actuator. When this occurs the main landing gear retract actuator resembles a basic shock absorber and may therefore be represented in a simular way as shown in figure 2.4a. The emergency valve has a two stage characteristic as can be seen in figure 2.4b. The first stage is used in emergency landings where no fuselage contact is allowed. In the crash landing case the second stage is used.

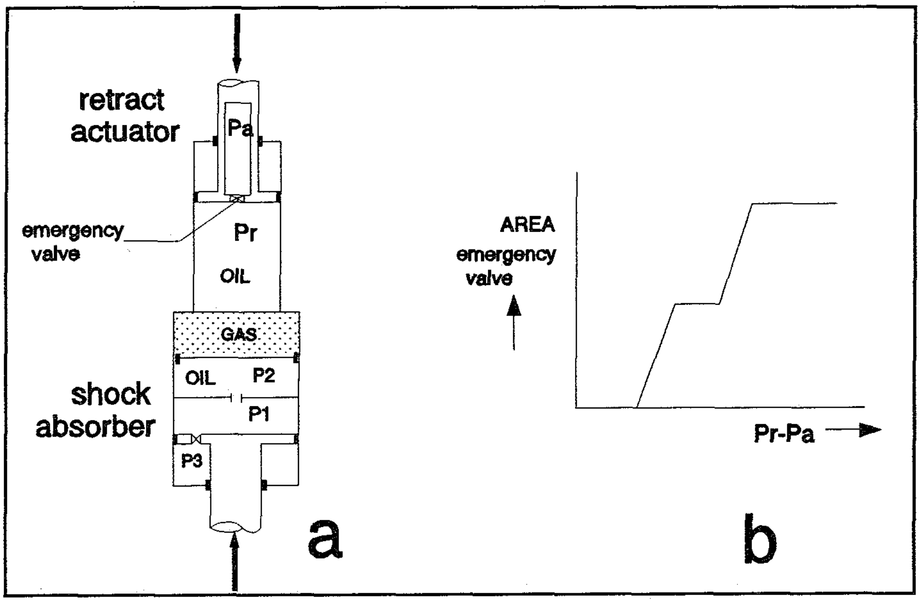


Figure 2.4a/b main gear shock absorber and retract actuator

2.3 Tire

The tire behaviour has major impact upon the dynamic behaviour of the landing gear and a large influence upon the loads which occur during the different landing cases. It is therefore important to use a tire model which describes the phenomena that occur during the different landing cases. An enumaration of the different phenomena related to the tire behaviour and a summary of the way in which these phenomena have been characterised for modelling follows.

Tire bottoming

Initially the radial tire behaviour is mainly dependent on the dimensions and the inflation pressure of the tire. In case of an emergency landing however, the helicopter will hit the ground very hard and large radial tire displacements will occur. In this case bottoming of the tire is possible. Once tire bottoming has occured the radial tire stiffness is determined by the compression of the tire material between rim and contact surface.

Slipping and rolling

In the horizontal contact plane between tire and surface a longitudinal (y-direction, see figure 2.1b), lateral force (x-direction) and a self aligning moment (about the z-axis) act on the tire. The longitudinal force depends on the rolling resistance between tire and surface and the longitudinal slip. Longitudinal slip occurs during spin-up at landing as the wheel is accelerated from rest to a peripheral velocity equal to the ground speed. The lateral force and self aligning moment are function of the slip angle which occur at yawed landings with forward velocities. It is obvious that in most cases a combination of the above mentioned forces occur. Similar reaction forces occur during braking and turning.

Camber

In case of landings with roll angle (rotation about the y-axis), lateral and vertical tire forces will occur due to wheel camber.

Shimmy

According to den Engelsman [4] the nose gear arrangement on both versions of the NH90 is sensitive for shimmy. The shimmy control on the TTH version is obtained by using a shimmy damper and on the NFH version by using the steering system. Because of the large trail and the way in which the trailing arm is mounted to the fuselage no shimmy will occur on the main landing gear.

Models

In order to describe the tire behaviour, several tire models have been developed and each has a specific field of application. In general in these models the radial tire behaviour is described with a linear spring (no tire bottoming is assumed). According to Pacejka [11] the brush-type model gives a good insight in the behaviour in the horizontal plane. The so called "Magic Formula" described in [1] appears to be a very useful model for describing the different forces in the horizontal plane. Like most of the tire models these assume a stationary condition for the discribtion of loads and moment in the horizontal plane. To describe the spin-up effect it is necessary to take the longitudinal transient behaviour of the tire into account, according to van Slagmaat [13]. For the description of the shimmy phenomenon a lateral transient tire model is also said to be required (cf. Moreland [8], Pacejka [10], den Engelsman [4]). At the moment a lot of research is undertaken on the influence of camber on the different tire loads. The main problem in using most tire models is determining of the different required tire parameters.

2.4 Crash tube

To meet the higher energy absorbtion requirements for crashworthiness of the TTH version of the helicopter additional energy at the nose landing gear is absorbed by a crash tube. This feature is not maintained on the NFH version as there is no crashworthiness requirement and a conflict exists with the projected steering arrangements. Figure 2.5 shows the manner in which the crash tube is integrated into the nose landing gear.

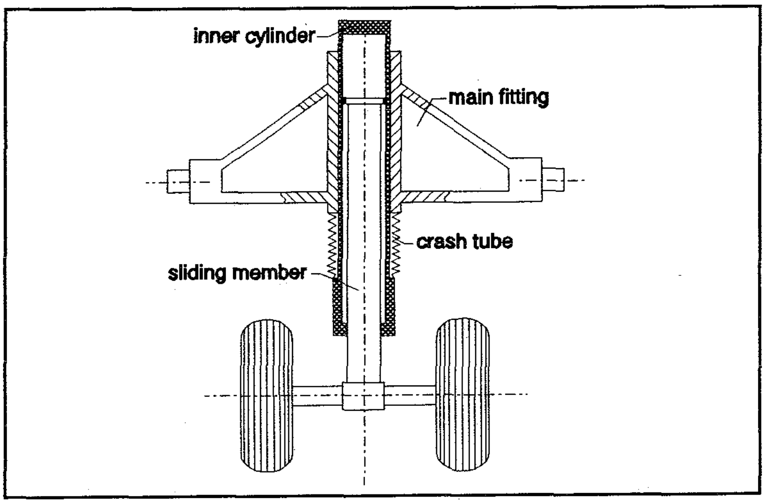


Figure 2.5 schematic view of the nose landing gear

Plastic deformation of the crash tube occurs when the initiation load (F_{int}) is reached. With further deformation of the crash tube the load remains at a constant value. This process is illustrated in figure 2.6. Effectively, the crashtube acts as a load limiter.

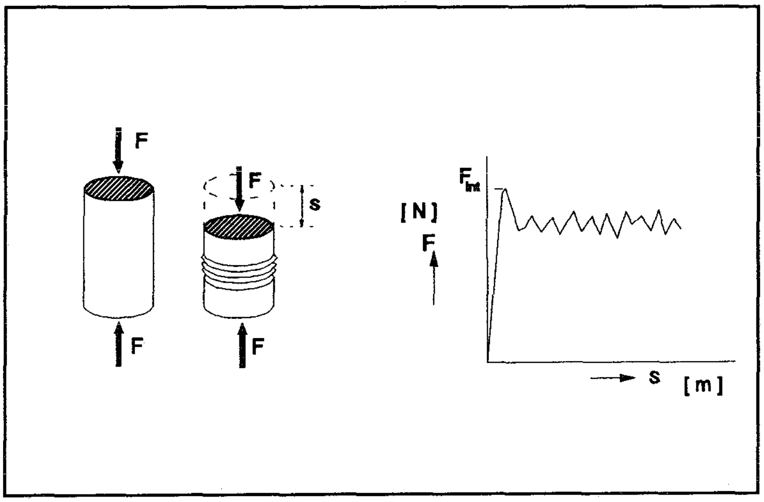


Figure 2.6 crash tube behaviour

The irregular form of the curve is a result of local buckling of the tube. This phenomenon occurs with crash tubes which are made of steel or aluminium. If composites are used the form of the curve is much smoother.

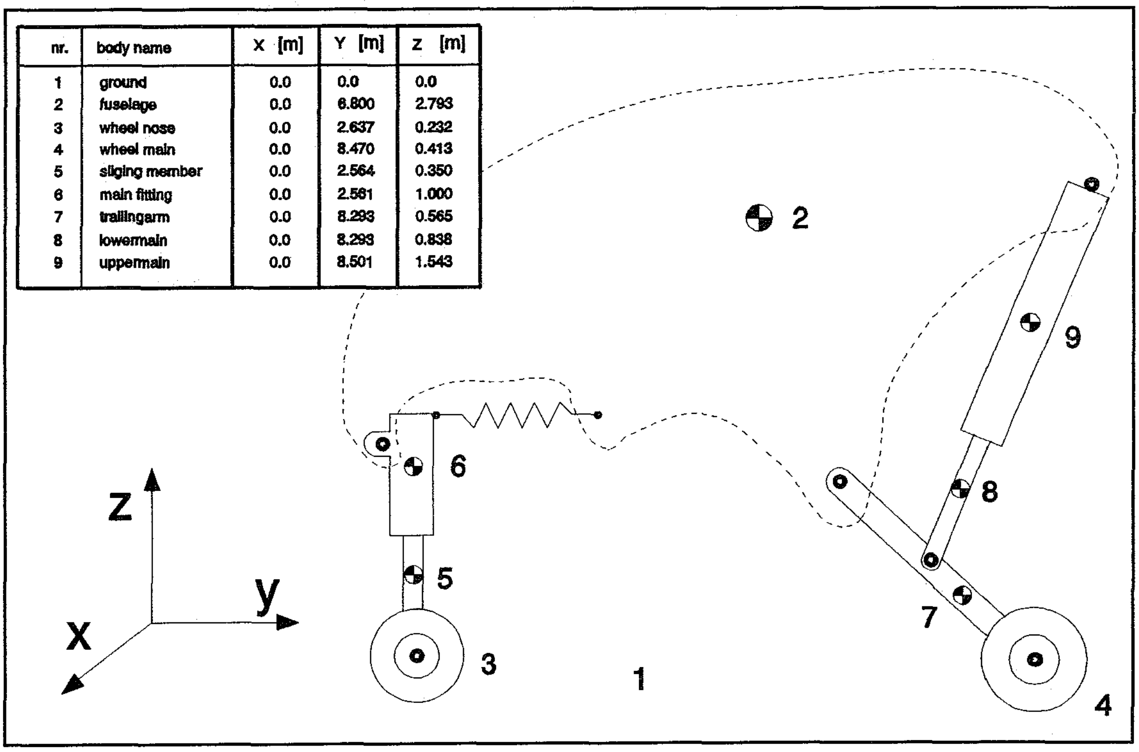
3 MODELLING IN DADS

Starting from complete commonality between the TTH and the NFH version one model is used to set up the dynamic analysis. Further on in this chapter the model is extended with a crash tube (for the TTH version).

3.1 Geometry

To produce landing loads for all operational landing conditions it is desirable to set up a three dimensional landing gear model. As time was limited and the priorities of the company had to be taken into account, however, it was decided to generate a 2D model, first. A further consideration to use a 2D model was that the crash tube model had yet to be developed for a level suitable for a 3D analysis.
The 2D model described, is actually modelled in DADS 3D so that it will be relatively simple to extend this model to a complete 3D model at a later date.

Figure (3.1) shows the model as it is implemented in DADS 3D.



nr.	body name	X [m]	Y [m]	Z [m]
1	ground	0.0	0.0	0.0
2	fuselage	0.0	6.800	2.793
3	wheel nose	0.0	2.637	0.232
4	wheel main	0.0	8.470	0.413
5	sliging member	0.0	2.564	0.350
6	main fitting	0.0	2.561	1.000
7	trailingarm	0.0	8.293	0.565
8	lowermain	0.0	8.293	0.838
9	uppermain	0.0	8.501	1.543

Figure 3.1 2D model landing gear

notes:

The retract actuator in the nose gear is modelled as a Translational Spring Damper Actuator (TSDA) element and represents the elasticity of the gear in for and aft direction.

The contact between the tires and ground is modelled with the original DADS tire-road elements (TIRE, ROAD).

The required lift force (ref. to critical design cases defined in chapter 4 "SIMULATIONS") is added to the centre of gravity of the fuselage by using the user defined force elements FRC48.

