

Markus Tuominen

**Multibody simulations as a product
development tool: introduction to ADAMS
and two analyses**

School of Engineering

Thesis submitted for examination for the degree of Master of
Science in Technology.

Espoo 12.01.2015

Thesis supervisor:

Prof. Matti Pietola

Thesis advisor:

M.Sc. (Tech.) Vesa Nuotio

Author: Markus Tuominen

Title: Multibody simulations as a product development tool: introduction to ADAMS and two analyses

Date: 12.01.2015

Language: English

Number of pages: 7+79

Department of Mechanical Engineering

Professorship: Machine Design

Code: K3001

Supervisor: Prof. Matti Pietola

Advisor: M.Sc. (Tech.) Vesa Nuotio

Multibody system is a system consisting of multiple parts and components attached together with joints and connectors. Multibody simulation is a field of applied mechanics researching multibody systems and their dynamics. In this thesis multibody systems' dynamic theory, methods and history are presented along with virtual prototyping process and usage applications. Additionally, two different real life problems are analyzed and solved with MSC ADAMS software. The problems have been obtained from Thermo Fisher Scientific's research and development team. The goal of this problem solving is to demonstrate the capabilities of the software and how using it eases and speeds up product development. In the first problem, a gas spring is designed to lighten the opening of the main cover of an analyzer. In the second problem, a model is constructed to investigate the vibrations of a dispensing needle and the root causes behind the phenomenon inspected.

Keywords: Multibody System, Multibody Dynamics, MSC ADAMS, Simulation, Static, Dynamic, Flexible Bodies, Vibrations, Virtual Prototyping Process

Tekijä: Markus Tuominen		
Työn nimi: Monikappalesimulointi tuotekehityksen työkaluna: johdatus ADAMS-ohjelmistoon ja kaksi analyysia		
Päivämäärä: 12.01.2015	Kieli: Englanti	Sivumäärä: 7+79
Koneenrakennuksen laitos		
Professori: Koneensuunnittelu		Koodi: K3001
Valvoja: Prof. Matti Pietola		
Ohjaaja: DI Vesa Nuotio		
<p>Monikappalejärjestelmä on systeemi joka koostuu useasta osasta tai komponentista ja jotka on liitetty toisiinsa nivelin. Monikappalesimulointi on sovelletun mekaniikan ala joka tutkii monikappalejärjestelmiä ja niiden dynamiikkaa. Tässä työssä monikappalejärjestelmien dynamiikan teoriaa, menetelmiä ja historiaa esitellään kuten myös virtuaalinen suunnitteluprosessi sekä simuloinnin käyttö-tarkoituksia. Tämän lisäksi MSC ADAMS -ohjelmalla simuloimalla tutkitaan ja analysoidaan kaksi erilaista tosielämän ongelmaa jotka on poimittu Thermo Fisher Scientificin tuotekehityksestä. Ongelmanratkaisun tarkoitus on esitellä simulointiohjelmiston kykyjä ja että miten ohjelmiston käyttö helpottaa ja nopeuttaa tuotekehitystä. Ensimmäisessä ongelmassa suunnitellaan ja mitoitetaan kaasujousi analysaattorin pääkannen avaamisen keventämiseksi. Toisessa ongelmassa mallinnetaan analysaattorin annosteluneulan värähtelyitä sekä niiden syntyyn vaikuttavia syitä.</p>		
Avainsanat: Monikappalejärjestelmät, Monikappalesimulointi, MSC ADAMS, Simulointi, Staattinen, Dynaaminen, Värähtelyt, Virtuaalinen suunnittelu		

Preface

Haluan kiittää Professori Matti Pietolaa ja ohjaajaani Vesa Nuotiota niin hyvästä kuin huonosta ohjauksesta sekä Thermon työkavereita kannustavasta ja inspiroivasta ilmapiiristä sekä työn edistymisen valvomisesta.

Kiitokset myös perheelleni kaikesta tuesta ja TkT Petri Makkoselle jonka innoittamana tutustuin simuloinnin maailmaan.

