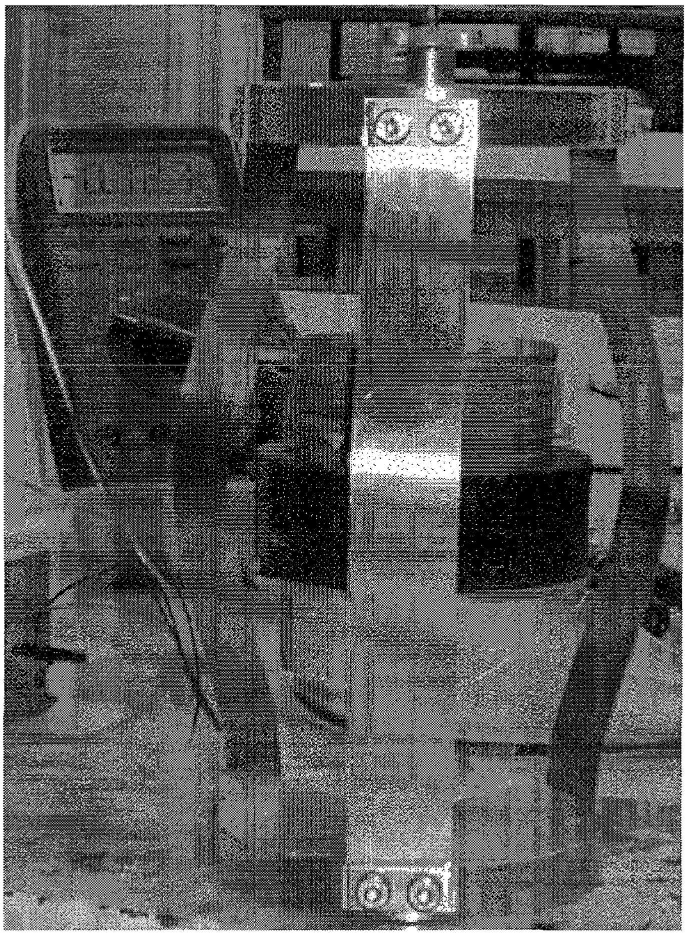
Analysis of a non-linear system

M.P.M. Hendriks s456643

DCT 2002.62

/Technical University Eindhoven
/Mechanical Engineering
/Dynamics and Control

Analysis of a non-linear system



M.P.M.Hendriks s456643

Attendants: Prof.dr. H.Nijmeijer
Ir. L.Kodde
Dr.ir. R.Fey

/Eindhoven, 14 October 2002

Summary

Introduction

In this report the investigation of a non-linear mass-spring system will be given. Both static and dynamic analysis is done and reasons for differences between theory and reality are given.

Chapter 1 Determination of the buckling force

General derivation of the buckling force of a beam, which is loaded by a compressive force, will lead to the buckling force of the system with help of the specific boundary conditions. The buckling force of one leaf spring is equal to 5.45N. The buckling force of the whole system will be $3 \cdot 5.45 = 16.35$ N, because the three leaf springs are coupled parallel.

Chapter 2 Static experiment

With help of a displacement sensor the deflection curve of the system can be determined. First the sensor is calibrated with help of least square fit. After that the upper mass is enlarged and the corresponding voltage is read. The buckling force of the experiment will be smaller than the one obtained in the theoretical calculations. The leaf springs will loose some of their stiffness after using them several times. Theoretical analyses however, assume a perfect geometry.

Chapter 3 Static simulation

A static simulation using the finite element method program Marc/ Mentat will check results of theoretical calculations and static experiment. Three models are made and results and differences are discussed. The results of the best model will equal the theoretical analysis.

Chapter 4 Dynamic numerical analysis

The deflection curve obtained in the static experiment is used to make a model of the system using the program Matlab. First this is done without friction and after that with friction included. The algorithms, which are used to make the model, are discussed and results can be seen in this chapter.

Chapter 5 Dynamic experiment

The results of the numerical analysis can be checked by a dynamic experiment. In this experiment a acceleration sensor is placed at the top of the upper mass and after the system is brought into an oscillation the acceleration is measured in time. The results of dynamic experiment and numerical analysis are nearly the same. Differences are caused by sensitivity of the springs and approximations in the numerical analysis.

Chapter 6 Recommendations

To avoid some of the differences between theoretical analysis and reality the following recommendations are given:

- Suitable construction
- Determination of friction with help of a filter
- 3-D model using Marc/ Mentat
- 3-D model using Matlab
- Comparison reality-Marc-Matlab

Conclusions

Both static and dynamic analyses are done and satisfying results are found. Differences between theoretical analysis and reality are caused by sensitivity of the leaf springs and approximations in the models.

Table of contents

Introduction

In this report a non-linear mass-spring system will be investigated. First of all the static behaviour of the spring will be analysed theoretically and discussed. After that, a *statical* experiment will be done in order to obtain the non-linear behaviour of the spring. Then a simulation of the non-linear behaviour of the spring will be made using the computer program MARC. The results will be compared with those obtained in the *statical* experiment and in the theoretical calculations. Next this static behaviour is used to make a dynamic model of the system in order to analyse the *dynamic* properties of the system. The dynamical results of this mathematical model, which is made using the computer program MATLAB, are compared with the results of a *dynamical* experiment. Finally some recommendations are made and conclusions are drawn.

1. Determination of the buckling force

The system, that is investigated, can be seen in *figure 1.1*. Two cylindrical masses are connected by three leaf-springs. The angle between every two springs is 120 degrees, which means that the system is axial symmetric. The lower mass is placed at a flat ground, so it can be seen as 'fixed to the world'. The weight of the upper mass is too large to be held by the leaf springs, so they will buckle. After buckling the resultant vertical force of the leaf springs increases. So the force of the leaf springs together will compensate first and then get larger than the gravity force of the upper mass during their buckling. The upper mass will be slowed down and eventually be pushed upwards by the leaf springs. So when the upper mass is moved upwards by a person, until the springs are fully stretched, and then released, it will make an oscillation, which is not sinusoidal. The upper mass will be guided by a pin, which is connected to the lower mass and enters a centered hole in the upper mass. The cause of this non-sinusoidal oscillation is the fact that the deflection curve is non-linear. In this report this relation is investigated.

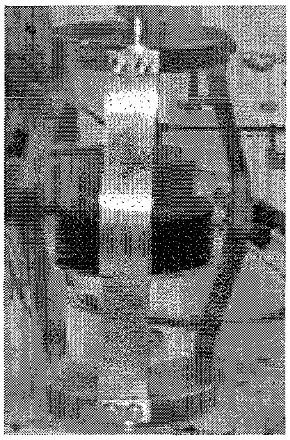


figure 1.1 System

First the buckling force of one leaf spring is investigated, in other words: The force at which a leaf spring will just buckle. A general derivation of the buckling force of a beam, which is buckled by a compressive force, is given. Then, with the specific boundary conditions, the buckling force of the under investigation leaf springs is determined.

The potential energy of a beam, which is buckled by a compressive force (see *figure 1.2*), is:

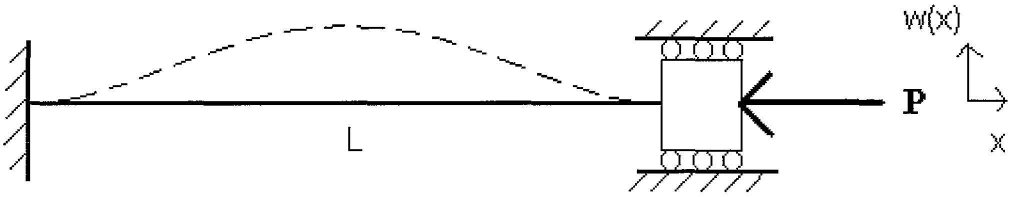


figure 1.2 Buckling by compressive force

$$U = \int_0^L \frac{1}{2} EIw^2{}_{,xx} dx - \int_0^L \frac{1}{2} Pw^2{}_{,x} dx \tag{1.1}$$

in which:

E	=	E-modulus of beam
I_s	=	Second moment of area of the cross section of the beam
w	=	Position of the beam at point *x*
L	=	Length of the beam
P	=	Compressive force

At the buckling point the system will be in equilibrium, the potential energy will have its optimum and the change of the potential energy will thus be equal to zero:

$$\partial U = \int_0^L EIw_{,xx} \partial w_{,xx} dx - \int_0^L P_b w_{,x} \partial w_{,x} dx = 0 \tag{1.2}$$

In which P_b = *Buckling force*

This results in the following differential equation (the derivation is shown in *Appendix 1*):

$$EIw_{,xxxx} + P_b w_{,xx} = 0 \tag{1.3}$$

with solution (for derivation see *Appendix 2*):

$$w(x) = c + dx + e\sin(\alpha x) + f\cos(\alpha x) \tag{1.4}$$

$$\alpha^2 = \frac{P_b}{EI} \tag{1.5}$$

The constant values of α and *c*, *d*, *e* and *f* have to be determined with help of the boundary conditions of the leaf springs in the system. The configuration for one leaf spring of the system is shown in *figure 1.2*: Both ends are fixed and the following boundary conditions are valid:

$$\begin{aligned} w(x=0) &= w(x=L) = 0 \\ w_{,x}(x=0) &= w_{,x}(x=L) = 0 \end{aligned} \tag{1.6}$$

When these values are substituted in the found solution, four equations arise which can be written in the following form:

$$A_l u = 0 \tag{1.7}$$

$$A_l = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & L & \sin(\alpha L) & \cos(\alpha L) \\ 0 & 1 & \alpha & 0 \\ 0 & 1 & \alpha\cos(\alpha L) & -\alpha\sin(\alpha L) \end{bmatrix} \qquad u = \begin{bmatrix} c \\ d \\ e \\ f \end{bmatrix}$$

A non-trivial solution for this "eigenvalue"-problem will only exist if *det(A$_l$)=0.* This leads to the solution: $\alpha = \frac{2\pi}{L}$ (The values of *c, d, e* and *f* are not of any importance here for determining the buckling force).

From: $\frac{P_b}{EI_s} = \alpha^2$ follows: $P_b = \alpha^2 EI_s = \frac{4\pi^2 EI_s}{L^2}$ (1.8)

In this system (with leaf springs) $I_s = \frac{1}{12} bh^3$ (1.9)

In the system leaf springs are made of steel and while both ends are fixed in a groove of 20mm the width *b* of the leaf springs also is 20mm. Thus only the length *L* and the thickness *h* of the leaf springs can be varied. In the *figures 1.3 a* and *b* the buckling force of one leaf spring can be seen dependent on length and thickness, respectively. The E-modulus and width are held constant at 200 GN/m^2 and 20 mm respectively.