The inputfile of the geometry is listed in appendix A.

3.2. Initial conditions

The simulation starts at moment of touch down, which means that it is necessary to define the position, orientation and the initial velocity of the model. This can be done within DADS with use of initial condition elements. The definition of initial velocity in combination with small initial displacements for the landing gear model doesn't present any difficulties in the assembly process. However, if large displacements (for example a pitch angle of 20 degrees) are defined, the model can not be assembled within the defined dimensional assembly tolerance of 10^{-3} units. With the use of a restart procedure this problem can be avoided. When employing a restart procedure a restart element is used to read the initial positions and velocities for all bodies in the model form a restart file. To generate the restart file a dummy model is used. The position of the dummy model is achieved by using driver elements in a dynamic analysis. A more detailed description of the use and the restrictions of the initial conditions and the alternative restart method is given in Appendix B.

3.3 Shock absorber models

The shock absorbers in the nose and main gear are modelled as TSDA elements. The subroutines calculating the forces of an oleo are based on the work of Sijpkes [14]. They have been rewritten by van Slagmaat [13] into a user defined subroutine UFRC10 to cooperate with DADS. In this study the subroutine is extended to full use of all original functions during the development of the NH90 landing gear model. This subroutine can be used to calculate the forces in different shock absorber configurations.

The nose landing gear shock absorber as described in paragraph 2.2 is simply modelled using this subroutine. To represent the main landing gear the shock absorber and the retract actuator must be considered. To do this the shock absorber and retract actuator as described in paragraph 2.3 are reduced to a single unit as shown in figure 3.2a/b.

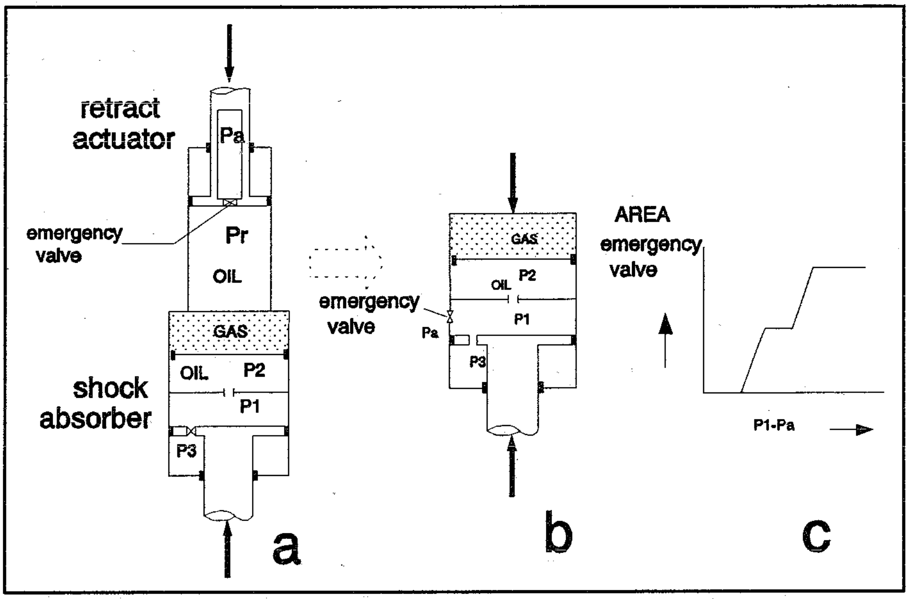


Figure 3.2 Simulation model of the shock absorber in the main landing gear

In this manner the stroke of the retract actuator caused by opening of the emergency valve is represented by additional stroke of the shock absorber.

3.3 Tire model

The standard DADS tire force and road element are used to model the tire-ground contact.

Within the tire model there are three types of force modes;

1. Basic, which ignores the rotational inertia of the wheel and the steering angle is input by the user
2. Intermediate, which ignores the rotational inertia of the wheel and calculates the steering and camber angle from the system geometry.
3. Full, which is the same as the intermediate model but also accounts for the rotational inertia.

In order to calculate the forces caused by spin-up effects during landing with forward velocities a "full" tire model is used.

Radial tire behaviour

The radial behaviour of the tires used for the NH90 can be described by the curve as shown in figure 3.3.

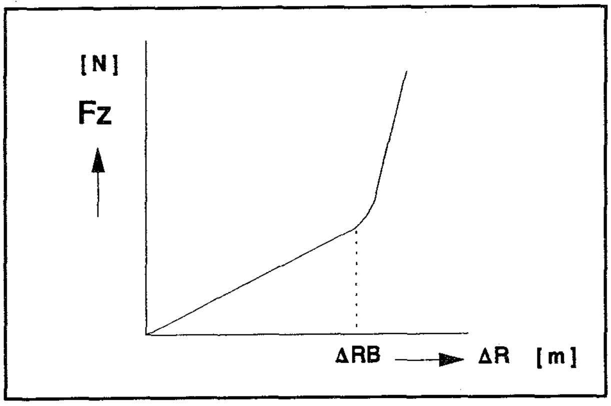


Figure 3.3 radial tire behaviour

Until the tire bottoms (point ΔRB), the tire behaviour can be described by a linear tire stiffness. Beyond this point the stiffness increases. This phenomenon is added to the force calculations within the tire subroutine and is implemented as follows:

$$Fz = \Delta R \cdot T_{stiff} \qquad \Delta R < \Delta RB$$

$$Fz = (\Delta R - \Delta RB) \cdot R_{stiff} + \Delta RB \cdot T_{stiff} \qquad \Delta R > \Delta RB$$

with:

Fz : radial tire force
ΔR : tire deflection
T_{stiff} : tire stiffness
R_{stiff} : Rim stiffness
ΔRB : deflection of the tire at which the tire bottoms

Damping of the tire has slight influence on the radial behaviour and is therefore not taken into account.

Remark on DADS:

In DADS the tire reaction forces are represented in the chassis local refrence frame. In this model the ground is chosen as chassis body to represent these forces in the global refrence frame.

3.5 Crash tube model

A simplified representation of the behaviour of this tube is shown in figure 3.4.

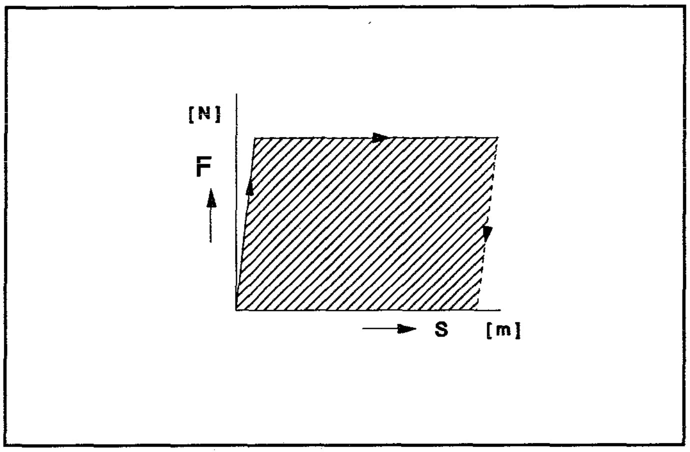


Figure 3.4 simplified representation of the crashtube behaviour

The main function of the crash tube is to absorb energy. The absorbed energy is represented by the shaded area in figure 3.4. The initial slope in the curve represents the initial linear elasticity of the crash tube.

The crash tube has to be modelled by a dissipative element which can absorb energy. Within DADS there are three methods to model this phenemenon:

1) A friction element
2) A damper element
3) By describing the process of figure 3.4 by a user defined force element.

The friction element model

The friction element defines a friction force in an existing revolute, translational or planar joint. The friction force in such a joint may be calculated by multiplying the normal force acting on the joint by the coefficient of dynamic friction. This friction force law is only valid during the period of relative motion of contacting surfaces however. If the relative velocity between the surfaces becomes zero, the force available in the joint may be less than the magnitude of the friction force and stiction of the translational joint will occur. During the period of stiction an additional constraint is added to the kinematic constraint equations which reduces the system by one degree of freedom (cf. Haug [7]). When the force available in the joint increases and becomes greater than the friction breakout force, the additional constraint is deleted.

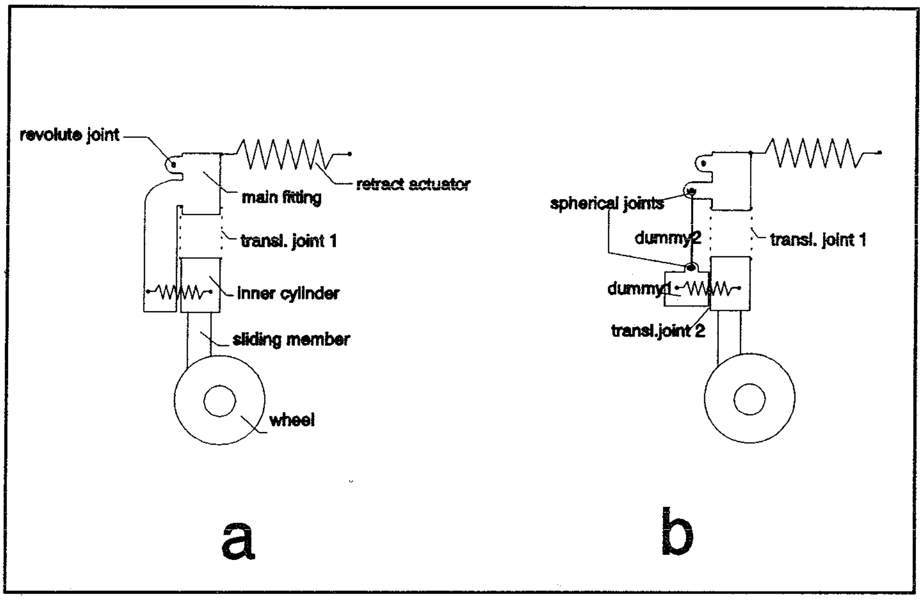


Figure 3.5 crash tube modelled with friction element

This might be employed to represent the crash tube as shown in figure 3.5a, the friction element acts on translational joint 1. In order to use the friction element for modelling the crash tube it is necessary to define the normal force acting on the translational joint, therefore a spring between main fitting and inner cylinder is employed. A constant friction force can be obtained by defining the normal force caused by the spring stiffness and dynamic and static friction coefficients which are equal. This friction force can be defined in such a manner that it corresponds with the threshold value of the crash tube. The elasticity of the crash tube is not taken into account in this model. The normal forces and moments, caused by the reaction forces on the tire, revolute joint and retract actuator, are also conducted by translational joint 1 and translated by the friction element into a friction force. To calculate the friction due to moments acting on the translational joint the DADS friction element requires the x, y, z length of the translational joint. It is not clear however which dimension affects which moment. The contribution of the moments to the friction force can be reduced by taking very small values for all three dimensions. The contribution of the transverse forces can not be reduced in this way.

To avoid these problems a superior parallel construction was developed (see figure 3.5b). The friction element in this construction is acting on translational joint 2. Due to this parallel construction no moments and transverse force are acting on translational joint 2. The normal force on translational joint 2 is induced by the spring force between dummy 1 and inner cylinder.

In the initial position (see figure 3.5b) no spring force is acting in the direction of movement. However, a relative movement between the two bodies will cause a spring force in the direction of movement which has a major impact on the collapse force in the nose gear. The influence of this phenomenon can be reduced by increasing the length of the spring. This is illustrated in figure 3.6a. However, increasing the spring length causes a phase delay of the local peaks in the shock absorber forces, which are a result of the tire bottoming. The phase delay is probably caused by the spring force in the direction of movement. A detailed investigation of this phenomenon has not been undertaken.