Vantaa, 12.01.2015

Markus T. Tuominen

Contents

Abstract	ii
Abstract	iii
Preface	iv
Contents	v
Symbols and abbreviations	vi
1 Introduction	1
2 Background	3
2.1 Brief History	4
2.2 Usage and Applications	5
2.3 Virtual Design Process	7
3 Methods and Tools	10
3.1 Topologies	11
3.2 Basic Kinematics	12
3.2.1 Frame of Reference and Coordinates	12
3.2.2 Positions and Translations	12
3.2.3 Rotations	13
3.2.4 Velocities	15
3.2.5 Accelerations	16
3.3 Dynamics	16
3.3.1 Newtonian Method	17
3.3.2 Equations of Motion	18
3.3.3 Virtual Work	18
3.3.4 Lagrangian and Hamiltonian Dynamics	19
3.4 Flexible Bodies	20
3.4.1 Kinematics	20
3.4.2 Strain	20
3.4.3 Stress	21
3.4.4 Discrete Flexible Link	21
3.4.5 Area Moments of Inertia	23
3.5 Degrees of Freedom and Joints	23
3.6 Vibrations and Damping	26
3.6.1 Mass-spring Absorber	26
3.6.2 Modal Analysis	27
3.6.3 Modal Neutral Files	28
3.6.4 Natural Frequencies of Beams	29
3.7 MSC ADAMS	29
3.7.1 Simulation Types	29

3.7.2	Solvers	30
3.7.3	Optimization	31
4	Analysis	35
4.1	Problem 1: Lifting aid for main cover	35
4.1.1	Background	35
4.1.2	Model	36
4.1.3	Analysis	41
4.1.4	Results Verification	49
4.1.5	Conclusions	49
4.2	Problem 2: Vibration analysis of a needle	52
4.2.1	Background	52
4.2.2	Stepper Motor Flexibility	53
4.2.3	Constructing the Model	55
4.2.4	Analysis	64
4.2.5	Results Verification	65
4.2.6	Conclusions	70
5	Conclusions	72
	References	74
A	First appendix: HAHN G 10-28	77

Symbols and abbreviations

Symbols

x_i, r_i	coordinates
$\theta, \alpha, \beta, \gamma$	angles of rotation
$\theta_0, \theta_1, \theta_2, \theta_3$	Euler parameters
$\gamma_1, \gamma_2, \gamma_3$	Rodriguez parameters
m	mass of a body, kg
F	force, N
I	mass moment of inertia, kgm^2 or area moment of inertia, mm^4
T	torque, Nm
δW	virtual work
ε	strain
σ	stress
k	spring constant, N/mm
ζ	damping ratio
c	damping factor, Ns/m
\mathbf{i}_i	unit vector
\mathbf{r}	3 x 1 position vector
$\mathbf{v} = \dot{\mathbf{r}}$	3 x 1 linear velocity vector
$\mathbf{a} = \dot{\mathbf{v}}$	3 x 1 acceleration vector
\mathbf{q}_r^i	3 x 1 location vector of body i in 2D
\mathbf{q}	n x 1 vector of generalized coordinates
\mathbf{Q}	n x 1 vector of generalized forces
$\delta \mathbf{r}^i$	n x 1 virtual displacement vector
\mathbf{S}_i	3 x 1 rotation vector
$\boldsymbol{\omega}$	3 x 1 angular velocity vector
$\boldsymbol{\alpha}$	3 x 1 angular acceleration vector
$\boldsymbol{\varepsilon}$	6 x 1 strain vector
$\boldsymbol{\sigma}$	6 x 1 stress vector
$\tilde{\mathbf{v}}$	3 x 3 skew symmetric velocity matrix
\mathbf{I}	n x n identity matrix
\mathbf{A}	3 x 3 transformation matrix
\mathbf{A}_4	4 x 4 transformation matrix
$\mathbf{E}, \bar{\mathbf{E}}$	3 x 4 Euler parameter matrices
$\tilde{\boldsymbol{\omega}}$	3 x 3 skew symmetric angular velocity matrix
$\boldsymbol{\varepsilon}_m$	3 x 3 strain tensor
$\boldsymbol{\sigma}_m$	3 x 3 stress tensor
$\bar{\mathbf{J}}$	3 x 3 gradient of displacement vector
\mathbf{E}	6 x 6 matrix of elastic constants