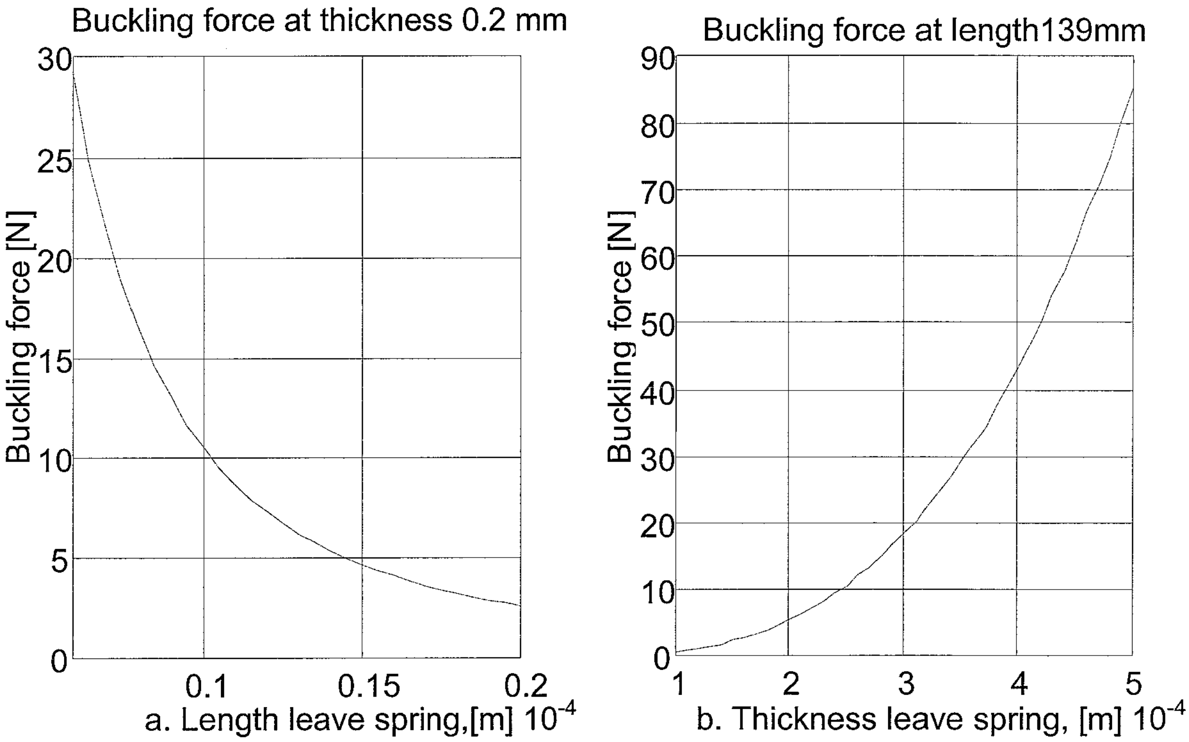


figure 1.3 Buckling force dependent on Length (a) and Thickness (b)

In the system the leaf springs have the following dimensions: Length L=139 mm, width b=20mm and thickness h=0.2 mm. In *figure 1.4* one leaf spring with its dimensions is shown.

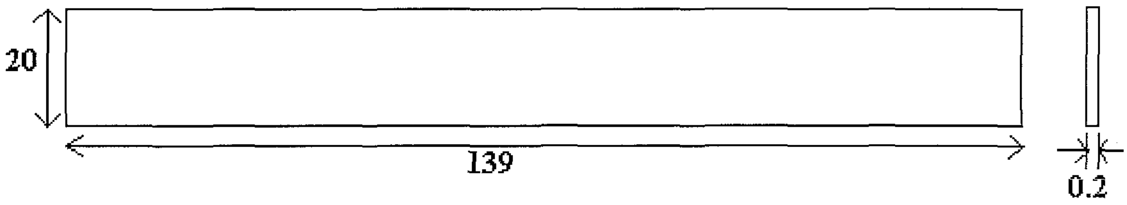


figure 1.4 Leaf spring dimensions (in mm)

Now the buckling force for this specific leaf spring can be calculated:

$$I_s = \frac{1}{12} bh^3 = \frac{1}{12} \cdot 20 \cdot 10^{-3} \cdot (0.2 \cdot 10^{-3})^3 m^4$$

$$P_b = \frac{4\pi^2 EI}{L^2} = 5.45N$$

The three leaf springs are coupled parallel (see *figure 1.1*), so the total force of them can be calculated by summing the separate forces. While the leaf springs are all the same it's evident that the weight of the upper mass has to exceed: $3 \cdot 5{,}45 = 16.35N$, resulting in an upper mass of *1.76 kg* to get the system buckled.

2. Static Experiment

In chapter 1 the buckling force of the system has been determined. When this is translated to a force-displacement-diagram it is the point on the y-axis of this diagram, see *figure 2.1.* The displacement of the upper point of the leaf springs is approximately equal to zero when the leaf springs are fully stretched. Actually there is a small displacement *u* of the upper point at the buckling point:

$$u = \frac{P_b}{3k} = \frac{P_b L}{3EA} = \frac{P_b L}{3Ebh} = \frac{16.35 \cdot 13.9 \cdot 10^{-2}}{3 \cdot 200 \cdot 10^9 \cdot 2 \cdot 10^{-2} \cdot 2 \cdot 10^{-4}} = 9 \cdot 10^{-3}\,mm$$

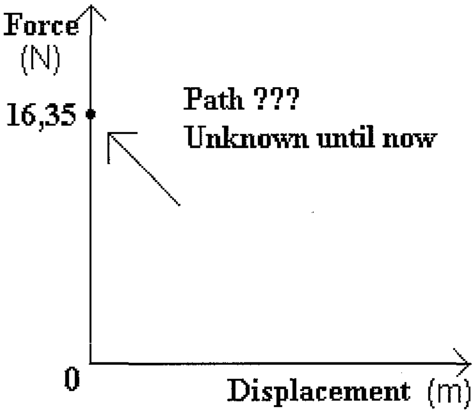


figure 2.1 *Buckling force in diagram*

When a mass with weight lower than this buckling force is connected at the upper end of the leaf springs they will not buckle. Also when the system is then manually buckled and released the leaf springs will automatically stretch and stay in stretched position. When an upper mass is used with a weight greater than the buckling force the system will buckle. The interesting part now is the force, which the leaf springs produce together, when the system is buckled further. To determine the force of the leaf springs dependent on the displacement of the upper point of the leaf springs, in other words the force-displacement diagram, a statical experiment is done. By putting extra weight on it the system will buckle further to compensate the weight. When equilibrium is found the force of the leaf springs together is equal to the weight of the mass. It can be concluded that the force of the leaf springs will increase when they buckle further! The accompanying displacement is measured with a displacement sensor (see *figure 2.2).*

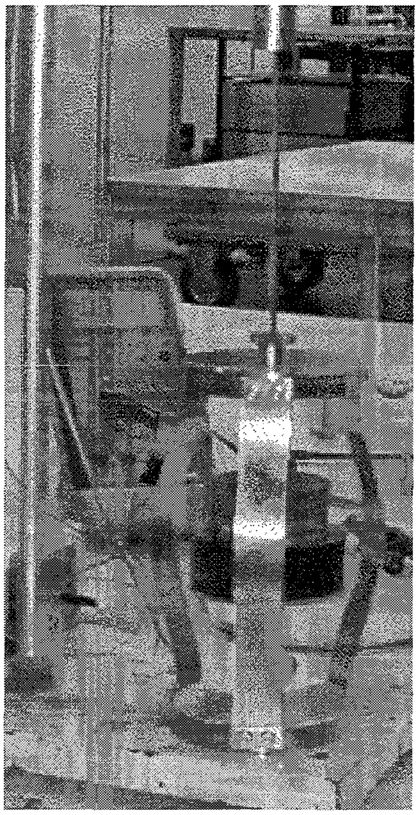


figure 2.2 Adapted system with displacement sensor

To carry out this experiment the system is adapted (see *figure 2.2)*: The upper mass is replaced by a mass of smaller weight and under it a screw thread is fixed. At the end of this thread a platform is fixed on which the extra weights can be placed. The displacement sensor can be connected to the upper face of the upper weight and needn't to be replaced every time, which should have if the weights were placed upon the mass!
The sensor supplies a voltage, which is a measure for the displacement of the object it's connected to. It is known that the relation between the voltage and displacement is linear, but the specific formula is not known. So first the sensor has to be calibrated. In order to do that the system without extra masses (leaf springs fully stretched) is placed underneath the sensor. The voltage at this situation is related to the displacement zero. After that, blocks of known heights are placed underneath the sensor and the difference in height between system and block is the displacement going with the supplied voltage (see *figure 2.3)*.

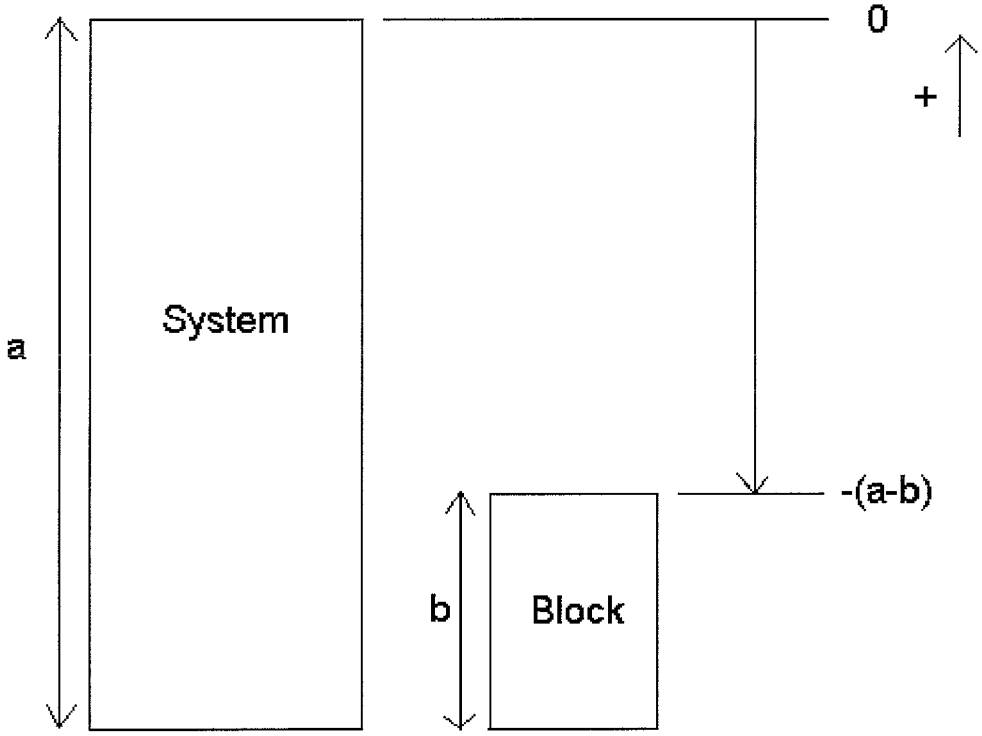


figure 2.3 Calibrating displacement sensor

The following values are found (*table 2.1*):