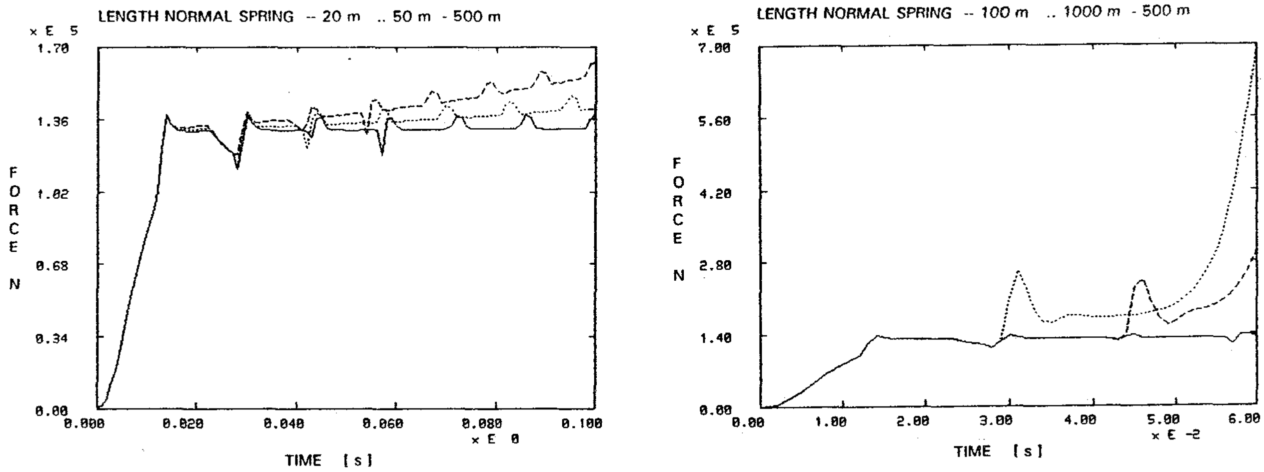


FIGURE 3.6 a

FIGURE 3.6 b

Figure 3.6a/b influence spring length on the performance of the friction element

The increasing of the normal force caused by the extension of the spring due to the relative movement between the two bodies has a minor influence.

With varying the spring length a problem arises. When for example a spring length of 100m or 1000m is used the friction element locks up despite the force available in the translational joint being much higher than the maximum friction force. Due to the lock up the total load on the nose gear has to be absorbed by the shock absorber. This results in a rapid increase of the shock absorber force as is shown in figure 3.6b. Detailed data investigations of accelerations, forces etc, in the critical areas did not clearify the reasons of this lock up problem. This experience limits the confidence in the performance of the friction element even when employed in the superior parallel construction model.

It appears that the performance of the friction element is also highly sensitive for changes in the overall geometry of the model such as displacements of centres of gravity of existing bodies and the addition or deletion of bodies and joints. In the worst case, the simulation did not start due to a singular matrix detection.

The damper element model

A damper may be introduced between main fitting and inner cylinder to represent the crash tube (see figure 3.7a).

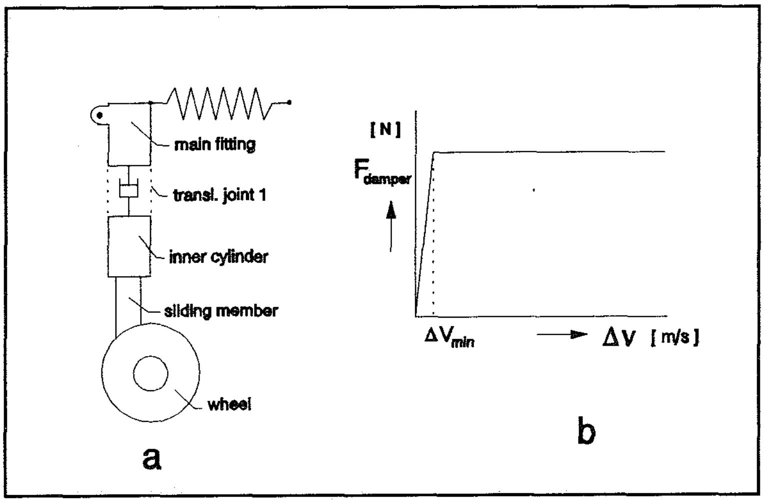


Figure 3.7 crash tube modelled with damper element

With the use of a curve element a damping force as shown in figure 3.7b can be defined. The damper force has to be a function of the relative velocity (ΔV) between the two bodies to which it is attached. As a consequence it is impossible to define a true threshold force, and relative displacement (relaxation) occurs, for lesser loads between the two bodies during the simulation. To minimise this problem, the minimum relative velocity (ΔV_{min}) needed to obtain the "threshold" value, must be as small as possible. This requires the steep initial gradient of the curve element shown in figure 3.7b.

User defined element

By representing the crash tube with a dummy TSDA element the crash tube behaviour can be described by a user defined force element (URFC10). Within this subroutine the force can be described as a function of the stroke. In this way the process of figure 3.3 can be described.

Due to the lack of time this alternative way of modelling the crash tube, was not investigated.

Comparison

Modelling a problem quickly in DADS requires for the use of standard DADS elements. Therefore a friction element and a damper element are used in the first attempt to model the crash tube behaviour. In figure 3.8 the results obtained by using a friction element and results obtained by using a damper element are compared.

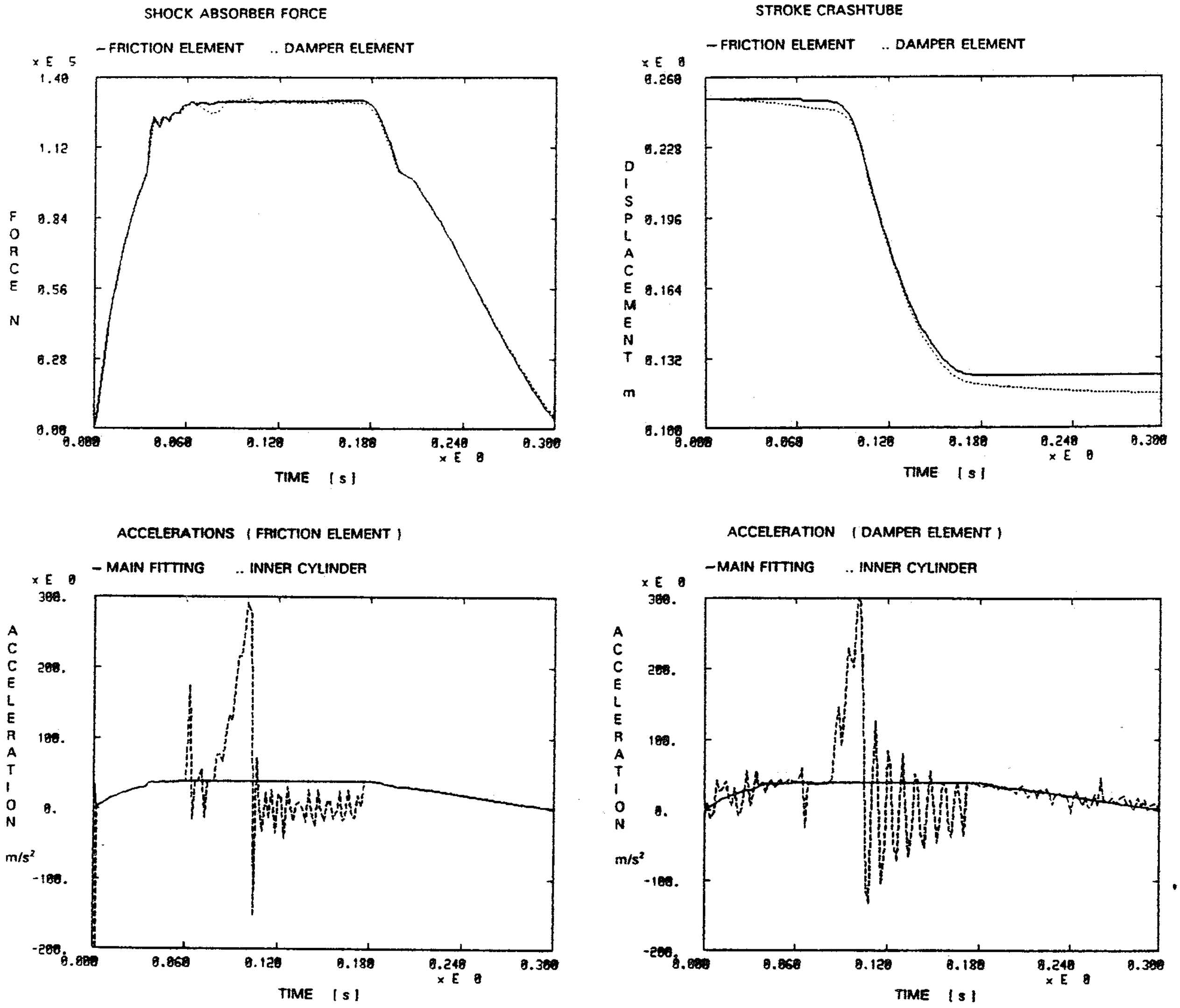


Figure 3.8 Simulation results with a damper element and a friction element

In both cases the shock absorber is limited at a level of 130 kN (threshold value of the crash tube). In the case where a friction element is used, the accelerations of the main fitting and inner cylinder, clearly illustrate the process of constraint addition and deletion correctly representing the performance of a crash tube. For the damper element the large gradient required to minimize the relaxation phenomenon (compare the strokes of the crashtube in figure 3.8) results in high frequency vibrations of the inner cylinder and an increase of the CPU time caused by the stepsize reduction in the time integration process. The simulations with the damper element and friction element took respectively 4098 and 674 CPU time. Both simulations were generated on a VAX 3900 computer. Despite incurring this significant CPU time penalty, the damper element still fails to adequately represent the performance of a crash tube. A friction element represents the behaviour of the crash tube best and the required CPU time is within acceptable limits. Therefore the friction element is prefered to model the crash tube. The simulations (described in chapter 4) were undertaken with a friction element and a spring length of 500 m.

4. SIMULATIONS

The landing gears must meet the performance requirements defined in the WSDS ([9],[15]) and derived requirements from general helicopter standards. A summary of the critical design cases is presented in table 4.1. To illustrate the behaviour of the landing gear two landing cases are selected for simulation from cases presented in table 4.1 and discussed in this chapter. The landing cases discussed are:

- hard landing (NFH version)
- crash landing (TTH version).

The system parameters for the landing gears represent the current design compromise to meet the requirements such as ground clearance, for all the critical design cases. Figure 4.1 shows an animation of a typical simulation.

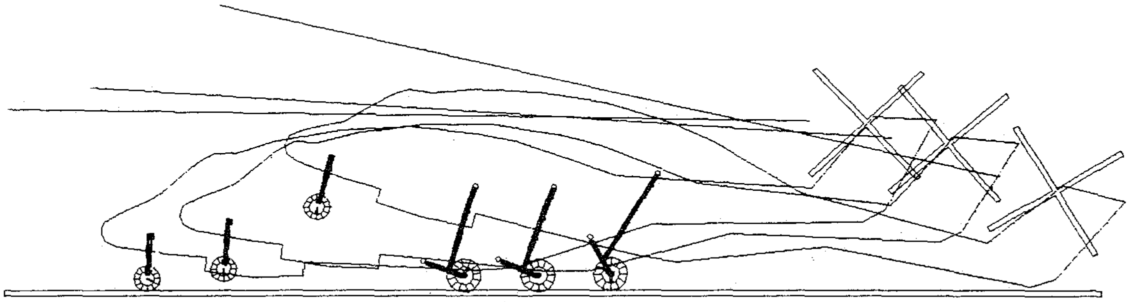


Figure 4.1 Animation of a simulation of a normal landing with pitch angle and forward velocity

Table 4.1 Critical design cases

landing case	version	limit sinking speed (m/s)	forward speed (m/s)	rotor lift to weight ratio	plastic deformation landing gear/fuselage permisseble	pitch angle degrees	roll angle degrees	bottoming of shock absorber/ tire permissible
normal landing	TTH	2.44	26.4	2/3	no/no	10	2	no/no
	NFH	3.66	-	1	no/no	±15	9 *	no/no
hard landing	TTH	4.0	-	2/3	no/no	-	-	no/no
	NFH	4.0	-	2/3	no/no	-	-	no/no
emergency landing	TTH	6.0	-	1	yes/no	-	-	**
	NFH	5.6	-	1	yes/no	-	-	**
crash landing	TTH	6.0 ..11.0	10	1	yes/yes	15	10	**
	NFH	-	-	-	-	-	-	-

* depends on the sinking speed

** either, not simultaneously

4.1 Hard landing case (NFH)

The total weight of the NFH helicopter is estimated at 9100 kg. The hard landing case is a simple vertical level landing case (see the requirements as shown in table 2.1) with a limit sinking speed of 4.0 m/s. The lift to weight factor of 2/3 means that a lift force of 2/3 of the helicopter weight is acting on the centre of gravity of the helicopter. This force represents the lift force generated by the main rotor. Plastic deformation of the landing gear and the fuselage is not allowed. Bottoming of the shock absorber and tire is also not permitted.