Operators

$\frac{d}{dt}$ time derivative

$\frac{\partial}{\partial t}$ partial derivative

Abbreviations

ADAMS	Automatic Dynamic Analysis of Mechanical Systems
CAD	Computer Aided Design
CAE	Computer Aided Engineering
FEA	Finite Element Analysis
MBD	Multibody Dynamics or Microbiology Division
MBS	Multibody System
MSC	MacNeal-Schwendler Corporation
NASTRAN	NASA Structural Analysis
NVH	Noise, Vibration and Harshness
TFS	Thermo Fisher Scientific

1 Introduction

In product development there is always a need to get things done quickly and efficiently to keep costs down and projects on time. Businesses are often looking for tools to help to achieve these goals. *Simulations* are, for example, such accessories. With simulations, real life phenomena, processes and systems can be replicated, fine tuned and predicted on computers.

One popular utilization of simulations is in the field of traditional Newtonian physics. *Multibody systems* are system consisting of multiple parts and components attached together with various kinds of joints. These systems have a certain dynamics and behaviour depending on the components and their geometry as well as external forces and motions applied to them. The area researching multibody systems is called *Multibody Dynamics*. There are several simulation programs designed to solve these MBD problems. The software chosen for this thesis is MSC Software Corporation's ADAMS due to its familiarity and features. Additionally, MSC ADAMS has not been used before at Thermo Fisher Scientific so it also represents an interesting new possibility.

Thermo Fisher Scientific, whose logo is shown in Fig 1, is a manufacturer and distributor of laboratory supplies and devices employing 50 thousands personnel world wide. Their mission is described as [1]:

...to enable our customers to make the world healthier, cleaner and safer. We help our customers accelerate life sciences research, solve complex analytical challenges, improve patient diagnostics and increase laboratory productivity.

In Finland, TFS has an office in Vantaankoski, Vantaa and a factory in Joensuu. The Vantaa site has multiple research and development teams designing and building new products and researching new possibilities. As the instruments are becoming increasingly advanced and the schedules increasingly tight, an interest towards simulation programs has risen. No programs of this type have been utilized before at TFS Vantaa. One purpose of this thesis is to present the employees of TFS Vantaa the basics of multibody dynamics, introduce the Virtual Design Process and show two practical examples and how they can be solved with MBD software.



Figure 1: Thermo Fisher Scientific logo [1]

This thesis has two main intentions: first, the reader is acquainted with the fundamentals of multibody dynamics, simulations and methods along with the Virtual Design Process. The aim is to familiarize the reader with the subject and to give an idea of what multibody simulations are capable of. The second goal is to show the reader how actual problems are solved and what benefits there are in using MBS

software. This is done with two practical examples selected from everyday problems faced at TFS Vantaa: designing a lifting aid for a main cover and investigating vibrations of a dispensing needle.

The instruments always have different hatches and covers that are opened by the user. The covers come in all shapes and sizes but they all must be light to operate. With large covers, there is often a need for a gas spring to lighten the load. Accordingly, the purpose of the first problem is to position and choose a gas spring for this function. The needle vibrations in the second problem has been a nuisance for some time but the exact cause has remained a mystery. When the dispensing robot moves in the instrument its needle vibrates correspondingly. The situation is therefore simulated and analysed and attempted to resolve the root cause of the vibrations.

In the first parts of this thesis background information is presented including a brief history of multibody dynamics along with its typical usage and some applications. The *Virtual Design Process* is also described. In Section 3 Methods and Tools are introduced. Some basic mathematics and equations behind the physics and dynamics are listed. *MSC ADAMS* is also announced and the fundamentals behind it, such as integrators, briefly explained. In the Analysis Section the two problems are described, analysed and solved with aforementioned tools. Finally, Section 5 concludes the thesis.