Voltage (V)	Displacement (m)10^{-3}
-5.53	-40
-5.07	-36.7
-4.63	-33.3
-4.29	-30.1
-3.73	-26.6
-3.28	-22.9
-2.79	-19.8
-2.38	-16.2
-1.90	-13.2
-1.44	-9.6
-0.86	-4.5
-0.42	-0.9
-0.21	0
0.04	2.1
0.51	4.8
1.44	12.3

table 2.1

Because it is known that the relationship between voltage and displacement is linear the following can be formulated:

$$y_i = Ax_i + B \qquad (2.1)$$

in which:

y_i = *displacement(m) at measurement i*

x_i = *voltage(V) at measurement i*

To get a nice fit a least-square method is used: $\frac{1}{2}\sum_{i=1}^{n}(y_i - (Ax_i + B))^2$ (2.2)

This equation depends on two parameters, namely *A* and *B*, when the measurements are known. Values for *A* and *B* have to be chosen in such a way that above sum is as small as possible. So a minimum has to be found: The derivatives to *A* and *B* have to be zero. The calculations can be found in *Appendix 3.*

The solution of this equation is $\vec{z} = \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} 0.0075 \\ 0.0015 \end{bmatrix}$

The values of the measurements and the fit are plotted in the same figure as can be seen below (see *figure 2.4*). The M-file for this can be found in *Appendix 4, A.4.1.*

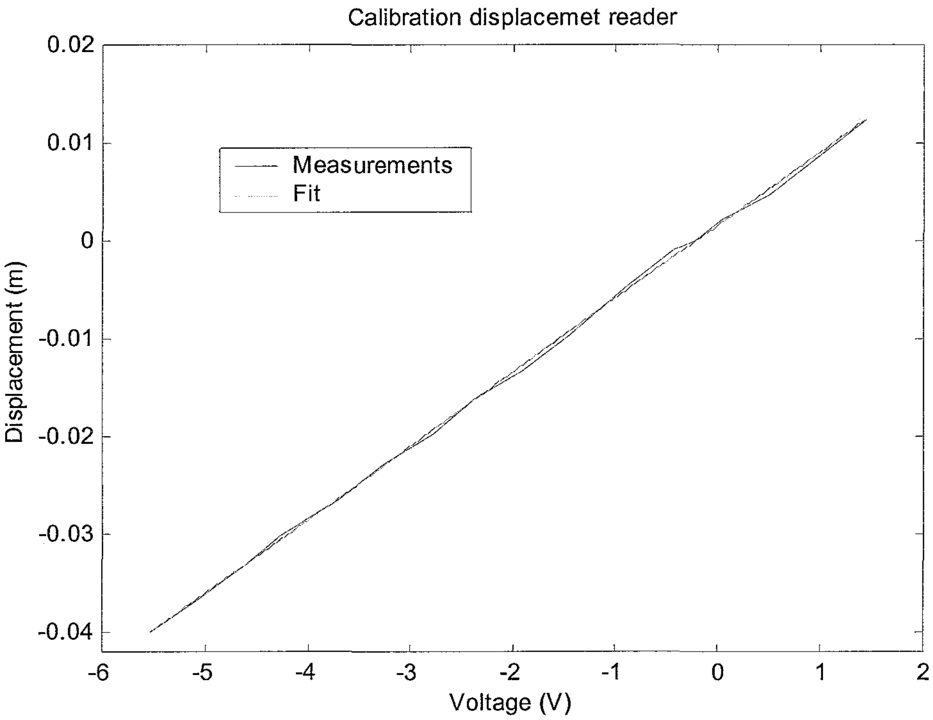


figure 2.4 *Calibration displacement sensor*

Now the displacement sensor has been calibrated, the real experiment can be done: Again and again, the weight on the leaf springs is enlarged and when force equilibrium is found the voltage is read each time. When this is done for 57 different weights the voltages are translated into displacements and a force-displacement diagram can be drawn. This can be seen in *figure 2.5* below.

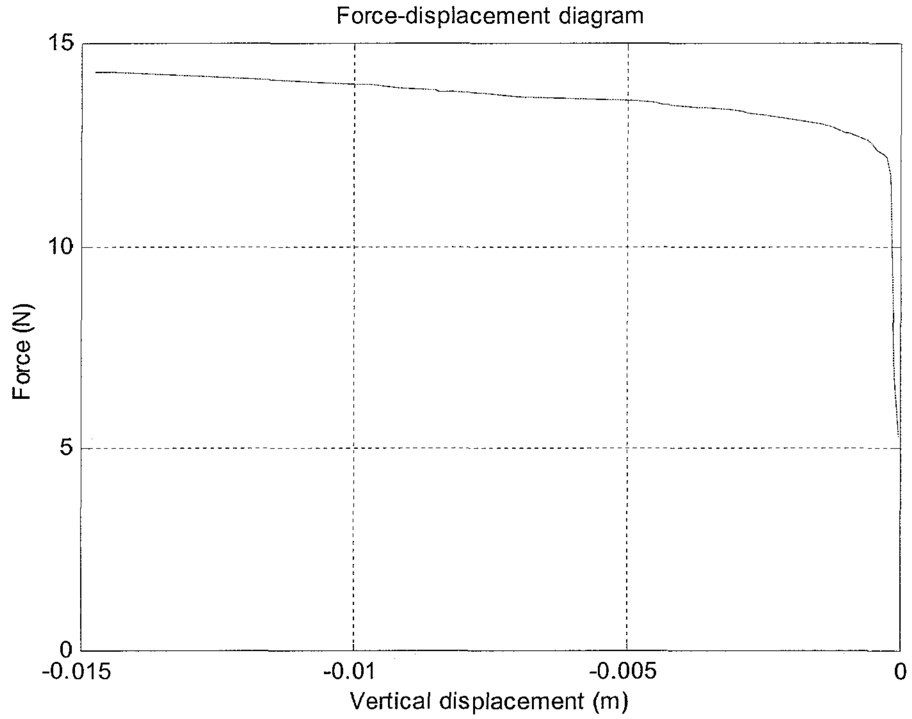


figure 2.5 *Force-displacement diagram, system*

Because the three leaf springs are connected parallel, the force diagram of one leaf spring will be the following (all forces divided by three, see *figure 2.6*).

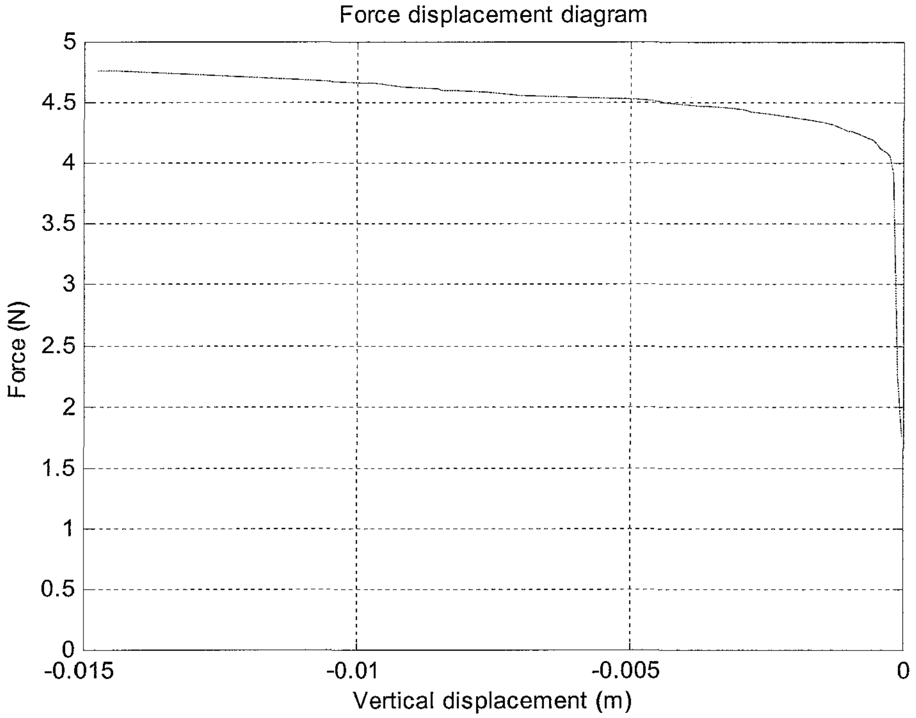


figure 2.6 Force-displacement diagram, one leaf spring

From *figure 2.5* the buckling force of the system can be determined: it will be somewhere near 13 Newton. The theoretical value, computed in chapter one (16.35 N), exceeds this value. A reason for this is the fact that the leaf springs are very sensitive and after some buckling they will loose some stiffness. Theoretical analysis, however, assume a perfect geometry. The experimental diagram will lie closer to the x-axis and so will the buckling force!

3. Static simulation

In chapter two a force-displacement diagram for the system is determined experimentally. At the end of chapter two, the discrepancy between theoretical value and experimental force-displacement curve is explained. In this chapter a numerical analysis of the force-displacement curve is carried out using the finite element package Marc/ Mentat. A simulation of a loaded 3-D construction can be built and the deformations can be visualized together with the reaction forces.
The program consists of several menu's, of which the relevant ones will be mentioned here. Different models are made to simulate the leaf spring. The one with the best result is considered below, two alternatives together with their differences are discussed later.

3.1 Finite element model

Mesh generation
In this menu the finite element mesh of the structure has to be built. In the case of the leaf spring four nodes have to be drawn: The coordinates of the four corner points of the front view (see *figure 3.1*) of the leaf spring have to be given here. After that the four nodes have to be connected by adding a quad-element. So the leaf spring then is considered as one element. Next this element can be subdivided in smaller elements. In this case a one-by-fourteen subdivision is chosen (see *figure 3.1*). The element size is $20mm$ x $10mm$.