Nose landing gear shock absorber pressures

Figure 4.2a shows the curves of the pressures in the shock absorber of the nose landing gear. The oil pressure P1 increases rapidly due to the hydraulic damping. After reaching the first peak the closure velocity of the shock absorber decreases and as a result of this the pressure P1 decreases. Due to the flow through the main orifice the gas pressure P2 increases until the stroke has reached its maximum. At this point the closure velocity in the shock absorber becomes zero and P1 is equal to P2. The pressure difference between P1 and P2 during this compression is not sufficient to open the pressure relief valve. During compression the pressure in the recoil chamber (P3) is equal to P1 due to the fact that the recoil orifice area is very large.

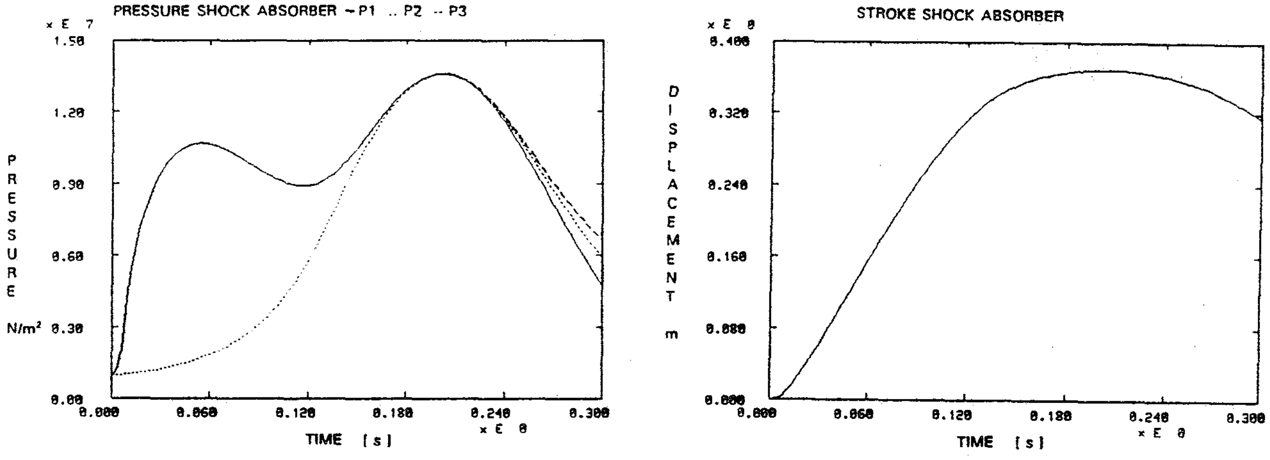


Figure 4.2a pressures and stroke of the nose landing gear shock absorber (4.0 m/s)

Main landing gear shock absorber

The same process of compression of the shock absorber can be described for the shock absorber in the main landing gear. In the hard landing case the pressure P1 in the main shock absorber is not sufficient to open the emergency valve.

Nose and main landing gear forces

Figure 4.2b shows the forces in the shock absorber of the main and nose landing gear and additionaly the vertical tire forces. The smooth form of the tire force curves indicates that no tire bottoming occurs in the nose and main landing gear for the 4.0 m/s case.

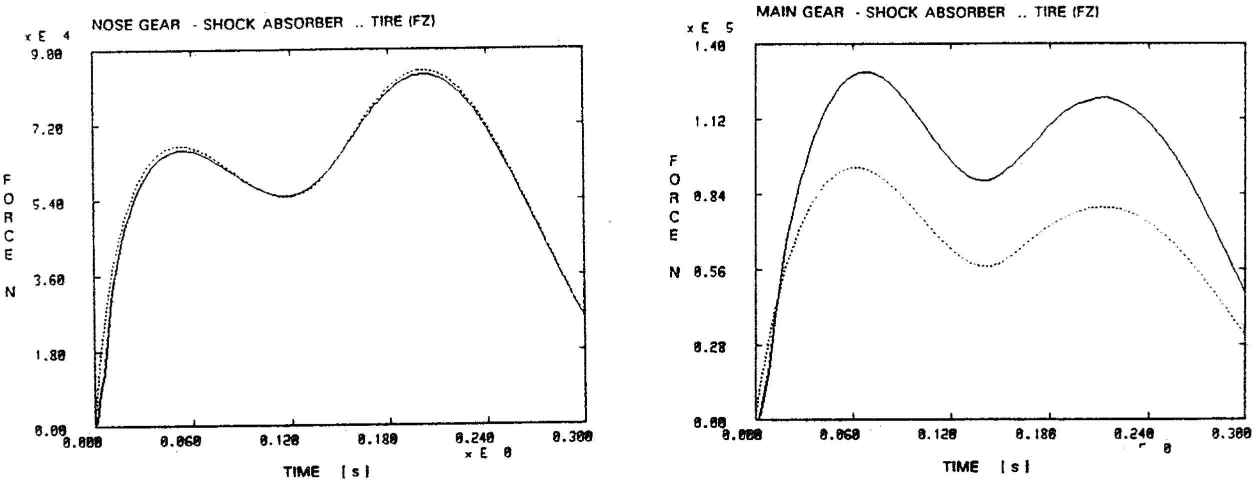


Figure 4.2b shock absorber and tire forces in the nose and main landing gear

4.2 Crash landing case (TTH)

Crashworthiness is only required for the TTH version. This version has an estimated weight of 8700 kg. To illustrate the performance of the crash tube and emergency valve two crash landing cases are discussed: 6.0 m/s and 11.0 m/s. Both are level landing cases and occur with zero forward velocities and a lift to weight ratio of 1.

For the level landing case of 6.0 m/s no fuselage ground contact is allowed, so the total helicopter kinetic energy has to be absorbed by the landing gear. In this case plastic deformation of the landing gear is acceptable. For the level landing case of 11.0 m/s, a requirement is prescribed that the landing gear must have decelerated the helicopter to 8.5 m/s to the point when the fuselage contacts the ground. In other words a part of the total helicopter kinetic energy is absorbed by the landing gear and part of it by the fuselage. To simulate the nose landing gear of the TTH version, the landing gear model is extended with the crash tube model. In this model the standard DADS friction element is used as described in paragraph 3.4.

Main landing gear

Figure 4.3a shows the results for the main landing gear in the 6.0 m/s and 11 m/s case. In the crash landing case bottoming of the tire is permissible. In case of the 6.0 m/s no bottoming of the tire occurs as shown by smooth shape of the tire force curve. In the 11.0 m/s case however the tire bottoms massively and results in a high peak load in the tire force.

The emergency valve opens when the pressure P1 exceeds 230 bar. See points (1) in figure 4.3. In the 6.0 m/s case the valve hardly opens but is just sufficient to reduce the peak loads in the shock absorber and tire. In the 11 m/s case the valve massively opens mainly as a result of the tire bottoming. Opening of the emergency valve gives better efficiency of the shock absorber*.

---

* *The shock absorber efficiency is defined as energy absorbed by the shock absorber during compression divided by the maximum shock absorber force multiplied by the maximum stroke.*

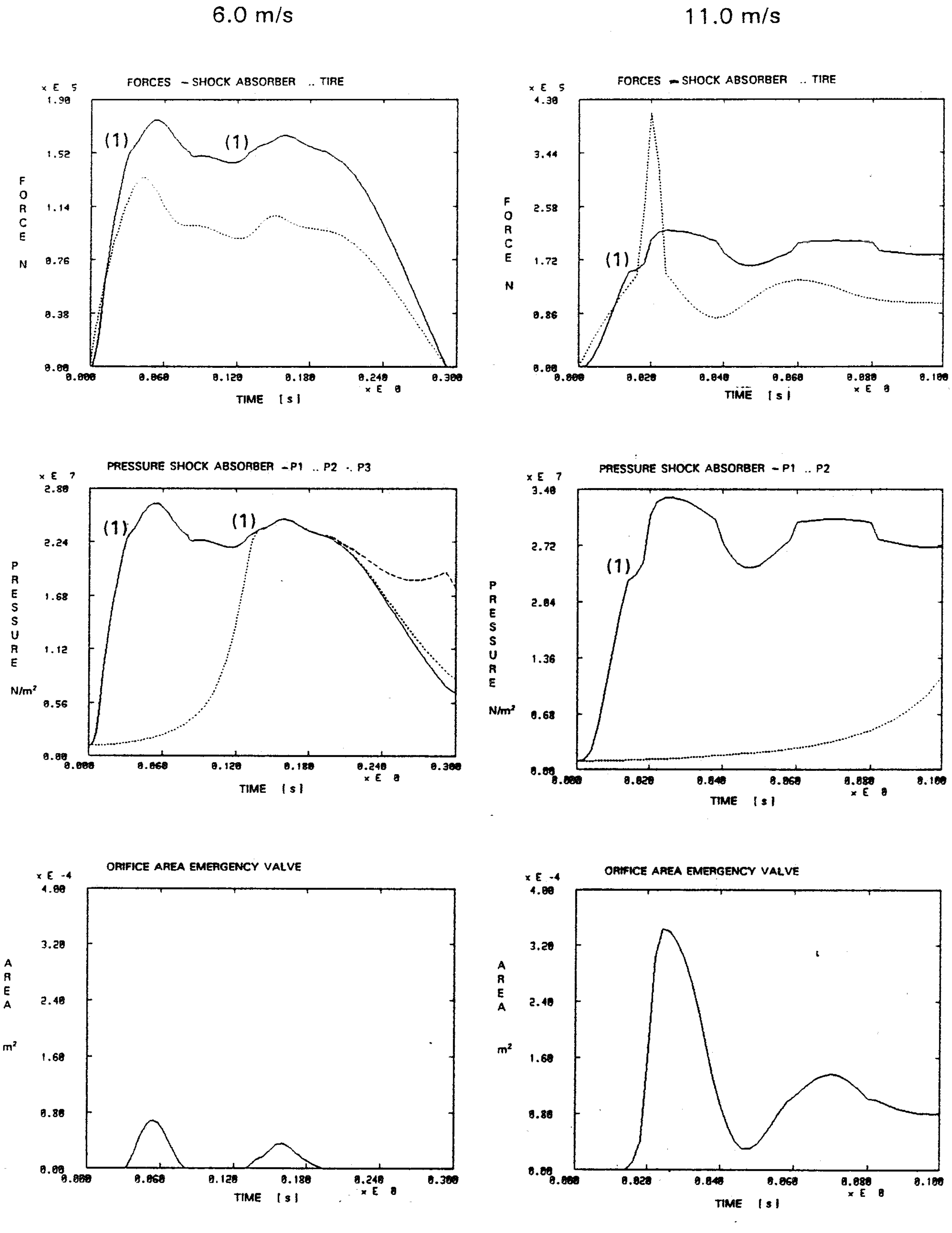


Figure 4.3a results main landing gear (left 6.0 and right 11.0 m/s)

Nose landing gear

The impact of the crash tube on the behaviour of the nose landing gear is shown in figure 4.3b. The shock absorber force in the nose gear is limited due to the threshold value of the crash tube. The crash tube in this model collapses at a level of 130 kN. In the 6.0 m/s landing case this level is reached in the second force peak due to a combination of tire bottoming and the steep gas spring curve (1). The stroke of the crash tube in this case is nearly 10 mm. The real effective stroke of the crash tube is 230 mm. In the model however, this stroke is not limited. This explains the unrealistic stroke in the 11 m/s case. To decelerate the fuselage to 8.5 m/s takes about 0.064 seconds. At this point the deformed length of the crash tube has reached a value of 65 mm. After this point no more energy has to be absorbed by the landing gear, as the plastic deformation of the fuselage takes over the energy absorbing function, following the design specifications.

In the 11 m/s landing case the tire forces are very high due to tire bottoming. The pressure relief valve opens only in the 11 m/s case and results in a decreasing of the first three tire force peaks. After 0.055 seconds the pressure difference between P1 and P2 is not sufficient to keep the valve open. This explains the increasing of the (fourth) tire force peak. The difference between tire force and shock absorber force is a result of the acceleration of the unsprung mass (wheel and sliding member).