2 Background

Multibody system is a system where one or more bodies are connected to each other with joints. The bodies may be rigid, flexible or plain links that connect two pieces together. The joints' purpose is to limit the bodies' relative movement. There are many different kinds of joints for different purposes, most common are perhaps translational, rotational and spherical joints. There may also be springs, dampers and external forces that apply to the system. For example, car suspension, door lock and industrial robots are typical multibody systems as shown in Fig 2. [34]

Multibody system dynamics is a field of applied mechanics that studies these systems. Often the objective is to determine how a mechanism behaves or moves under the influence of external forces. For example, when you pull a door lever how does the door act. This type of analysis is called *forward dynamics*. It can also be analysed the reverse problem: what kind of forces must be applied to make the mechanism move in a specific manner. This is called *inverse dynamics*. [23]

Multibody simulation is tool with which multibody systems can be simulated using computational methods. Most often a computer is utilized. Simulating systems and mechanisms is important in modern engineering. It allows the designer to observe how the mechanism functions without having to build an expensive real world model and doing labour intensive measurements. For example, car industry uses simulations software extensively during the development of new cars. Every design feature is simulated to confirm it performs as intended. Vehicle dynamics are thoroughly analysed before building a prototype and testing it on a track. The focus of using MBS during research and development is often to reduce costs, shorten the product design cycle and optimize models. [5]

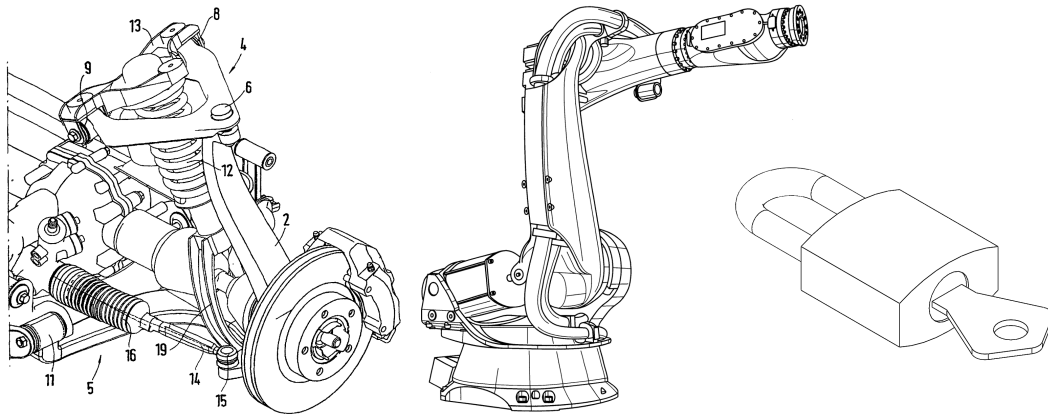


Figure 2: Examples of multibody systems: a car suspension [25], a robotic arm [33] and a padlock

MBS is also closely related to control design and vibration theory [11]. Control systems can be easily implemented in to the multibody model allowing designers to test the functionality of a whole process or system. The control systems can be further improved with MATLAB integration. [22]

Vibrations are also naturally solved by MBS software since the equations for vibrations can be acquired from d'Alembert's principle used for dynamic analysis of multibody systems [11].

Furthermore, modern MBS softwares can also perform dynamic strength analyses, contact and impact problems and interaction with fluids. Strength analysis is often utilized for life-cycle predictions including fatigue and crack growth. FEA tools are already available for even more accurate results and some are even integrated into MBD software. Fluid interaction is another concurrent topic. Typical problems are with satellites and tank vehicles. At the moment, all such models have strong simplifications of the fluid motion but as computer calculating power increases the models become more complex and accurate. [31]

2.1 Brief History

The history of multibody dynamics begin with Isaac Newton's *Philosophiae Naturalis Principia Mathematica* published in 1687 [24]. In this book Newton introduces the concept of a free particle. A particle is free when there are no external forces affecting it. Almost a century later, in 1776, Euler defined the rigid body – the most fundamental building block in multibody physics – in his article *Nova methodus motum corporum rigidorum determinandi* [12]. Combined with Newton's work, a new set of equations were created: the *Newton-Euler equations*. These formulas describe the movement of a free body whose constraints and joints are modelled as reaction forces and external forces apply. Euler also made significant progress in the field of gyroscope kinematics and dynamics. A picture of a gyroscope can be seen in Fig 3.