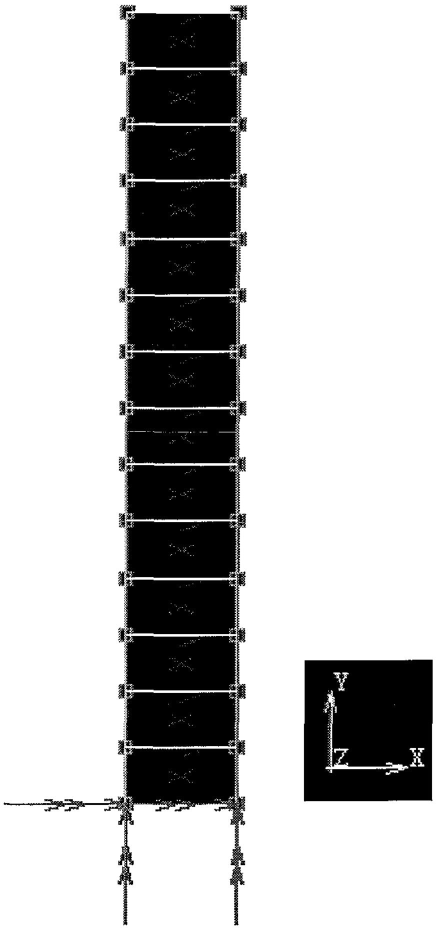


figure 3.1 *Mesh generation*

Kinematic and dynamic boundary conditions

Bottom

The bottom of the leaf spring is fixed and cannot move in any direction. Therefore the two nodes at the bottom of the simulation model are fixed in all degrees of freedom (3 translations and 3 rotations).

Top

The top of the leaf spring will, in the perfect case, move only in vertical (y) direction. Therefore the upper two nodes of the simulation model are fixed in all degrees of freedom but one: The translation in vertical direction.

Movement

In the experiment a force (mass) will cause a vertical displacement of the top of the leaf spring. In this numerical simulation the vertical displacement of the top of the leaf spring will be prescribed. The displacement in vertical direction of the upper two nodes is incrementally increased (see *figure 3.2*). In this the x-axis agrees with 100 increments and the y-axis with a displacement from zero to 10 mm in negative y-direction. To achieve this displacement a vertical force will be needed, which is as large as the vertical (reaction) force generated by the spring. So at the top or the bottom of the spring the reaction forces can be used to get the force displacement diagram.

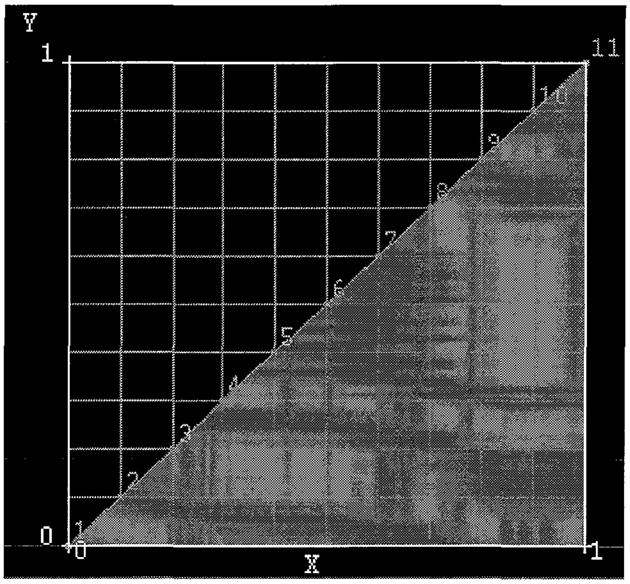


figure 3.2 Prescribed displacement(y-axis)-increment(x-axis) diagram

Sidepressure

When only the upper three boundary conditions would be used the spring would be compressed until the buckling point. At the buckling point numerical problems may be expected because the stiffness matrix becomes singular. To circumvent this problem a small initial imperfection is needed to force the system into a buckling form. To accomplish this a small side pressure is introduced at the front view of the spring. This pressure will decrease quickly (see *figure 3.3*) so it will not be of any influence later on. The y-axis represents pressures from zero to 100 N/m^2 and the x-axis represents 100 increments. It is made sure the buckling force has been exceeded before the sidepressure has gone to zero.

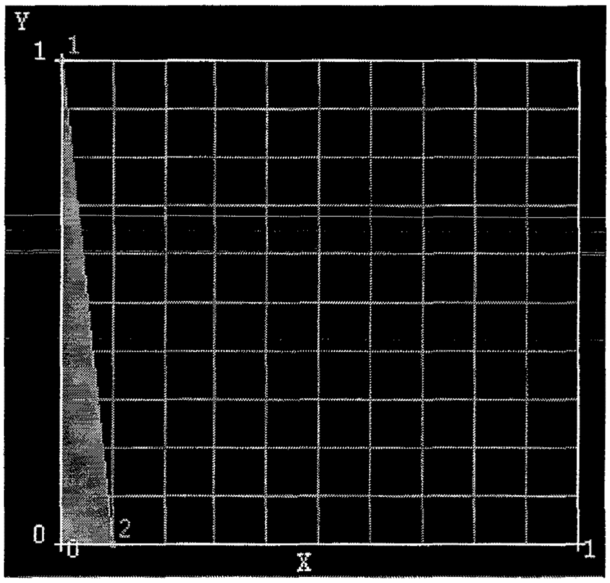


figure 3.3 *Sidepressure (y-axis)-increment (x-axis) diagram*

Material properties
The leaf springs are made of steel and therefore the material properties are as follows:
$E = 200 \cdot 10^9 N/m^2, \nu = 0.3,\ \rho = 7850 kg/m^3$.

Element type
As mentioned before the leaf spring will be treated as a thin shell. Therefore element type 75 is chosen as type for this simulation.

Geometric properties
The thickness of the shell elements is 0.2mm.

Number of increments
100 equidistant increments are used.

Results
A force-displacement-diagram is wanted so as a result the reaction forces of the upper two nodes have to be selected and plotted against the displacement. After that, the two diagrams, which are equal, have to be summed to get the diagram of the leaf spring. This leads to the following result (see *figure 3.4*).

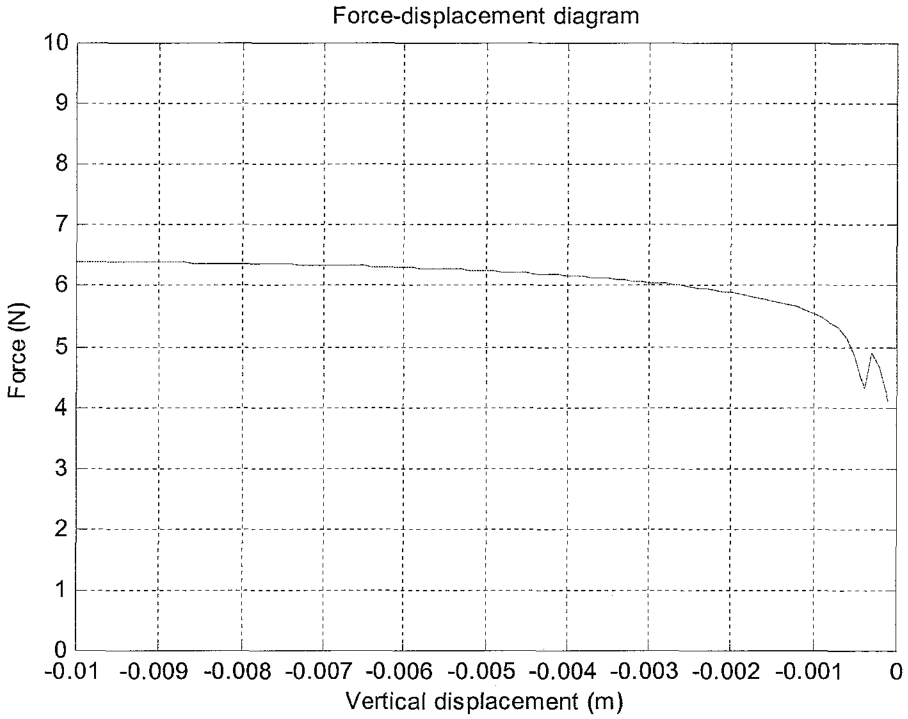


figure 3.4 Force-displacement diagram

The form of this diagram is the same as the one obtained at the experiment. However there are two differences:

1. The values of the forces at any displacement are greater than those in the experiment at the same displacement. A reason for this is already given: The geometry of the leaf springs in practice will not be perfect and moreover the springs are used several times, which could have caused some local plastic deformation. As can be seen in the figure the buckling force will be about 5.5 Newton. This value agrees with the analytical buckling value: 5.45 Newton.
2. The diagram shows a strange peak at the beginning. Local buckling could cause this.

3.2 ALTERNATIVES

With use of symmetry

In this file, only the left half of the leaf spring will be modeled: The width is 10 mm instead of 20 mm. The points at the right side of this model (in the whole spring these are the points in the middle of it (see *figure 3.5*)) can be supplied with symmetry conditions: The can only translate in Y and Z direction and rotate around the X-axis.

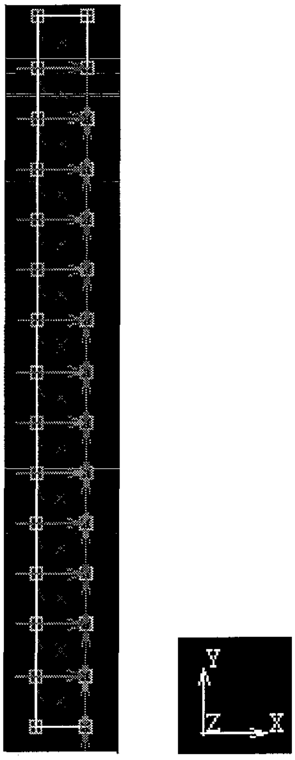


figure 3.5 *Symmetry*

The program won' t finish its run, but will get stuck at some time. The results until then can be shown.

To obtain the total force the sum of the force on the right node and twice the one on the left is taken (see *figure 3.6*).