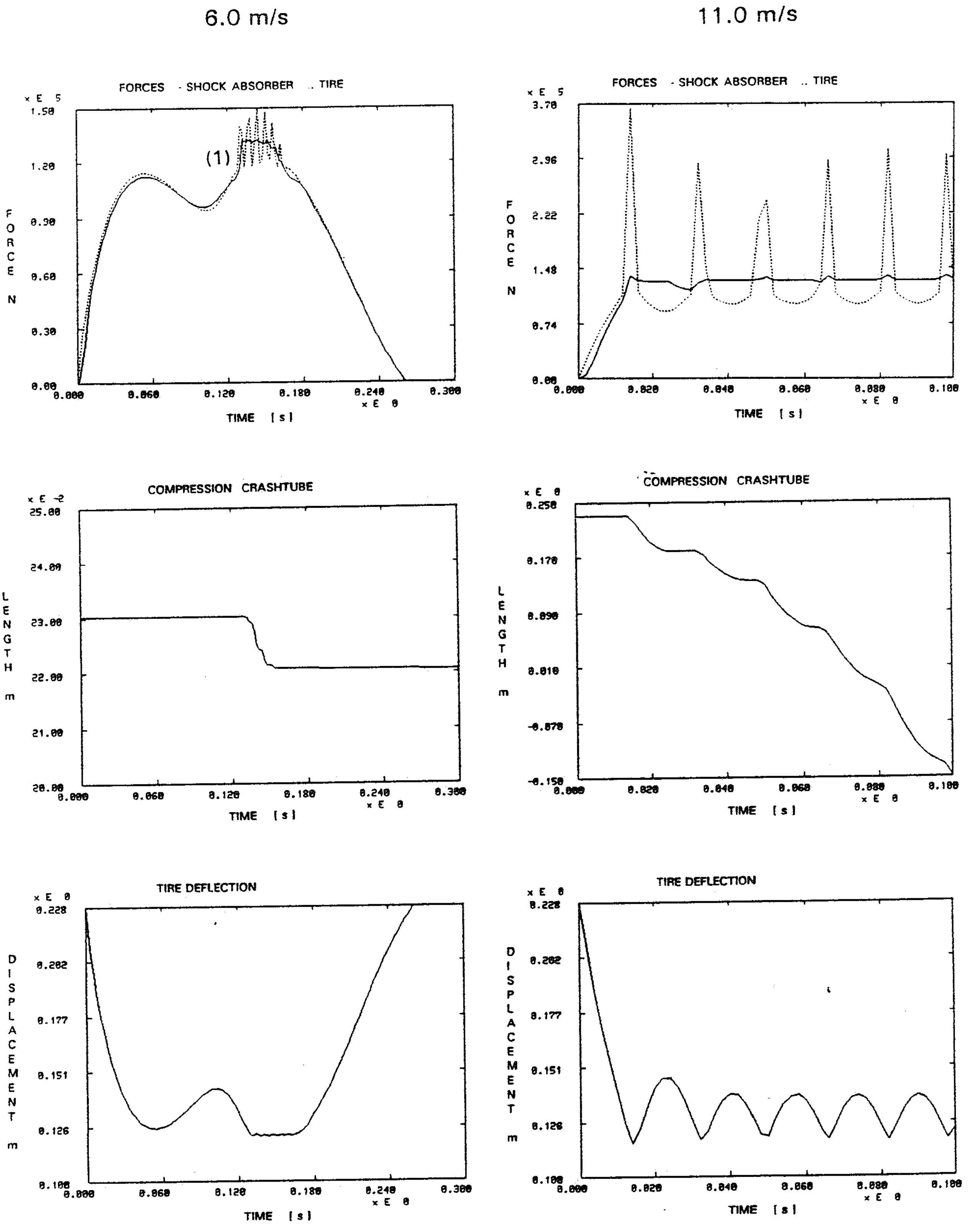


Figure 4.3b results nose landing gear (6.1 and 11.0 m/s)

5. CONCLUSIONS

Initial conditions

Positioning the model at the beginning of the simulation with the use of initial conditions can cause a failure in the assembly process with complex models if large initial displacements are defined. When this occurs a restart procedure may be used to generate the initial conditions. Such failures did occur in the modelling and this alternative procedure was used.
LMS* confirms this problem and suggest the same alternative.

Crash tube modelling

To avoid the disadvantages of a damper element (such as relaxation displacements and large CPU times) a standard friction element was used to model the crash tube behaviour. The performance of the friction element however, is not very reliable and in the worst case, the use of a friction element can result in a singular system matrix.
LMS* also confirms this problem.

Discontinuities

The discontinuities in the tire behaviour (tire bottoming) and shock absorber (opening emergency and pressure valves) cause no problem in the time integration algorithm and increase of the CPU time is limited.

Design tool

Generating a dynamic model in DADS is relatively simple. However, using DADS as a design tool as described in this report, demands a lot of skill of the user.

In this report some pitfalls and imperfections of DADS are discussed. To draw firm conclusions on the usability of DADS in general one would have to compare the performance with other multidynamic programs.

**LMS International, Leuven, Belgium is the sales and support organisation for DADS in the Benelux.*

6. FURTHER RESEARCH

Further research will be undertaken to advance the design of NH90 landing gear. Therefore the dynamic model of the landing gear will be extended in the following areas.

Geometry

For simulations with a roll angle and side loads a complete 3D model (one nose gear and two main gears) is required. Extension of the existing model is easy and with the use of the restart procedure any initial position of the model can be defined.
To make the model more representative and to reduce the peak loads on the main landing gear caused by the bottoming of the tire, a flexible trailing arm will be modelled.

Crash tube

Because of the bad perfomance of the friction element, an attempt will be made to model the crash tube with the use of a user defined subroutine (UFRC10).

When the fuselage is decelerated to 11.0 m/s from 8.5 m/s (in the crash landing case), the nose landing gear of the TTH version is retracted to avoid the gear penetrating the crew area. In order to simulate this design feature, the existing landing gear model will be extended.

Tire

In order to model the spin-up effect, the tire model has to be extended with a transient behaviour in longitudinal direction (cf. van Slagmaat [13]).
During crash landings high vertical tire forces occur due to the bottoming of the tire. In reality the rim will probably deform plasticly. This will be taken into account.

Shock absorber

Verbeek [16] points out that for a more accurate model of the shock absorber, the heat transfer between the inflation gas and strut walls under dynamic conditions have to be taken into account. The results of his research will be used to improve the mathematical model for the simulation program. However, adding the first order differential equation in the temperature representing the thermo dynamics is not possible in DADS. Therefore another solution must be found to take this feature into account.

The performance of the shock absorber largely depends on charateristics of the valves, used for various damping functions. The analysis work reported has lead to a design requirement for valves and suitable valves must now be designed. As a part of this activity detailed dynamic models of these valves will be developed. Ultimately, the detailed models will be added to the complete landing gear model.

According to [5] and [16] strut friction should be included in the simulation models. Friction in the bearings of the strut depends on the normal forces acting on the bearings of the strut. These normal forces can be caused by an offset of the wheel loads from the strut centre line, wheel spin-up and breaking or turning.
Friction in the seals depends on the inflation pressure in the shock strut. Strut friction will be represented in the model.

The influence of the expansion of the strut due to the high oil and gas pressure should be investigated

Drop test
When the first prototypes of the landing gear are available they will be subjected to a drop test. The models will be compared with the results of these tests in order to verify and optimize the models.

7 REFERENCES

1. Bakker E., Lidner L. & Pacejka H.B., 1987, Tyre modelling for use in vehicle dynamic studies, SAE 870421, Volvo car corp. & Delft university of technology, Delft, Netherlands.

2. Curry N.S., 1982, Landing gear design handbook, First Edition.

3. DADS user manual,1991, rev. 6.5, CADSI, Oakdale, Iowa, USA.

4. Engelsman W. P.,1992, Shimmy, an introduction, 2199-59000/92-1287, DAF Special Products, Eindhoven, Netherlands.

5. Greer McClain J., Grenshaw B.M., 1990, Landing gear improvements for transport aircraft, Wright-Patterson AFB, Ohio, USA, and Lockheed Aeronautical Systems Company, Georgia, USA.

6. Haug E. J.,1988, Computer aided kinematics and dynamics of mechanical systems, volume 1: basic methods, Iowa City, Iowa, USA.

7. Haug E.J., Yang S.M., Shih C.W., 1984, Dynamics of mechanical systems with coulomb friction, stiction, impact and constraint addition deletion, Centre for Computer Aided Design and Department of Mechanical Engineering, The University of Iowa.

8. Moreland W.L., 1954, The story of shimmy, Journal of Aeronautical Science, vol. 21, no. 12.

9. NATO Frigate Helicopter (NFH) Weapon System Development Specification, DD-0002/G.

10. Pacejka H.B., 1966, The wheel shimmy phenomenon, a theoretical and experimental investigation with particular reference to the non-linear problem, PhD Thesis, University of Technology, Delft, Netherlands.

11. Pacejka H.B., 1988, Modelling of the pneumatic tire and its impact on vehicle dynamic behaviour, University of Technology, Delft, Netherlands.

12. Sauren A., 1988, Multibody dynamica, college notes, nr. 4659, University of Technology, Eindhoven, Netherlands.

13. Slagmaat M.T.P. van, Non-linear modelling and simulation of landing gear, 2199-59000/91-661, DAF Special Products, Eindhoven, Netherlands.

14. Sijpkes T., 1987, Calculation of hydro-pneumatic spring damper elements, DAF Special Products, Eindhoven, Netherlands.

15. Technical Transport Helicopter (TTH) Weapon System Development Specification, DD-0001/G.

16. Verbeek G., 1992, Parametric identification of an aircraft landing gear damper, by means of periodic exicitation, University of Technology, Eindhoven, Netherlands.