Great advances were also made in 18th century France where d'Alembert published his *Traité de Dynamique* in 1743 [10]. In his book d'Alembert examined a system of constrained rigid bodies. A little later Lagrange further worked on d'Alembert's idea and combined it with the principle of virtual work in his *Mécanique Analytique* in 1788 [17]. This resulted in a set of ordinary differential equations with which complex systems could be analysed faster. In the same publication Lagrange also presented the *Euler-Lagrange Differential equations*. With these equations systems could be analysed utilizing kinetic and potential energies.

The most recent advance in the field was made by Kane and Levinson in their book "Dynamics, Theory and Applications" (1985) [16]. In the book the generalized velocities are identified as partial velocities leading to a compact description of multibody systems. In relation to this, a new *Kane's method* was introduced for solving systems quickly and without the need of differentiating [28] [16].

After the invention of the computer the development of multibody dynamics has been geared towards algorithms, software and numerical formalisms. These include pre- and postprocessing, CAE coupling, animation and strength calculation. Lately, symbolical formula manipulation with equations of motion has become available as well. In the scientific field issues such as flexibility, friction, contact, impact and control are being researched. [31]

The origins of MSC.ADAMS are in the mid-1960s when computers begun to emerge. Programs were developed to calculate kinematic behaviour such as dis-

placement, velocity and acceleration as well as reaction forces for a limited set of linkages and simple suspension models. The first program to solve time histories of dynamic systems was called DAMN (Dynamic Analysis of Mechanical Networks) made at the University of Michigan in 1967. These programs were inherently 2D only. The basis of MSC.ADAMS is regarded to be a program completed in 1973 and published in a series of ASME papers. This was the first time 3D objects were made possible. Over time a number of other softwares were implemented and bought into ADAMS. The main emphasis has been vehicles and their dynamics lately. [5]

2.2 Usage and Applications

Earliest applications of multibody dynamics are found in biomechanics, namely, the walking motion of humans. Biomechanics was also strongly supported by athletic training and sports beginning from the 1950s. There have been several body motion studies for humans as well as animals. Another early application were gyroscopes and their dynamics, namely the field of gyro dynamics. In relation to gyroscopes, there was a substantial emphasis in satellites and other spacecraft during the space race from 1955 to 1972 which also accelerated the development of the whole field of dynamics. [31]

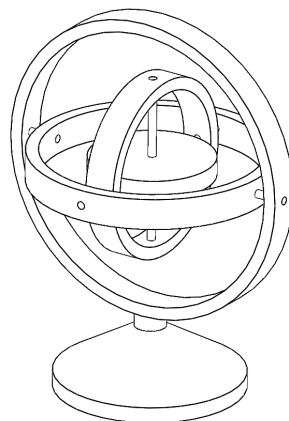


Figure 3: A gyroscope

According to MSC Software, their software, ADAMS, can be used for many different types of analyses and problems and it has many features, such as [23] [22]:

- Rigid and flexible multibody systems
- Sensitivity analysis
- Vibration analysis
- Vehicle design & testing
- Coupled control/mechanical system analysis

- Kinematics and kinetics
- Contact and friction
- Loads and displacement
- Durability and life-cycle analysis
- Fracture or fatigue calculations
- Kinetic, static, and dissipative energy distribution
- Vehicular cornering, steering, quasi-static, and straight-line analysis
- Control system analysis

Some industry uses include [23]:

- *Aerospace & Defence*: Aircraft engines, space vehicles, landing gears, helicopter fuselage, weapon systems, armaments, aircraft control mechanisms, ejection seats, flight simulators, battlefield vehicles, mission-critical spacecraft mechanisms
- *Automotive*: Suspension systems, Drivetrains, brake systems, steering systems, engines, control systems, transmissions, boot joints, bearings, clutches, chassis structure
- *Manufacturing*: robotic manipulators, conveyor belts, pumps, machine tools, packaging equipment, gears, stepper and server motors
- *Heavy Equipment*: excavators, agricultural equipment, hydraulic control systems, tracked vehicles, fork lifts, amusement park rides
- *Medical*: Orthopaedics, human locomotion, biodynamics, dynamometry, ergonomic analyses, robotic limbs
- *Consumer Products*: Sporting goods, bicycles, tools, printers
- *Energy*: Wind turbines, solar panels, offshore structures, drilling rigs, centralizer mechanisms

In modern engineering, one of the most dominant users of multibody simulations is the automotive industry. Vehicle dynamics is a major application field with many different problems – many of which can be solved with MBS software.

In Thermo Fisher Scientific Oy Vantaa and more specifically Microbiology Division, the employer and funder of this thesis, multibody simulations have not been used in R&D before. However, there has been a rising interest because the devices are becoming more and more complicated and the schedules increasingly tight. There are also some vibration, flexibility and durability issues that are difficult to analyse using conventional methods.

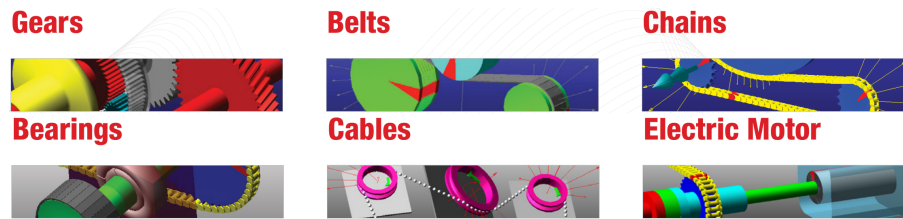


Figure 4: Features of MSC ADAMS/Machinery courtesy of MSC Software Corporation [22]

2.3 Virtual Design Process

Virtual design process is a method for virtually testing and validating a model or a product before physically building a prototype. Virtual prototyping is becoming increasingly popular in various industrial fields. Benefits and goals include cutting design time and costs, increasing quality and increasing efficiency. Virtually testing a product enables the designers to notice faults and weaknesses early and correct them as soon as possible as well as estimating different important factors to product performance and customer acceptance. Common tools are e.g. Computer Aided Design and Engineering (CAD and CAE) programs, multibody and multiphysics simulators and Finite Element Methods (FEM). Also lifetime, fatigue and Noise, Vibration and Harshness (NVH) may be determined virtually. [6] [32]

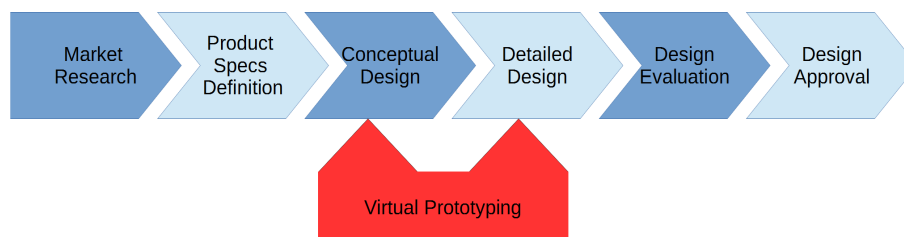


Figure 5: The product design process with virtual prototyping [6]

In Fig 5 is shown a typical product development process. The whole process start with a market research which collects the needs and wishes of the user. Based on this data, the product features and performance are defined in the next step. This specification and requirements guide designers towards the end product and clearly state what is needed from the design. The designers can also interpret the requirements and what can be done with current technologies. A *concept design* is created as a result. [6]