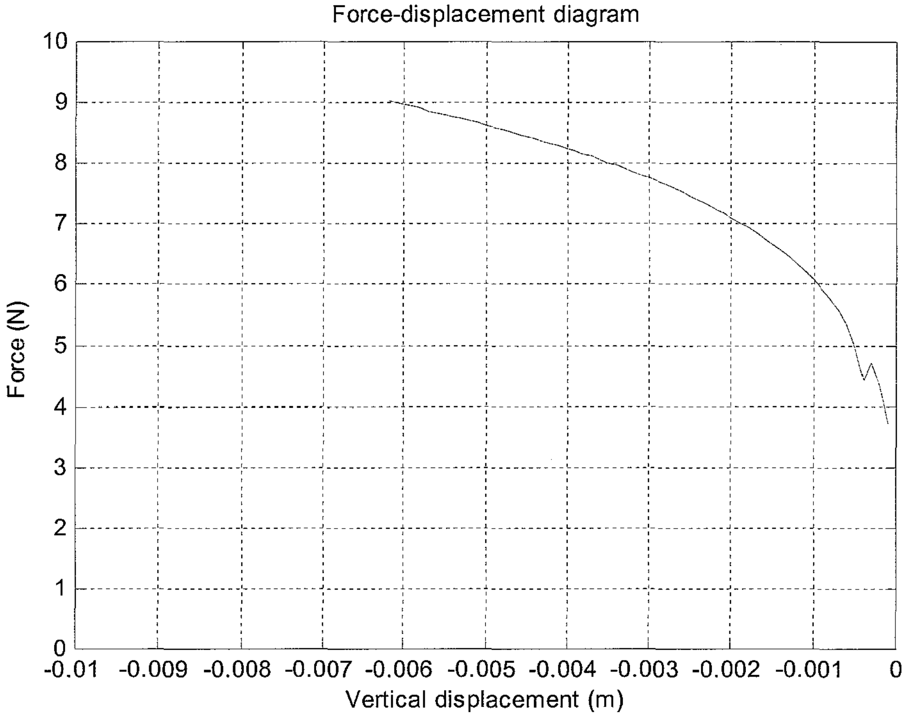


figure 3.6 Force-displacement diagram

From this figure it can be concluded this buckling force is roughly the same as the one obtained in the simulation before. However the path is steeper and will not fit the experiment data as good as the path obtained in the simulation before.

With pressure holding on

In the first simulation some sidepressure at the front view of the leaf spring is introduced. This sidepressure disappears after 10 iterations. When the pressure holds on longer (see *figure 3.7*) the result will be as can be seen in *figure 3.8*.

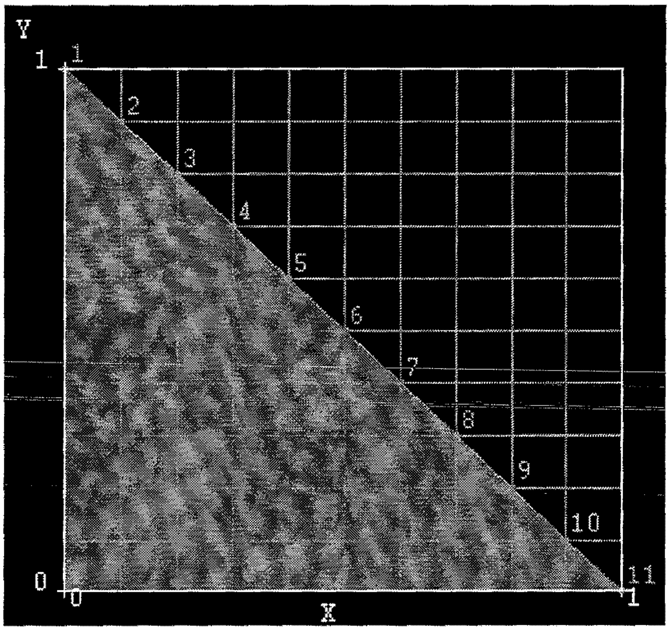


figure 3.7 *Pressure(y-axis)-increment(x-axis) diagram*

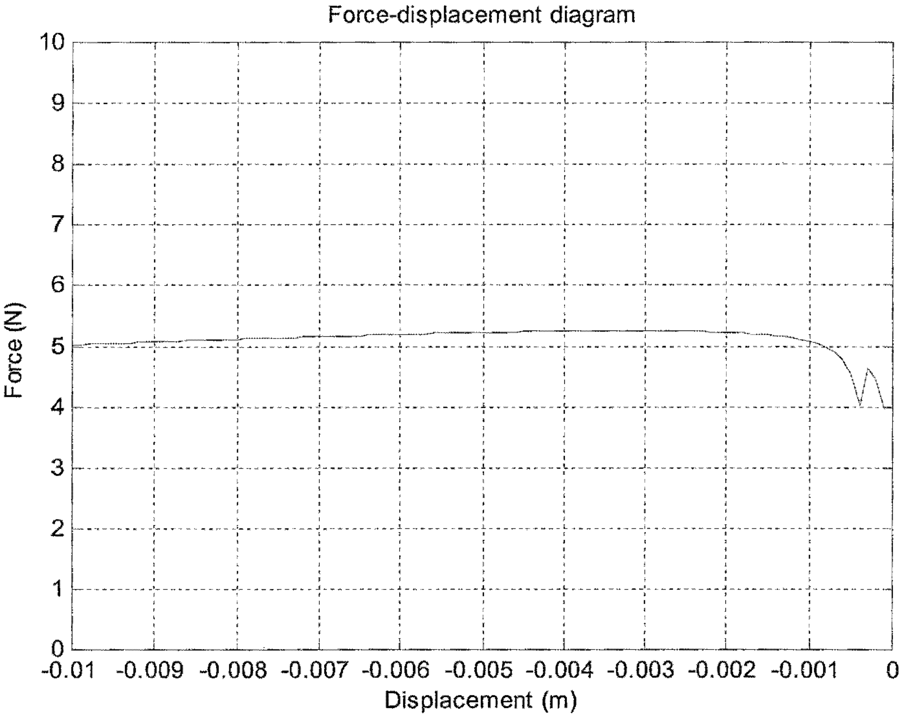


figure 3.8 *Force displacement diagram*

As can be seen in *figure 3.8* the force of the leaf spring will decrease when it buckles further. The leaf spring will collapse after a displacement of about 3.5 mm due to negative stiffness.

4. Dynamic numerical analysis

Until now, static analyse is done: only the forces of the leaf springs at several displacements are determined. The dimension time has not been mentioned yet. The only thing that is said about the movement of the system in time is the fact that it will oscillate when the upper mass is lifted up, until the leaf springs stretch fully, and then released. In this chapter this oscillation will be analysed. The acceleration of a mass is determined by the resultant of forces on it:

$$F = m \cdot a \tag{4.1}$$

in which

$F = force(N)$

$m = mass(kg)$

$a = acceleration(m/s^2)$

4.1 Friction excluded

When friction is neglected for the time being, the only forces on the upper mass are the gravity force F_z and the total force of the three leaf-springs together F_v (see *figure 4.1*).

The acceleration is the second derivative of the displacement to time and the force of the leaf spring is dependent on the displacement of the upper mass. The mass itself will be constant during the movement and so the gravity force of it will be. Equation (4.1) then can be written as follows:

$$m\ddot{y}(t) = -F_z + F_v(y(t)) \tag{4.2}$$

in which

m	=	*upper mass(kg)*
y(t)	=	*displacement upper mass(m)*
t	=	*time(s)*
Fz	=	*gravity force(N)*
Fv	=	*total force of leaf springs(N)*

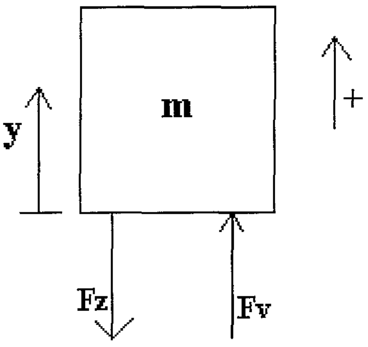


figure 4.1 Direction agreement

It is intended to analyse the movement of the system as good as possible. That is why the force-displacement diagram obtained by the static experiment is used in this analysis. The relation between F_v and y is not written in a formula (the force displacement diagram has been determined from a finite number of experiment values and therefore is not continuous, but discrete). A mathematical relation could be obtained by a fit, but in this situation the reality will be reconstructed in analysis and so the discrete values must be used. The next problem arises here: When a formula for F_v should exist, differential equation (4.2) could be solved in the program MATLAB with the command *ode45* (Numerical integration). Now the discrete force displacement diagram is used and after every time step the force of the leaf springs should be determined. This is done as follows. The second order differential equation (4.2) can be written in two first order differential equations:

$$\dot{y}(t) = v(t) \tag{4.3}$$

$$\dot{v}(t) = a(t) = \frac{1}{m}(-mg + F_v(y(t))) \tag{4.4}$$

The left parts of these equations are derivatives to time of displacement and velocity respectively. We can translate these continuous equations into discrete equations by using forward differentiation:

$$\frac{y_{k+1} - y_k}{dt} \approx \dot{y} \tag{4.5}$$

$$\frac{v_{k+1} - v_k}{dt} \approx \dot{v} \tag{4.6}$$

in which dt is a very small time step. Then the equations become:

$$y_{k+1} = y_k + v_k dt \tag{4.7}$$

$$v_{k+1} = v_k + a_k dt = v_k + (\frac{1}{m}(-mg + F_v(y_k)))dt \tag{4.8}$$

These algorithms are used in MATLAB to determine the displacement, velocity and acceleration in time: By knowing the displacement and velocity at one time, the displacement and velocity one *time step* further can be calculated by these iterations.

When the new displacement is calculated the force of the leaf springs can be determined from the experiment-data. When a displacement is found between two data points of the experiment a linear interpolation is used to calculate the corresponding spring force. The M-file is given in *Appendix 4, A.4.2.* The upper mass will make an oscillation. This will happen around some equilibrium point, the point at which gravity force and spring force will be equal. At all points above this point, the gravity force exceeds the spring force and acceleration will be negative. The mass will go down, beyond the equilibrium point eventually. At all points below this equilibrium point, the spring force will exceed the gravity force and so the acceleration will be positive. The mass will slow down, will get velocity equal to zero and will than go upwards beyond the equilibrium point and so on. The weight of the mass which has to be given in the MATLAB program therefore has to be chosen somewhere between the upper point and the buckling force of the Force-displacement diagram to get an oscillation.

Hypotheses

Before looking at the results of the MATLAB program hypotheses can be done: Therefore we need *figure 2.5* from chapter two. From this figure can be concluded that the absolute value of the stiffness of the leaf springs $\left\|\frac{dF}{dy}\right\|$ decreases when the leaf springs buckle further. When a linear spring-mass system, with mass m and spring stiffness k, is taken into consideration its eigen frequency will lie at $\omega_e = \sqrt{\frac{k}{m}}$. So when the stiffness decreases at equal mass, the eigen frequency also decreases. The frequency at points below the equilibrium point will therefore be smaller than the frequency at points above it. In one oscillation the upper mass then will be below the equilibrium point during a greater period than being above it.