APPENDIX A Input file

```
CREATE HEADER
2 Dimensional model of the NH90 helicopter ( TTH version ),
including the crash tube model.
File name : 2D_TTH_1100

ANALYSIS
CREATE SYSTEM.DATA
   UNITS                                   := 'SI'
   ANALYSIS.TYPE                           := 'DYNAMIC'
   STARTING.TIME                           := '0.0'
   ENDING.TIME                             := '0.1'
   PRINT.INTERVAL                          := '0.002'
   GRAVITY.SEA.LEVEL                       := '9.80665'
   X.GRAVITY                               := '0.0'
   Y.GRAVITY                               := '0.0'
   Z.GRAVITY                               := '-1.0'
   SCALE.GRAVITY.COEF                      := '1.0'
   MATRIX.OPERATIONS                       := 'SPARSE'
   REDUNDANCY.CHECK                        := 'TRUE'
   LU.TOL                                  := '1.0D-12'
   ASSEMBLY.TOL                            := '1.0D-3'
   BYPASS.ASSEMBLY                         := 'FALSE'
   OUTPUT.FILE                             := 'ALL'
   REFERENCE.FRAME                         := 'GLOBAL'
   DEBUG.FLAG                              := 'FALSE'
UP
CREATE DYNAMIC.DATA
   REACTION.FORCES                         := 'TRUE'
   FORCE.COORDINATES                       := 'GLOBAL'
   PRINT.METHOD                            := 'INTERPOLATED'
   MAX.INT.STEP                            := '0.0002'
   SOLUTION.TOL                            := '0.00002'
   INTEGRATION.TOL                         := '0.000002'
   METHOD.INTEGRATION                      := 'VARIABLE'
   PRINT.FREQ                              := '0'
UP
UP
CONSTRAINTS
CREATE POSITION.CONSTRAINT
   NAME                                    := 'XPOS_FUSELAGE'
   BODY.NAME                               := 'FUSELAGE'
   TYPE.CONSTRAINT                         := 'X'
   CONSTRAINT.VALUE                        := '0.0'
   P.ON.BODY                               := ( 0.0, 6.800, 2.793 )
   Q.ON.BODY                               := ( 0.0, 6.800, 3.793 )
   R.ON.BODY                               := ( 1.0, 6.800, 2.793 )
UP
CREATE POSITION.CONSTRAINT
   NAME                                    := 'XPOS_WHEELMAIN'
   BODY.NAME                               := 'WHEELMAIN'
   TYPE.CONSTRAINT                         := 'X'
   CONSTRAINT.VALUE                        := '0.0'
   P.ON.BODY                               := ( 0.0, 8.470, 0.413 )
   Q.ON.BODY                               := ( 0.0, 8.470, 1.413 )
   R.ON.BODY                               := ( 1.0, 8.470, 0.413 )
UP
```

```
CREATE POSITION.CONSTRAINT
   NAME                                    := 'XPOS_WHEELNOSE'
   BODY.NAME                               := 'WHEELNOSE'
   TYPE.CONSTRAINT                         := 'X'
   CONSTRAINT.VALUE                        := '0.0'
   P.ON.BODY                               := ( 0.0, 2.637, 0.232 )
   Q.ON.BODY                               := ( 0.0, 2.637, 1.232 )
   R.ON.BODY                               := ( 1.0, 2.637, 0.232 )
UP
UP
FORCE
CREATE FRICTION
   NAME                                    := 'FRIC_CRASHTUBE'
   JOINT.NAME                              := 'TR_DUMMY'
   STATIC.COEFF                            := '0.1'
   DYNAMIC.COEFF                           := '0.1'
   VELOCITY.TOL                            := '0.001'
   X.DIMENSION                             := '1.0'
   Y.DIMENSION                             := '1.0'
   Z.DIMENSION                             := '1.0'
UP
CREATE TIRE
   NAME                                    := 'TIRE_NOSE'
   TIRE.BODY                               := 'WHEELNOSE'
   CHASSIS.BODY                            := 'GROUND'
   TYPE                                    := 'FULL'
   P.ON.TIRE                               := ( 0.0, 2.637, 0.232 )
   RADIUS                                  := '0.258'
   ROLLING.RESISTANCE                      := '0.0'
   DAMPING.CONSTANT                        := '0.0'
   VERTICAL.STIFF                          := '0.0'
   LATERAL.STIFF                           := '0.0'
   STEER.ANGLE                             := '0.0'
   FRICTION.COEFF                          := '0.0'
   CURVE.UTILITY                           := 'NONE'
   CURVE.VERTICAL                          := 'NONE'
   CURVE.TORQUE                            := 'NONE'
   CURVE.STEER                             := 'NONE'
   ANGULAR.UNITS                           := 'DEGREES'
   ALIGN.COEFF                             := '0.0'
UP
CREATE TIRE
   NAME                                    := 'TIRE_MAIN'
   TIRE.BODY                               := 'WHEELMAIN'
   CHASSIS.BODY                            := 'GROUND'
   TYPE                                    := 'FULL'
   P.ON.TIRE                               := ( 0.0, 8.470, 0.413 )
   RADIUS                                  := '0.334'
   ROLLING.RESISTANCE                      := '0.0'
   DAMPING.CONSTANT                        := '0.0'
   VERTICAL.STIFF                          := '0.0'
   LATERAL.STIFF                           := '0.0'
   STEER.ANGLE                             := '0.0'
   FRICTION.COEFF                          := '0.0'
   CURVE.UTILITY                           := 'NONE'
   CURVE.VERTICAL                          := 'NONE'
   CURVE.TORQUE                            := 'NONE'
   CURVE.STEER                             := 'NONE'
   ANGULAR.UNITS                           := 'DEGREES'
   ALIGN.COEFF                             := '0.0'
```

```
UP
CREATE TSDA
   NAME                              := 'OLEO1'
   BODY.1.NAME                       := 'INNER_CYLINDER'
   BODY.2.NAME                       := 'SLIDING_MEMBER'
   SPRING.CONSTANT                   := '0.0'
   FREE.LENGTH.SPRING                := '0.4000'
   DAMPING.COEFFICIENT               := '0.0'
   ACTUATOR.FORCE                    := '0.0'
   P.ON.BODY.1                       := ( 0.0, 2.561, 0.750 )
   P.ON.BODY.2                       := ( 0.0, 2.561, 0.350 )
   Q.ON.BODY.1                       := ( 0.0, 2.561, 1.750 )
   Q.ON.BODY.2                       := ( 0.0, 2.561, 1.350 )
   R.ON.BODY.1                       := ( 1.0, 2.561, 0.750 )
   R.ON.BODY.2                       := ( 1.0, 2.561, 0.350 )
   CURVE.SPRING                      := 'NONE'
   CURVE.DAMPER                      := 'NONE'
   CURVE.ACTUATOR                    := 'NONE'
   NODE.1                            := '0'
   NODE.2                            := '0'
   TYPE                              := 'BIDIRECTIONAL'
UP
CREATE TSDA
   NAME                              := 'OLEO2'
   BODY.1.NAME                       := 'UPPERMAIN'
   BODY.2.NAME                       := 'LOWERMAIN'
   SPRING.CONSTANT                   := '0.0'
   FREE.LENGTH.SPRING                := '0.73504'
   DAMPING.COEFFICIENT               := '0.0'
   ACTUATOR.FORCE                    := '0.0'
   P.ON.BODY.1                       := ( 0.0, 8.501, 1.543 )
   P.ON.BODY.2                       := ( 0.0, 8.293, 0.838 )
   Q.ON.BODY.1                       := ( 0.0, 8.796, 2.543 )
   Q.ON.BODY.2                       := ( 0.0, 8.588, 1.838 )
   R.ON.BODY.1                       := ( 1.0, 8.501, 1.543 )
   R.ON.BODY.2                       := ( 1.0, 8.293, 0.838 )
   CURVE.SPRING                      := 'NONE'
   CURVE.DAMPER                      := 'NONE'
   CURVE.ACTUATOR                    := 'NONE'
   NODE.1                            := '0'
   NODE.2                            := '0'
   TYPE                              := 'BIDIRECTIONAL'
UP
CREATE TSDA
   NAME                              := 'DRAGBRACE'
   BODY.1.NAME                       := 'MAIN_FITTING'
   BODY.2.NAME                       := 'FUSELAGE'
   SPRING.CONSTANT                   := '1E7'
   FREE.LENGTH.SPRING                := '0.9413825'
   DAMPING.COEFFICIENT               := '1E6'
   ACTUATOR.FORCE                    := '0.0'
   P.ON.BODY.1                       := ( 0.0, 2.660, 1.311 )
   P.ON.BODY.2                       := ( 0.0, 3.600, 1.260 )
   Q.ON.BODY.1                       := ( 0.0, 3.660, 1.257 )
   Q.ON.BODY.2                       := ( 0.0, 4.600, 1.206 )
   R.ON.BODY.1                       := ( 1.0, 2.660, 1.311 )
   R.ON.BODY.2                       := ( 1.0, 3.600, 1.260 )
   CURVE.SPRING                      := 'NONE'
   CURVE.DAMPER                      := 'NONE'
   CURVE.ACTUATOR                    := 'NONE'
```

```
      NODE.1                                := '0'
      NODE.2                                := '0'
      TYPE                                  := 'BIDIRECTIONAL'
   UP
   CREATE TSDA
      NAME                                  := 'NORMAL_FORCE'
      BODY.1.NAME                           := 'INNER_CYLINDER'
      BODY.2.NAME                           := 'DUMMY2'
      SPRING.CONSTANT                       := '2600000.0'
      FREE.LENGTH.SPRING                    := '499.5'
      DAMPING.COEFFICIENT                   := '0.0'
      ACTUATOR.FORCE                        := '0.0'
      P.ON.BODY.1                           := ( 0.0, 2.561, 0.750 )
      P.ON.BODY.2                           := ( 500.0, 2.561, 0.750 )
      Q.ON.BODY.1                           := ( 1.0, 2.561, 0.750 )
      Q.ON.BODY.2                           := ( 501.0, 2.561, 0.750 )
      R.ON.BODY.1                           := ( 0.0, 2.561, 1.750 )
      R.ON.BODY.2                           := ( 500.0, 2.561, 1.750 )
      CURVE.SPRING                          := 'NONE'
      CURVE.DAMPER                          := 'NONE'
      CURVE.ACTUATOR                        := 'NONE'
      NODE.1                                := '0'
      NODE.2                                := '0'
      TYPE                                  := 'BIDIRECTIONAL'
   UP
   UP
   JOINTS
   CREATE REVOLUTE.JOINT
      NAME                                  := 'REV_WHEELNOSE'
      BODY.1.NAME                           := 'SLIDING_MEMBER'
      BODY.2.NAME                           := 'WHEELNOSE'
      P.ON.BODY.1                           := ( 0.0, 2.637, 0.232 )
      P.ON.BODY.2                           := ( 0.0, 2.637, 0.232 )
      Q.ON.BODY.1                           := ( 1.0, 2.637, 0.232 )
      Q.ON.BODY.2                           := ( 1.0, 2.637, 0.232 )
      R.ON.BODY.1                           := ( 0.0, 2.637, 1.232 )
      R.ON.BODY.2                           := ( 0.0, 2.637, 1.232 )
      NODE.1                                := '0'
      NODE.2                                := '0'
   UP
   CREATE REVOLUTE.JOINT
      NAME                                  := 'REV_WHEELMAIN'
      BODY.1.NAME                           := 'TRAILINGARM'
      BODY.2.NAME                           := 'WHEELMAIN'
      P.ON.BODY.1                           := ( 0.0, 8.470, 0.413 )
      P.ON.BODY.2                           := ( 0.0, 8.470, 0.413 )
      Q.ON.BODY.1                           := ( 1.0, 8.470, 0.413 )
      Q.ON.BODY.2                           := ( 1.0, 8.470, 0.413 )
      R.ON.BODY.1                           := ( 0.0, 8.470, 1.413 )
      R.ON.BODY.2                           := ( 0.0, 8.470, 1.413 )
      NODE.1                                := '0'
      NODE.2                                := '0'
   UP
   CREATE REVOLUTE.JOINT
      NAME                                  := 'REV_MAIN_FITTING'
      BODY.1.NAME                           := 'MAIN_FITTING'
      BODY.2.NAME                           := 'FUSELAGE'
      P.ON.BODY.1                           := ( 0.0, 2.545, 1.046 )
      P.ON.BODY.2                           := ( 0.0, 2.545, 1.046 )
      Q.ON.BODY.1                           := ( 1.0, 2.545, 1.046 )
```

```
      Q.ON.BODY.2                         := ( 1.0, 2.545, 1.046 )
      R.ON.BODY.1                         := ( 0.0, 2.545, 2.046 )
      R.ON.BODY.2                         := ( 0.0, 2.545, 2.046 )
      NODE.1                              := '0'
      NODE.2                              := '0'
   UP
   CREATE REVOLUTE.JOINT
      NAME                                := 'REV_TRAILINGARM'
      BODY.1.NAME                         := 'TRAILINGARM'
      BODY.2.NAME                         := 'FUSELAGE'
      P.ON.BODY.1                         := ( 0.0, 7.880, 1.065 )
      P.ON.BODY.2                         := ( 0.0, 7.880, 1.065 )
      Q.ON.BODY.1                         := ( 1.0, 7.880, 1.065 )
      Q.ON.BODY.2                         := ( 1.0, 7.880, 1.065 )
      R.ON.BODY.1                         := ( 0.0, 7.880, 2.065 )
      R.ON.BODY.2                         := ( 0.0, 7.880, 2.065 )
      NODE.1                              := '0'
      NODE.2                              := '0'
   UP
   CREATE REVOLUTE.JOINT
      NAME                                := 'REV_LOWERMAIN'
      BODY.1.NAME                         := 'TRAILINGARM'
      BODY.2.NAME                         := 'LOWERMAIN'
      P.ON.BODY.1                         := ( 0.0, 8.242, 0.665 )
      P.ON.BODY.2                         := ( 0.0, 8.242, 0.665 )
      Q.ON.BODY.1                         := ( 1.0, 8.242, 0.665 )
      Q.ON.BODY.2                         := ( 1.0, 8.242, 0.665 )
      R.ON.BODY.1                         := ( 0.0, 8.242, 1.665 )
      R.ON.BODY.2                         := ( 0.0, 8.242, 1.665 )
      NODE.1                              := '0'
      NODE.2                              := '0'
   UP
   CREATE REVOLUTE.JOINT
      NAME                                := 'REV_UPPERMAIN'
      BODY.1.NAME                         := 'UPPERMAIN'
      BODY.2.NAME                         := 'FUSELAGE'
      P.ON.BODY.1                         := ( 0.0, 8.813, 2.600 )
      P.ON.BODY.2                         := ( 0.0, 8.813, 2.600 )
      Q.ON.BODY.1                         := ( 1.0, 8.813, 2.600 )
      Q.ON.BODY.2                         := ( 1.0, 8.813, 2.600 )
      R.ON.BODY.1                         := ( 0.0, 8.813, 3.600 )
      R.ON.BODY.2                         := ( 0.0, 8.813, 3.600 )
      NODE.1                              := '0'
      NODE.2                              := '0'
   UP
   CREATE SPHERICAL.JOINT
      NAME                                := 'SPH_DUMMY1'
      BODY.1.NAME                         := 'MAIN_FITTING'
      BODY.2.NAME                         := 'DUMMY1'
      P.ON.BODY.1                         := ( 0.0, 2.561, 1.000 )
      P.ON.BODY.2                         := ( 0.0, 2.561, 1.000 )
      Q.ON.BODY.1                         := ( 0.0, 3.561, 1.000 )
      Q.ON.BODY.2                         := ( 0.0, 3.561, 1.000 )
      R.ON.BODY.1                         := ( 0.0, 2.561, 2.000 )
      R.ON.BODY.2                         := ( 0.0, 2.561, 2.000 )
      NODE.1                              := '0'
      NODE.2                              := '0'
   UP
   CREATE SPHERICAL.JOINT
      NAME                                := 'SPH_DUMMY2'
```

```
         BODY.1.NAME                       := 'DUMMY1'
         BODY.2.NAME                       := 'DUMMY2'
         P.ON.BODY.1                       := ( 0.0, 2.561, 0.750 )
         P.ON.BODY.2                       := ( 0.0, 2.561, 0.750 )
         Q.ON.BODY.1                       := ( 0.0, 2.561, 1.750 )
         Q.ON.BODY.2                       := ( 0.0, 2.561, 1.750 )
         R.ON.BODY.1                       := ( 1.0, 2.561, 0.750 )
         R.ON.BODY.2                       := ( 1.0, 2.561, 0.750 )
         NODE.1                            := '0'
         NODE.2                            := '0'
      UP
      CREATE TRANSLATIONAL.JOINT
         NAME                              := 'TR_DUMMY'
         BODY.1.NAME                       := 'DUMMY2'
         BODY.2.NAME                       := 'INNER_CYLINDER'
         P.ON.BODY.1                       := ( 0.0, 2.561, 0.750 )
         P.ON.BODY.2                       := ( 0.0, 2.561, 0.750 )
         Q.ON.BODY.1                       := ( 0.0, 2.561, 1.750 )
         Q.ON.BODY.2                       := ( 0.0, 2.561, 1.750 )
         R.ON.BODY.1                       := ( 1.0, 2.561, 0.750 )
         R.ON.BODY.2                       := ( 1.0, 2.561, 0.750 )
         NODE.1                            := '0'
         NODE.2                            := '0'
      UP
      CREATE TRANSLATIONAL.JOINT
         NAME                              := 'TR_OLEONOSE'
         BODY.1.NAME                       := 'INNER_CYLINDER'
         BODY.2.NAME                       := 'SLIDING_MEMBER'
         P.ON.BODY.1                       := ( 0.0, 2.561, 0.750 )
         P.ON.BODY.2                       := ( 0.0, 2.561, 0.350 )
         Q.ON.BODY.1                       := ( 0.0, 2.561, 1.750 )
         Q.ON.BODY.2                       := ( 0.0, 2.561, 1.350 )
         R.ON.BODY.1                       := ( 1.0, 2.561, 0.750 )
         R.ON.BODY.2                       := ( 1.0, 2.561, 0.350 )
         NODE.1                            := '0'
         NODE.2                            := '0'
      UP
      CREATE TRANSLATIONAL.JOINT
         NAME                              := 'TR_CRASHTUBE'
         BODY.1.NAME                       := 'MAIN_FITTING'
         BODY.2.NAME                       := 'INNER_CYLINDER'
         P.ON.BODY.1                       := ( 0.0, 2.561, 1.000 )
         P.ON.BODY.2                       := ( 0.0, 2.561, 0.750 )
         Q.ON.BODY.1                       := ( 0.0, 2.561, 2.000 )
         Q.ON.BODY.2                       := ( 0.0, 2.561, 1.750 )
         R.ON.BODY.1                       := ( 1.0, 2.561, 1.000 )
         R.ON.BODY.2                       := ( 1.0, 2.561, 0.750 )