Results program

In *figures 4.2, 4.3* and *4.4* respectively the results of the MATLAB-program are shown for the following settings:

```
y0=0,v0=0,dt=0.00005,ts=2,m=1.32
```

These are the results for the vertical movement of the upper mass.

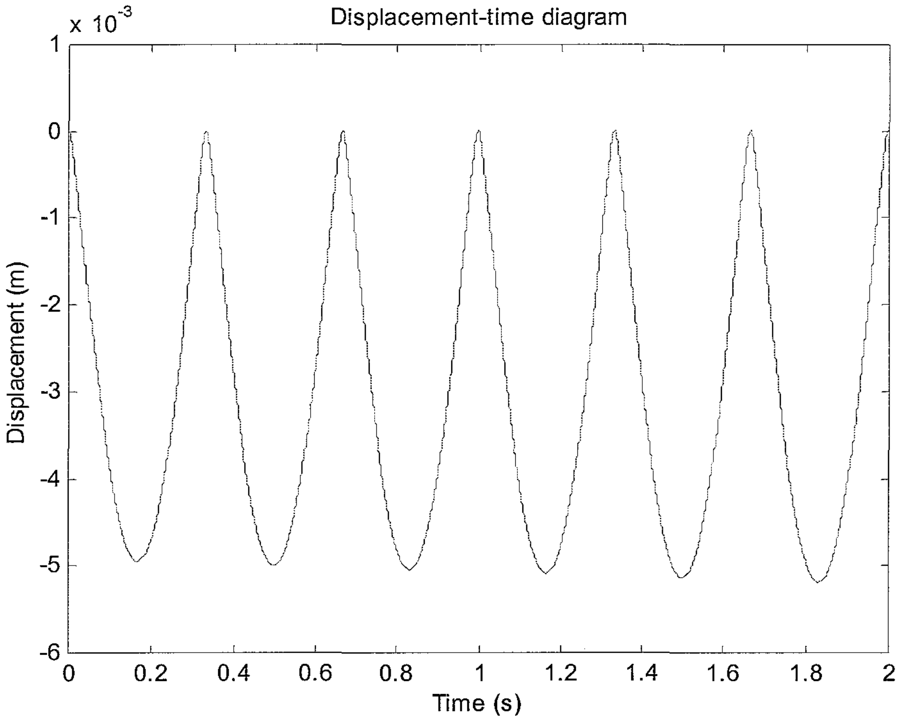


figure 4.2 Displacement-time diagram

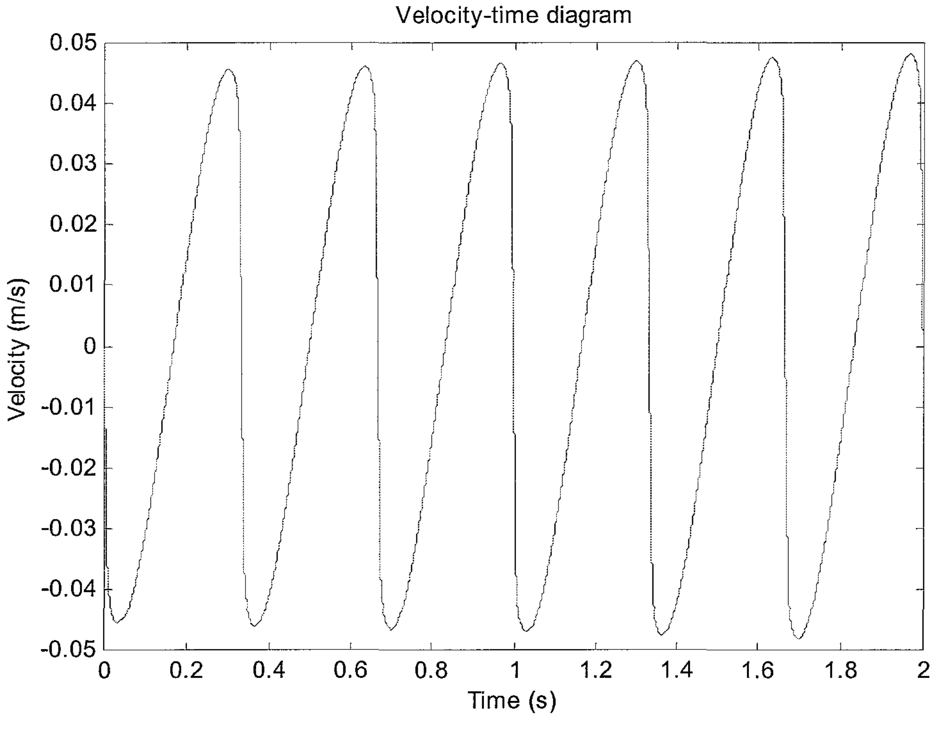


figure 4.3 Velocity-time diagram

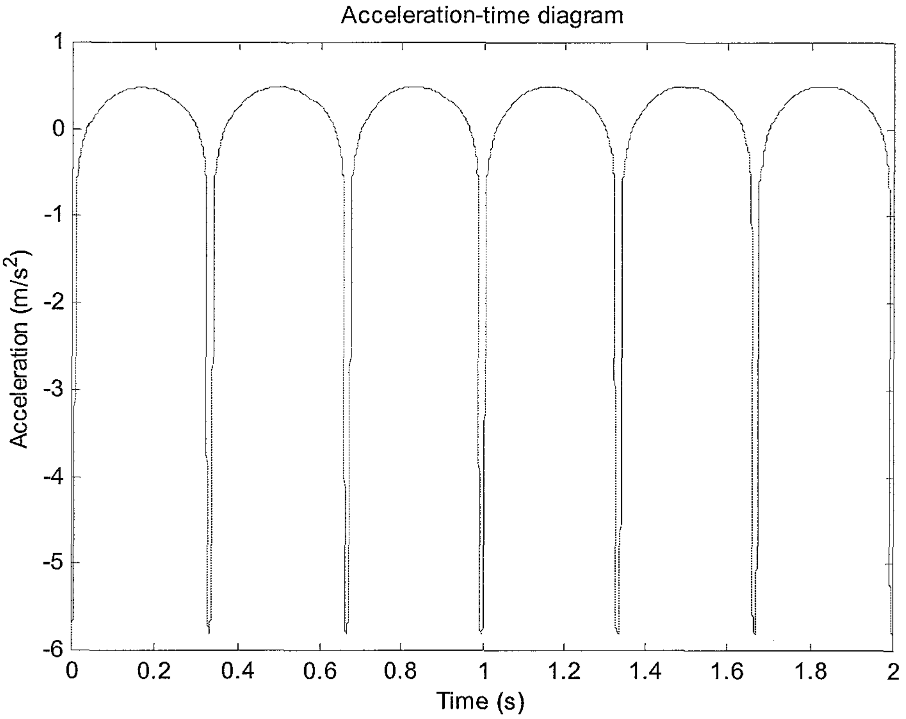


figure 4.4 Acceleration-time diagram

As predicted in the hypotheses, the upper mass will be underneath the equilibrium point during a greater time than it will be above it. The acceleration-time diagram shows sharp negative peaks with great amplitude and blunt positive peaks with relative small amplitude.

4.2 Friction included

The situation sketched above will never occur in reality: Friction is always present. In this case Coulomb and viscous friction are introduced and equations (4.3) and (4.4) will change into:

$$\dot{y}(t) = v(t) \tag{4.9}$$

$$\dot{v}(t) = a(t) = \frac{1}{m}(-mg + F_v(y(t)) - c_1 \frac{2}{\pi}\arctan(50v(t)) - c_2 v(t)) \tag{4.10}$$

and equations (4.7) and (4.8) into:

$$y_{k+1} = y_k + v_k dt \tag{4.11}$$

$$v_{k+1} = v_k + a_k dt =$$
$$v_k + \frac{1}{m}(-mg + F_v(y_k) - c_1 \frac{2}{\pi}\arctan(50v_k) - c_2 v_k)dt \tag{4.12}$$

In these equations the constants c_1 and c_2 represent the maximum value of the coulomb friction and the proportionality of the viscous friction respectively (see *figures 4.5 a and b*).

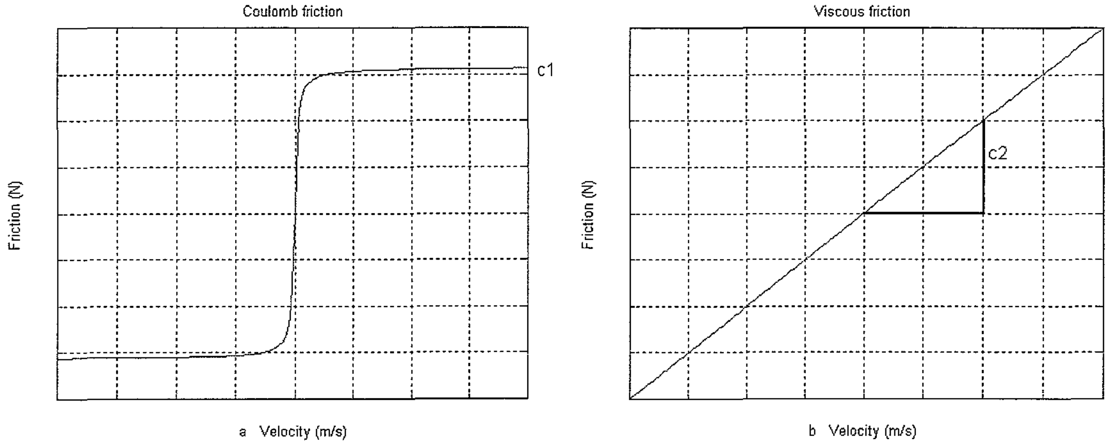


figure 4.5 (a) Coulomb and (b) viscous friction

The MATLAB-program is included in *Appendix 4, A.4.2*. The values of these constants can be determined by some methods of parameter estimation. However, because of lack of time the parameters are determined in the following chapter by the method of trial and error.

5. Dynamic experiment

5.1 Results

In this chapter the results of a dynamic experiment are compared with the results of a numerical model. With an accelerometer the acceleration is measured in time. The accelerometer is placed upon the upper mass, which has value m=1.32 kg) and the upper mass is lifted up until the leaf springs are fully stretched and from this position the upper mass is released (with velocity equal to zero). With help of the data-acquisition-tool SIGLAB the acceleration plot can be drawn. The following result is found (see *figure 5.1*).

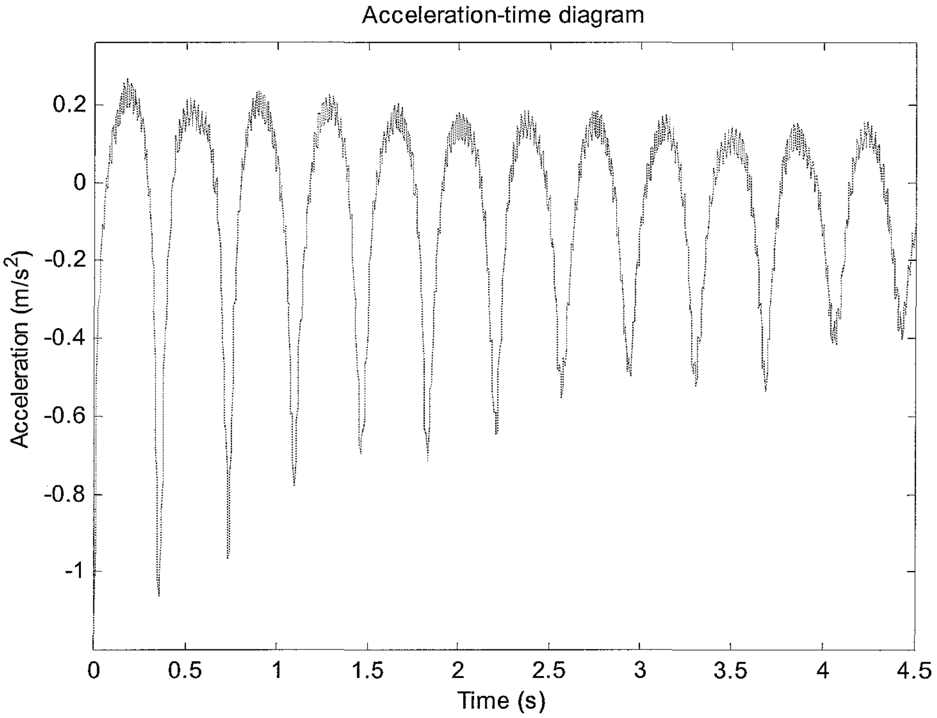


figure 5.1 Acceleration-time diagram

In this figure too, the sharp negative peaks and the blunt positive peaks can be recognised. The oscillation will damp due to friction. Now the parameters c_1 and c_2 can be determined by changing them in the MATLAB-program and comparing the result with the one found in the experiment until satisfying correspondence can be seen. For the values $c_1 = 0.01$ and $c_2 = 0.4$ the best result is obtained. In *figure 5.2* the experiment-result and the program result with foregoing values are plotted together. The values of time are left out here, because the figures have been put together for different times to avoid the problem of errors due to difference in initial conditions.

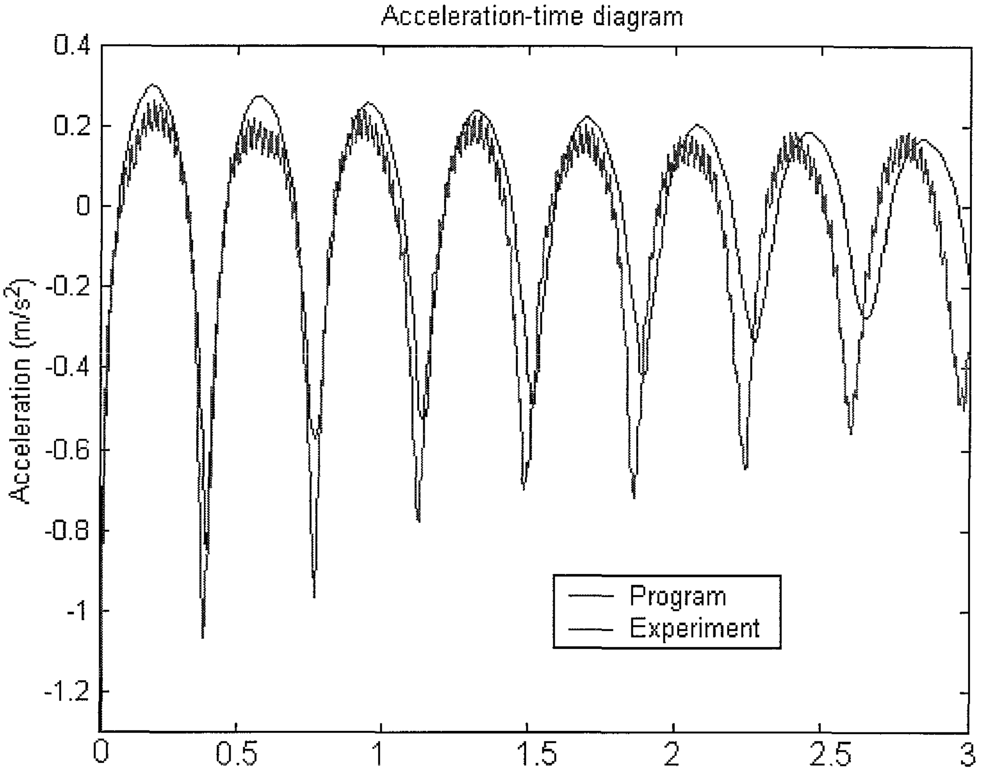


figure 5.2 Comparison of program and experiment

In the program the following settings are used:
`y0=0,v0=0.`

5.2 Causes of differences

As can be seen in the figure above differences in magnitude and frequency arise. Following reasons might explain these: First of all the leaf springs are very sensitive, they are easily damaged (plastic deformation): The static experiment has been done several days before the dynamic experiment. During the days between these two experiments, the leaf springs were damaged due to unknown causes. In favour of the dynamic experiment other springs were used. Therefore the stiffness of the leaf springs in the dynamic experiment will not be exactly the same as in the static experiment. Besides this, the dynamic program in MATLAB uses the numerical values of the data obtained in the statical experiment to interpolate linearly. The values used in the program therefore will be an approximation of the real values. Furthermore, the MATLAB program is one-dimensional, whereas the system is three-dimensional. The leaf springs are not fixed perfectly to the masses, which causes the upper mass to hang in a crooked position. Due to this the guiding pen will touch the upper mass and damp the oscillation very quickly. By omitting the leader pen the oscillation will not damp that fast, however, the mass will not make a pure translation anymore and side movements appear. They influence the oscillation and the acceleration measure for pure translation. These side-effects are not taken in consideration in the program.

Also the friction model can be seen as an approximation, which will not exactly show the reality. To overcome these problems some recommendations are made in the next chapter.

6. Recommendations

In the foregoing chapter, differences between experiment and program could be seen. These differences might be solved in a subsequent study, where the system is improved. Furthermore some other problems could be solved.
With respect to this the following recommendations are made for subsequent study:

1. A system can be developed in which the leaf springs are fixed better and because of that the upper mass will be guided better. The upper mass than makes a pure translation and experiment and theory can be compared better (if the leaf springs keep their stiffness).
2. With some parameter estimation, for example a Kalmann-filter, the Coulomb and viscous friction can be determined.
3. A model of the whole system instead of one leaf spring can be made in the program Marc/Mentat.
4. A model of the whole system (with side movements) can be made in MATLAB and results can be compared with those of the real one and those of the one in MARC.

Conclusions

A non-linear mass-spring system is analysed in this report: With help of static analyse, eventually a dynamical model is made of it. The results of both static and dynamic investigations are found in several ways: Theoretically, experimentally, by way of a simulation in Marc and by way of a mathematical program in MATLAB. Many similarities between experiment and numerical results have been found, however some differences appeared. In this report reasons have been given to explain the sources of these differences. The most important one seems to be the difference between perfect modelling and imperfect reality. Eventually satisfying results have been found and recommendations for a follow on study are given.

Afterword

After four years of courses, experiments and projects I could start my first stage at the Technical University in Eindhoven. At first I didn't know what to expect about it and how to use my experiences but later on, when the subject was explained to me, I saw a lot of problems had to be solved, which I should be capable of. There were some times I didn't know how to go further, but in then I could always fall back upon the experience and knowledge of Mr Kodde, Mr Fey and Mr Schreurs. They helped me to overcome al my problems and have contributed towards a good end of this assignment.
The fun about this assignment is the fact that several aspects of the mechanical engineering had to be used: Theoretical calculations, adjustment of the system, experiments, simulations in Marc and programming in Matlab. Because of this versatility it never got boring and I always enjoyed the work I was doing!

Thanks to Rens Kodde, Rob Fey and Piet Schreurs!

List of symbols

Symbol	Meaning	Dimension
α	Frequency	1/s
ρ	Density	kg/m^3
ω_e	Eigenfrequency	1/s
ν	Poisson's ratio	-
a	Acceleration	m/s^2
A	Constant	-
A_l	Matrix	-
A_m	Matrix	-
b	Width beam	m
$\vec{b}$	Vector	-
B	Constant	-
c	Constant	-
c_1	Coulomb friction constant	N/m/s
c_2	Viscous friction constant	N/m/s
d	Constant	-
dt	Time step	s
e	Constant	-
E	Elastic Modulus	N/m^2
f	Constant	-
F	Resultant force	N
F_z	Gravity force	N
F_v	Leaf spring force	N
g	Gravity force acceleration	m/s^2
h	Depth beam	m
I	Unity vector	-
I_s	Second moment of area	m^4
k	Spring stiffness	N/m
L	Length beam	m
m	Mass	kg
P	Compressive force	N
P_b	Buckling force	N
t	Time	s
ts	Program time	
$\vec{u}$	Vector	-
U	Potential energy	J
v	Velocity	m/s
w	Position beampoint	m
x	Position	m
x_i	Value voltage at measurement i	V
$\vec{x}$	Vector	-
y	Displacement	m
$\dot{y}$	Velocity	m/s
$\ddot{y}$	Acceleration	m/s^2

y_i	Value displacement at measurement i	m
$\vec{y}$	Vector	-
$\vec{z}$	Vector	-

List of literature

George J.Simitses, *An introduction to the Elastic Stability of Structures,* 1976
Roger T.Fenner, *Mechanics of Solids,* 2000
P.Schreurs and H.Giesen, *Handleiding Mentat en Marc,* 1999
W.J. Groot en C.M. Menken, *Stabiliteit en Knik,* 1985

Appendix 1 Partial integration

$$\partial U = \int_0^l EIw_{,xx}\partial w_{,xx}dx - \int_0^l Pw_{,x}\partial w_{,x}dx \qquad \text{(A.1.1)}$$