         NODE.1                            := '0'
         NODE.2                            := '0'
      UP
      CREATE TRANSLATIONAL.JOINT
         NAME                              := 'TR_OLEOMAIN'
         BODY.1.NAME                       := 'UPPERMAIN'
         BODY.2.NAME                       := 'LOWERMAIN'
         P.ON.BODY.1                       := ( 0.0, 8.501, 1.543 )
         P.ON.BODY.2                       := ( 0.0, 8.293, 0.838 )
         Q.ON.BODY.1                       := ( 0.0, 8.796, 2.543 )
         Q.ON.BODY.2                       := ( 0.0, 8.588, 1.838 )
         R.ON.BODY.1                       := ( 1.0, 8.501, 1.543 )
         R.ON.BODY.2                       := ( 1.0, 8.293, 0.838 )
```

```
      NODE.1                                 := '0'
      NODE.2                                 := '0'
   UP
   UP
   CREATE BODY
      NAME                                   := 'GROUND'
      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                          := 'DEGREES'
      FLEXIBLE                               := 'FALSE'
      SUPERELEMENT                           := 'FALSE'
   UP
   CREATE BODY
      NAME                                   := 'FUSELAGE'
      CENTER.OF.GRAVITY                      := ( 0.0, 6.800, 2.793 )
      TYPE.ANGULAR.COORD                     := 'EULER'
      ANGLE.1                                := '0.0'
      ANGLE.2                                := '0.0'
      ANGLE.3                                := '0.0'
      FIXED.TO.GROUND                        := 'FALSE'
      MASS                                   := '8500'
      INERTIA.XXL                            := '50000'
      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                       := 'DEGREES'
    FLEXIBLE                            := 'FALSE'
    SUPERELEMENT                        := 'FALSE'
UP
CREATE BODY
    NAME                                := 'WHEELNOSE'
    CENTER.OF.GRAVITY                   := ( 0.0, 2.637, 0.232 )
    TYPE.ANGULAR.COORD                  := 'EULER'
    ANGLE.1                             := '0.0'
    ANGLE.2                             := '0.0'
    ANGLE.3                             := '0.0'
    FIXED.TO.GROUND                     := 'FALSE'
    MASS                                := '27.0'
    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                       := 'DEGREES'
    FLEXIBLE                            := 'FALSE'
    SUPERELEMENT                        := 'FALSE'
UP
CREATE BODY
    NAME                                := 'WHEELMAIN'
    CENTER.OF.GRAVITY                   := ( 0.0, 8.470, 0.413 )
    TYPE.ANGULAR.COORD                  := 'EULER'
    ANGLE.1                             := '0.0'
    ANGLE.2                             := '0.0'
    ANGLE.3                             := '0.0'
    FIXED.TO.GROUND                     := 'FALSE'
    MASS                                := '60.6'
    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                          := 'DEGREES'
      FLEXIBLE                               := 'FALSE'
      SUPERELEMENT                           := 'FALSE'
   UP
   CREATE BODY
      NAME                                   := 'SLIDING_MEMBER'
      CENTER.OF.GRAVITY                      := ( 0.0, 2.564, 0.350 )
      TYPE.ANGULAR.COORD                     := 'EULER'
      ANGLE.1                                := '0.0'
      ANGLE.2                                := '0.0'
      ANGLE.3                                := '0.0'
      FIXED.TO.GROUND                        := 'FALSE'
      MASS                                   := '11.4'
      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                          := 'DEGREES'
      FLEXIBLE                               := 'FALSE'
      SUPERELEMENT                           := 'FALSE'
   UP
   CREATE BODY
      NAME                                   := 'INNER_CYLINDER'
      CENTER.OF.GRAVITY                      := ( 0.0, 2.561, 0.750 )
      TYPE.ANGULAR.COORD                     := 'EULER'
      ANGLE.1                                := '0.0'
      ANGLE.2                                := '0.0'
      ANGLE.3                                := '0.0'
      FIXED.TO.GROUND                        := 'FALSE'
      MASS                                   := '1.0'
      INERTIA.XXL                            := '1.0'
      INERTIA.YYL                            := '1.0'
      INERTIA.ZZL                            := '1.0'
      INERTIA.XYL                            := '0.0'
      INERTIA.XZL                            := '0.0'
      INERTIA.YZL                            := '0.0'
      XG.FORCE                               := '0.0'
```

```
    YG.FORCE                              := '0.0'
    ZG.FORCE                              := '0.0'
    XL.TORQUE                             := '0.0'
    YL.TORQUE                             := '0.0'
    ZL.TORQUE                             := '0.0'
    CURVE.XGF                             := 'NONE'
    CURVE.YGF                             := 'NONE'
    CURVE.ZGF                             := 'NONE'
    CURVE.XLT                             := 'NONE'
    CURVE.YLT                             := 'NONE'
    CURVE.ZLT                             := 'NONE'
    SIGN.E0                               := 'POSITIVE'
    ANGULAR.UNITS                         := 'DEGREES'
    FLEXIBLE                              := 'FALSE'
    SUPERELEMENT                          := 'FALSE'
UP
CREATE BODY
    NAME                                  := 'DUMMY1'
    CENTER.OF.GRAVITY                     := ( 0.0, 2.561, 0.900 )
    TYPE.ANGULAR.COORD                    := 'EULER'
    ANGLE.1                               := '0.0'
    ANGLE.2                               := '0.0'
    ANGLE.3                               := '0.0'
    FIXED.TO.GROUND                       := 'FALSE'
    MASS                                  := '1.0'
    INERTIA.XXL                           := '1.0'
    INERTIA.YYL                           := '1.0'
    INERTIA.ZZL                           := '1.0'
    INERTIA.XYL                           := '0.0'
    INERTIA.XZL                           := '0.0'
    INERTIA.YZL                           := '0.0'
    XG.FORCE                              := '0.0'
    YG.FORCE                              := '0.0'
    ZG.FORCE                              := '0.0'
    XL.TORQUE                             := '0.0'
    YL.TORQUE                             := '0.0'
    ZL.TORQUE                             := '0.0'
    CURVE.XGF                             := 'NONE'
    CURVE.YGF                             := 'NONE'
    CURVE.ZGF                             := 'NONE'
    CURVE.XLT                             := 'NONE'
    CURVE.YLT                             := 'NONE'
    CURVE.ZLT                             := 'NONE'
    SIGN.E0                               := 'POSITIVE'
    ANGULAR.UNITS                         := 'DEGREES'
    FLEXIBLE                              := 'FALSE'
    SUPERELEMENT                          := 'FALSE'
UP
CREATE BODY
    NAME                                  := 'DUMMY2'
    CENTER.OF.GRAVITY                     := ( 0.0, 2.561, 0.800 )
    TYPE.ANGULAR.COORD                    := 'EULER'
    ANGLE.1                               := '0.0'
    ANGLE.2                               := '0.0'
    ANGLE.3                               := '0.0'
    FIXED.TO.GROUND                       := 'FALSE'
    MASS                                  := '1.0'
    INERTIA.XXL                           := '1.0'
    INERTIA.YYL                           := '1.0'
    INERTIA.ZZL                           := '1.0'
```

```
    INERTIA.XYL                        := '0.0'
    INERTIA.XZL                        := '0.0'
    INERTIA.YZL                        := '0.0'
    XG.FORCE                           := '0.0'
    YG.FORCE                           := '0.0'
    ZG.FORCE                           := '0.0'
    XL.TORQUE                          := '0.0'
    YL.TORQUE                          := '0.0'
    ZL.TORQUE                          := '0.0'
    CURVE.XGF                          := 'NONE'
    CURVE.YGF                          := 'NONE'
    CURVE.ZGF                          := 'NONE'
    CURVE.XLT                          := 'NONE'
    CURVE.YLT                          := 'NONE'
    CURVE.ZLT                          := 'NONE'
    SIGN.E0                            := 'POSITIVE'
    ANGULAR.UNITS                      := 'DEGREES'
    FLEXIBLE                           := 'FALSE'
    SUPERELEMENT                       := 'FALSE'
UP
CREATE BODY
    NAME                               := 'MAIN_FITTING'
    CENTER.OF.GRAVITY                  := ( 0.0, 2.561, 1.000 )
    TYPE.ANGULAR.COORD                 := 'EULER'
    ANGLE.1                            := '0.0'
    ANGLE.2                            := '0.0'
    ANGLE.3                            := '0.0'
    FIXED.TO.GROUND                    := 'FALSE'
    MASS                               := '26.6'
    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                      := 'DEGREES'
    FLEXIBLE                           := 'FALSE'
    SUPERELEMENT                       := 'FALSE'
UP
CREATE BODY
    NAME                               := 'TRAILINGARM'
    CENTER.OF.GRAVITY                  := ( 0.0, 8.292, 0.565 )
    TYPE.ANGULAR.COORD                 := 'EULER'
    ANGLE.1                            := '0.0'
    ANGLE.2                            := '0.0'
    ANGLE.3                            := '0.0'
    FIXED.TO.GROUND                    := 'FALSE'
```

```
    MASS                                 := '43.2'
    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                        := 'DEGREES'
    FLEXIBLE                             := 'FALSE'
    SUPERELEMENT                         := 'FALSE'
  UP
  CREATE BODY
     NAME                                := 'LOWERMAIN'
     CENTER.OF.GRAVITY                   := ( 0.0, 8.293, 0.838 )
     TYPE.ANGULAR.COORD                  := 'EULER'
     ANGLE.1                             := '0.0'
     ANGLE.2                             := '0.0'
     ANGLE.3                             := '0.0'
     FIXED.TO.GROUND                     := 'FALSE'
     MASS                                := '11.4'
     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                       := 'DEGREES'
     FLEXIBLE                            := 'FALSE'
     SUPERELEMENT                        := 'FALSE'
  UP
  CREATE BODY
     NAME                                := 'UPPERMAIN'
     CENTER.OF.GRAVITY                   := ( 0.0, 8.501, 1.543 )
     TYPE.ANGULAR.COORD                  := 'EULER'
```

```
   ANGLE.1                               := '0.0'
   ANGLE.2                               := '0.0'
   ANGLE.3                               := '0.0'
   FIXED.TO.GROUND                       := 'FALSE'
   MASS                                  := '24.2'
   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                         := 'DEGREES'
   FLEXIBLE                              := 'FALSE'
   SUPERELEMENT                          := 'FALSE'
UP
CREATE INITIAL.CONDITION
   NAME                                  := 'IC_DIS_OLEONOSE'
   BODY.1.NAME                           := 'INNER_CYLINDER'
   BODY.2.NAME                           := 'SLIDING_MEMBER'
   ELEMENT.NAME                          := 'NONE'
   TYPE.INITIAL.COND                     := 'DISTANCE'
   INITIAL.VALUE                         := '0.400'
   TIME.DERIVATIVE                       := '0.0'
   OMEGA.Y                               := '0.0'
   OMEGA.Z                               := '0.0'
   P.ON.BODY.1                           := ( 0.0, 2.561, 0.750 )
   P.ON.BODY.2                           := ( 0.0, 2.561, 0.350 )
   EXTRA.COORD                           := '0'
   ANGULAR.UNITS                         := 'DEGREES'
UP
CREATE INITIAL.CONDITION
   NAME                                  := 'IC_DIS_CRASHTUBE'
   BODY.1.NAME                           := 'INNER_CYLINDER'
   BODY.2.NAME                           := 'MAIN_FITTING'
   ELEMENT.NAME                          := 'NONE'
   TYPE.INITIAL.COND                     := 'DISTANCE'
   INITIAL.VALUE                         := '0.250'
   TIME.DERIVATIVE                       := '0.0'
   OMEGA.Y                               := '0.0'
   OMEGA.Z                               := '0.0'
   P.ON.BODY.1                           := ( 0.0, 2.561, 0.750 )
   P.ON.BODY.2                           := ( 0.0, 2.561, 1.000 )
   EXTRA.COORD                           := '0'
   ANGULAR.UNITS                         := 'DEGREES'
UP
CREATE INITIAL.CONDITION
   NAME                                  := 'IC_DIS_OLEOMAIN'
```

```
   BODY.1.NAME                   := 'UPPERMAIN'
   BODY.2.NAME                   := 'LOWERMAIN'
   ELEMENT.NAME                  := 'NONE'
   TYPE.INITIAL.COND             := 'DISTANCE'
   INITIAL.VALUE                 := '0.73504'
   TIME.DERIVATIVE               := '0.0'
   OMEGA.Y                       := '0.0'
   OMEGA.Z                       := '0.0'
   P.ON.BODY.1                   := ( 0.0, 8.501, 1.543 )
   P.ON.BODY.2                   := ( 0.0, 8.293, 0.838 )
   EXTRA.COORD                   := '0'
   ANGULAR.UNITS                 := 'DEGREES'
UP
CREATE INITIAL.CONDITION
   NAME                          := 'IC_DIS_RETRACNOSE'
   BODY.1.NAME                   := 'MAIN_FITTING'
   BODY.2.NAME                   := 'FUSELAGE'
   ELEMENT.NAME                  := 'NONE'
   TYPE.INITIAL.COND             := 'DISTANCE'
   INITIAL.VALUE                 := '0.9413825'
   TIME.DERIVATIVE               := '0.0'
   OMEGA.Y                       := '0.0'
   OMEGA.Z                       := '0.0'
   P.ON.BODY.1                   := ( 0.0, 2.660, 1.311 )
   P.ON.BODY.2                   := ( 0.0, 3.600, 1.260 )
   EXTRA.COORD                   := '0'
   ANGULAR.UNITS                 := 'DEGREES'
UP
CREATE INITIAL.CONDITION
   NAME                          := 'IC_Y_WHEELMAIN'
   BODY.1.NAME                   := 'GROUND'
   BODY.2.NAME                   := 'WHEELMAIN'
   ELEMENT.NAME                  := 'NONE'
   TYPE.INITIAL.COND             := 'Y.DIFF'
   INITIAL.VALUE                 := '8.470'
   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, 8.470, 0.413 )
   EXTRA.COORD                   := '0'
   ANGULAR.UNITS                 := 'DEGREES'
UP
CREATE INITIAL.CONDITION
   NAME                          := 'IC_Z_WHEELMAIN'
   BODY.1.NAME                   := 'GROUND'
   BODY.2.NAME                   := 'WHEELMAIN'
   ELEMENT.NAME                  := 'NONE'
   TYPE.INITIAL.COND             := 'Z.DIFF'
   INITIAL.VALUE                 := '0.334'
   TIME.DERIVATIVE               := '-11.0'
   OMEGA.Y                       := '0.0'
   OMEGA.Z                       := '0.0'
   P.ON.BODY.1                   := ( 0.0, 0.0, 0.0 )
   P.ON.BODY.2                   := ( 0.0, 8.470, 0.413 )
   EXTRA.COORD                   := '0'
   ANGULAR.UNITS                 := 'DEGREES'
UP
CREATE INITIAL.CONDITION
   NAME                          := 'IC_Z_WHEELNOSE'
```

```
    BODY.1.NAME                          := 'GROUND'
    BODY.2.NAME                          := 'WHEELNOSE'
    ELEMENT.NAME                         := 'NONE'
    TYPE.INITIAL.COND                    := 'Z.DIFF'
    INITIAL.VALUE                        := '0.258'
    TIME.DERIVATIVE                      := '-11.0'
    OMEGA.Y                              := '0.0'
    OMEGA.Z                              := '0.0'
    P.ON.BODY.1                          := ( 0.0, 0.0, 0.0 )
    P.ON.BODY.2                          := ( 0.0, 2.637, 0.232 )
    EXTRA.COORD                          := '0'
    ANGULAR.UNITS                        := 'DEGREES'
UP
CREATE POINT.OF.INTEREST
    NAME                                 := 'RADAR_P'
    BODY.NAME                            := 'FUSELAGE'
    P.ON.BODY                            := ( 0.0, 4.612, 0.555 )
    NODE                                 := '0'
UP
CREATE CURVE
    NAME                                 := 'PRESSCU'
    TYPE.DATA                            := 'PAIRED.XY'
    SLOPE.LEFT                           := '0.0'
    SLOPE.RIGHT                          := '0.0'
    SCALE.X                              := '1.0E6'
    SCALE.Y                              := '1.0E-6'
    START.X                              := '0.0'
    INCREMENT.X                          := '0.0'
    INTERPOLATION                        := 'LINEAR'
    CYCLIC                               := 'FALSE'
    DATA
      0.0000000000E+00 0.0000000000E+00  17.00000000     0.0000000000E+00
       19.00000000      150.0000000       50.00000000     150.0000000
    ENDDATA
UP
CREATE CURVE
    NAME                                 := 'EMERCU'
    TYPE.DATA                            := 'PAIRED.XY'
    SLOPE.LEFT                           := '0.0'
    SLOPE.RIGHT                          := '0.0'
    SCALE.X                              := '1.0E6'
    SCALE.Y                              := '1.0E-6'
    START.X                              := '0.0'
    INCREMENT.X                          := '0.0'
    INTERPOLATION                        := 'LINEAR'
    CYCLIC                               := 'FALSE'
    DATA
      0.0000000000E+00 0.0000000000E+00  23.00000000     0.0000000000E+00
       28.00000000      100.0000000       30.00000000     100.0000000
       35.00000000      499.0000000       50.00000000     499.0000000
    ENDDATA
UP
```