Using partial integration for the left part of the right side of equation A.1.1

$$\int_0^l EIw_{,xx}\partial w_{,xx}dx = {}_0^l\left[EIw_{,xx}\partial w_{,x}\right] - \int_0^l EIw_{,xx,x}\partial w_{,x}dx \qquad \text{(A.1.2)}$$

Using partial integration again for the right part of the right side of equation A.1.2:

$$\int_0^l EIw_{,xx,x}\partial w_{,x}dx = {}_0^l\left[EIw_{,xx,x}\partial w\right] - \int_0^l EIw_{,xx,xx}\partial w dx \qquad \text{(A.1.3)}$$

Using partial integration for the right part of the right side of equation A.1.1:

$$\int_0^l Pw_{,x}\partial w_{,x}dx = {}_0^l\left[Pw_{,x}\partial w\right] - \int_0^l Pw_{,x,x}\partial w dx \qquad \text{(A.1.4)}$$

Now equation A.1.1 turns into

$$\partial U = \int_0^l \left[EIw_{,xx,xx} + Pw_{,x,x}\right]\partial w dx + {}_0^l\left[EIw_{,xx,x}\partial w_{,x}\right] - {}_0^l\left[EIw_{,xx,x}\partial w\right] - {}_0^l\left[Pw_{,x}\partial w\right] \qquad \text{(A.1.5)}$$

From $\partial U = 0$ follows:

$$EIw_{,xx,xx} + Pw_{,x,x} = 0 \qquad \text{(A.1.6)}$$

Appendix 2 Solving differential equation

$EIw_{,xxxx} + P_b w_{,xx} = 0$

Homogene solution to this equation:

$w(x) = ce^{\lambda x}$

This leads to:

$(\lambda^4 EI + \lambda^2 P_b)ce^{\lambda x} = 0$

$\lambda^2(\lambda^2 EI + P_b) = 0$

With solutions:

$\lambda_{1,2} = 0$

$\lambda_{3,4} = \pm i\sqrt{\frac{P_b}{EI}} = \pm i\alpha$

So the solution gets:

$w(x) = c + dx + e\cos(\alpha x) + f\sin(\alpha x)$

Appendix 3 Least-square fit

$$\frac{1}{2}\sum_{i=1}^{n}(y_i-(Ax_i+B))^2 = \frac{1}{2}(\vec{y}-(A\vec{x}+B\vec{I}))^T(\vec{y}-(A\vec{x}+B\vec{I}))$$

$$\frac{d}{dA}(\frac{1}{2}(\vec{y}-(A\vec{x}+B\vec{I}))^T(\vec{y}-(A\vec{x}+B\vec{I}))) =$$

$$-(\vec{y}-(A\vec{x}+B\vec{I}))^T\vec{x} = -\vec{y}^T\vec{x}+A\vec{x}^T\vec{x}+B\vec{I}^T\vec{x}=0$$

$$\frac{d}{dB}(\frac{1}{2}(\vec{y}-(A\vec{x}+B\vec{I}))^T(\vec{y}-(A\vec{x}+B\vec{I}))) =$$

$$-(\vec{y}-(A\vec{x}+B\vec{I}))^T\vec{I} = -\vec{y}^T\vec{I}+A\vec{x}^T\vec{I}+B\vec{I}^T\vec{I}=0$$

This can be written in the form $A_m\vec{z}=\vec{b}$
in which:

$$A_m=\begin{bmatrix}\vec{x}^T\vec{x} & \vec{I}^T\vec{x}\\ \vec{x}^T\vec{I} & \vec{I}^T\vec{I}\end{bmatrix},\qquad \vec{z}=\begin{bmatrix}A\\ B\end{bmatrix},\qquad \vec{b}=\begin{bmatrix}\vec{y}^T\vec{x}\\ \vec{y}^T\vec{I}\end{bmatrix}$$

Appendix 4 Matlab programs

A.4.1 Force displacement diagram

```
%beweging3.m
%M-file to determine first-order fit for calibration of the
displacement reader
%After that a force-displacement diagram is made with help of this
calibration

close all

Volt=[   -5.53
   -5.07
   -4.63
   -4.29
   -3.73
   -3.28
   -2.79
   -2.38
   -1.9
   -1.44
   -0.86
   -0.42
   -0.21
   0.04
   0.51
   1.44];

Verp=[   40
   36.7
   33.3
   30.1
   26.6
   22.9
   19.8
   16.2
   13.2
   9.6
   4.5
   0.9
   0
   -2.1
   -4.8
   -12.3];

Verp=-(10^-3)*Verp;

plot(Volt,Verp)
hold on

I=ones(length(Volt),1);

Voltkw=Volt'*Volt;
IVolt=I'*Volt;
VoltI=Volt'*I;
Ikw=I'*I;

VerpVolt=Verp'*Volt;
VerpI=Verp'*I;
```

```
A=[Voltkw IVolt
   VoltI Ikw];

b=[VerpVolt VerpI]' ;

x=A\b          %Vector x contains values of the first order fit

t=Volt(1,1):0.01:Volt(length(Volt),1);
h=x(1,1)*t+x(2,1);

%plot(t,h,'r')
%Determining force-displacement diagram

close all
Volt=[
-0.21
-0.21
-0.21
-0.22
-0.22
-0.22
-0.22
-0.22
-0.22
-0.22
-0.22
-0.22
-0.22
-0.23
-0.24
-0.25
-0.26
-0.26
-0.27
-0.27
-0.28
-0.31
-0.32
-0.33
-0.34
-0.35
-0.36
-0.37
-0.38
-0.4
-0.51
-0.54
-0.58
-0.59
-0.62
-0.68
-0.7
-0.74
-0.79
-0.8
-0.86
-1.14
-1.22
-1.3
-1.33
```

```
-1.34
-1.35
-1.43
-1.5
-1.53
-1.61
-1.69
-1.95
-2.05
-2.13
-2.17
];

Kracht=[
5.26
5.76
6.26
6.76
7.26
7.76
8.26
8.76
9.26
9.76
10.26
10.76
11.26
11.76
12.2
12.26
12.31
12.37
12.42
12.48
12.58
12.7
12.75
12.8
12.81
12.86
12.91
12.92
12.97
13.02
13.2
13.25
13.3
13.31
13.36
13.41
13.42
13.47
13.52
13.53
13.58
13.7
13.75
13.8
13.81
13.86
13.87
```

```
13.92
13.97
13.98
14.03
14.08
14.2
14.25
14.3
14.31
];

Verpl=x(1,1)*Volt+x(2,1);        %Displacement is determined with
help of calibration
Verpl=Verpl(2:length(Verpl),1);

Verpl=[0;Verpl]
Kracht

plot(Verpl,Kracht)
axis([-0.015 0 0 15])

length(Kracht)
```

A.4.2 Dynamical program

```
%beweging2.m
%M-file of the movement of a non-linear system
%Force-displacement diagram is determined in M-file beweging3.m
%Differential equations are integrated numerically here
%Points from force-displacemnt diagram are linearly interpolated

function beweging2(y0,v0,A,B,dt,ts,c,d,m)
%x0=initial position
%v0=initial velocity
%A=vector containing values of mass position
%B=vector containing springforce going with the values from vector A
%dt=timestep
%ts=simulation time
%c=maximal vale of Coulomb friction
%d=proportionality of viscous friction
st=ts/dt; %number of steps
y=zeros(ts,1);
v=zeros(ts,1);
a=zeros(ts,1);

y(1,1)=y0;
v(1,1)=v0;
for k=1:1:st
   if y(k,1) >= A(1,1)
      F(k,1) =  B(1,1);
   elseif y(k,1) <= A(length(A),1)
      F(k,1) = B(length(A),1);
   else p=1;
      while y(k,1) < A(p,1),
         p=p+1;
      end;
      F(k,1)=B(p-1,1)+((B(p,1)-B(p-1,1))/(A(p,1)-A(p-1,1)))*(y(k,1)-
A(p-1,1));
   end
   a(k,1)=(-9.81+(1/m)*(F(k,1)-c*(2/pi)*atan(50*v(k,1))-d*v(k,1)));
   v(k+1,1)=v(k,1)+a(k,1)*dt;
   y(k+1,1)=y(k,1)+v(k,1)*dt;
end

T=[];
for t=0:dt:ts
   T=[T;t];
end

length(T)
length(a)
length(v)
length(y)

figure(1)
plot(T,y)

figure(2)
plot(T,v)

T=T(1:(length(T)-1),1);

figure(3)
plot(T,a)
```

A.4.3 Programs from figures

```
%figures 1.3 a and b
%knikbelasting.m
%M-file to determine the buckling force

b=20e-3;
h=0.2e-3;
E=200e9;
I=(b*h^3)/12;
L=13.9e-2;
H=[];
P=[];
T=[];
M=[];

I=(b*h^3)/12;
   Pe=(4*pi^2*E*I)/L^2           %buckling force
   t=3*Pe;                       %buckling force for three leaf
springs
   m=t/9.81;

for h=0.1e-3:0.01e-3:0.5e-3
   I=(b*h^3)/12;
   Pe=(4*pi^2*E*I)/L^2;          %buckling force
   t=3*Pe;                       %buckling force for three leaf
springs
   m=t/9.81;
   H=[H;h];
   P=[P;Pe];
   T=[T;t];
   M=[M;m];
end
H;
M;

figure(1)
plot(H,P)
title('Buckling force at L=13.9cm')
xlabel('Thickness leaf spring, [m]')
ylabel('Buckling force [N]')
grid on

b=20e-3;
h=0.2e-3;
E=200e9;
I=(b*h^3)/12;
L=[];
P=[];
T=[];
M=[];
for l=7.5e-2:0.5e-2:20e-2
   I=(b*h^3)/12;
   Pe=(4*pi^2*E*I)/l^2;          %buckling force
   t=3*Pe;                       %buckling force for three leaf
springs
   m=t/9.81;
   L=[L;l];
   P=[P;Pe];
   T=[T;t];
   M=[M;m];
```

```
end

figure(2)
plot(L,P)
title('Buckling force at h=0.2mm')
xlabel('Length leaf spring, [m]')
ylabel('Buckling force [N]')
grid on

/////////////////////////////////////////////////////////////////

%figures 4.5 a and b
%wrijving.m
%Coulomb and viscous friction

t=-2:0.01:2;
y=-atan(50*t);
z=-t;

figure(1)
plot(t,y)
grid on

figure(2)
plot(t,z)
grid on
```