APPENDIX B INITIAL CONDITIONS

Using initial condition elements in DADS it is possible to define an initial value to the global coordinates (x, y, z) of a body and its time derivatives. It is also possible to define the difference between two coordinates or the distance between two bodies. For the definition of the initial angular positions in DADS the use of Euler parameters is required (cf. Sauren [10]). When a set of global initial conditions is used, precisely defined coordinates are required to translate the model to the correct position. To simulate each different landing cases it is necessary to define a different set of coordinates. It is best to use initial conditions of the type "distance". By using these initial conditions it is possible to lay down the length of the TSDA elements (representing the shock absorbers and the retract actuator in the nose) so that during the assembly proces the model can be treated as a rigid body. Now the model (with a prescribed initial velocity) can be positioned with the use of a select number of global or distance initial conditions or in combination with initial angular positions.

Assembly proces

Before starting simulations the model has to be assembled. This is realised by solving the system of constraint equations which consists of kinematic and driving constraints. (cf. Haug [6]) Since these equations are highly non-linear a Newton Raphson iteration is used. When initial conditions are applied to change the position of the model the set of constraint equations will be revised. The original position (generated in the input file) is no longer the solution for the revised constraint equations. In this case more iterations are needed to solve these revised constraint equations.

The following example explaines an other problem which can occur if large displacements are defined.

To prescribe an initial displacement of 0.5 m in positive z direction for body 1 and 2 (see figure B.1) the following type of initial condition elements can be used:

- "distance 0.2 m." to describe the distance between
  body 1 and 2
- "Z 0.5 m." to describe the global z-coordinate of
  body 1.

Using these elements results in two solutions as shown in figure B and C. Figure C shows a solution which is not preferred but can occur during the assembly proces.

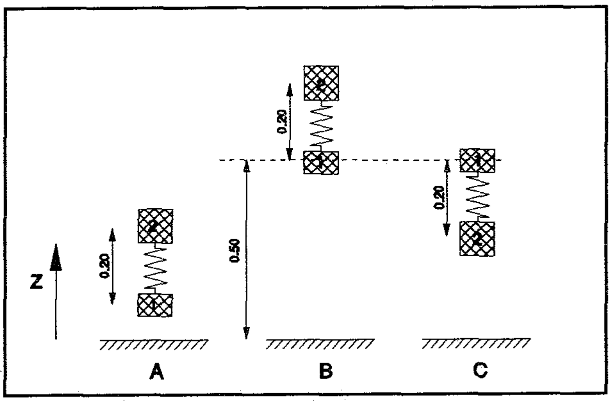


Figure B.1

Restart procedure

With the use of a restart procedure problems with the initial conditions described above can be evaded. This procedure can be described as follows.

The dynamic model is extended with a restart element. This element can be used to read the initial positions and velocities for all bodies in the model from a restart file. The only problem is to create a restart file which represents the initial conditions of the model properly. When a dynamic analysis is executed with a dummy model in which the complete model behaves as a "rigid body" (which can be realised by using distance constraints), the model can be positioned correctly with the use of driver elements. The restart file which results from this can be used as input for the dynamic analysis of the real model.

The only problem in this way is that it is not possible to prescribe the initial velocities for the bodies. Therefore it is necessary to manually edit the restart file so that the initial velocities comply with the values required.

Using this procedure it is possible to position the model in any desired position. An advantage is that this model can be assembled more precisely with regard to an assembly which is obtained by using initial condition elements. A disadvantage of this procedure is that it requires much labour, introduces possible errors and it is necessary to make an additional dummy model.