From the concept design the engineers begin modelling the product. During the conceptual design phase the main features and capabilities are decided and afterwards it will be increasingly difficult to alter these decision. Therefore it is vital to validate the performance of the design with virtual tools such as multibody and multiphysics simulators. Areas where simulations may be used include material se-

lection, feasibility and what-if studies, preliminary dynamical properties and weight estimation. This is where the virtual tools are extremely useful and most commonly utilized. For example, in aeronautical industries it is crucial to first build a virtual model and simulate all possible conditions to ensure its performance meets customer's expectations. [6]

The virtual prototyping process for MSC ADAMS is shown in Fig 6. First, a simple model is constructed following "crawl, walk, run" method. In this method the model is built one step at a time and the behaviour tested frequently. This is essential since complete simulation models are prone to be quite complex and their troubleshooting time consuming. It is therefore important to validate the model as often as reasonably possible. In the first model all basic components are inserted: bodies, forces, contacts, joints and motions. [23]

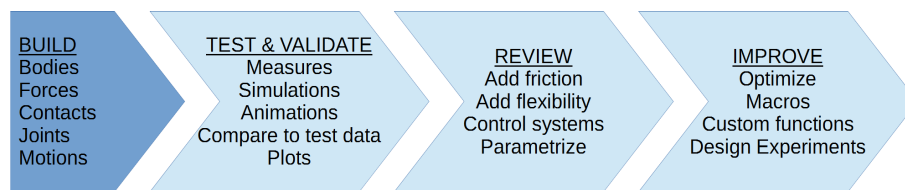


Figure 6: Virtual prototyping process [23]

Next, the model is tested thoroughly. Measures and plots are created and if possible, compared to real world test data. Different simulation scenarios are done according to actual environments planned for the product. Animations and plots help to visualize the results. If the model looks promising it is further developed. Friction is added to crucial points, bodies made flexible to find out strains and position errors, control systems are created to resemble actual controls and important variables parametrizes for easy changing and testing. The goal of all this is to enable new features into the model and make the results correspond more closely to real world measurements. [23]

The last step is to finalize the model and improve its performance and other factors. This is done with optimization routines including design experiments, macros and custom functions. Optimization is performed to make the product more effective, lighter and fine-tune other properties (see Section 3.7.3). In design experiments, several simulations are made with a single command altering the variables each time and recording the results. With custom functions and macros the user can, for example, create and emulate software of the actual product or create complex force functions. [23]

As the process continues and the main features of the product become more accurate, the amount of detailed design increases, see Fig 5. Simulations can be further utilized in this phase when designing the individual parts and sub-assemblies. Often these require special simulation models of their own if more accuracy is required or the user wishes to conduct complex testing. [23]

In Design Evaluation phase the virtual model is evaluated against the customers' needs with different tools are utilized. Some requirements may be abstract properties

such as appearance and usability but often there are also physical quantities. These include rigidity, weight and dynamical performance among other things. Rigidity can be determined with FEA tools, weight with CAD programs and dynamics with multibody simulations. If the design evaluation is successful and all requirements fulfilled, the design is approved for construction. Now a physical model is built and the same properties measured and checked that they correspond to the virtual model. The last steps of the product design process in Fig 5 often require multiple iterations before the final product is complete and ready to be launched to customers.

3 Methods and Tools

This section describes the methods used for solving multibody system problems. It also introduces some the required equations to give the reader an idea of what is needed mathematically. First, basic kinematics are reviewed with appropriate equations. Next, dynamics and then flexible systems and bodies. So called *scalar variational principles*, also known as energy-based methods, are also quickly reviewed as well as some vibration and damping issues. Finally, ADAMS is shortly introduced along with its most significant features. [26] [34] [4]

All equations introduced here are in Cartesian coordinates. While it would be plausible to present them also in polar coordinates, it was considered to unnecessarily complicate issues and make them harder for the reader to understand. Existing approaches can be classified into two sections:

1. Those based on vector variational principles (Newton-Euler)
2. Those based on scalar variational principles (Lagrange)

According to O. A. Bachau [4], multibody systems can be coarsely divided into three categories:

1. Rigid,
2. Linearly elastic and
3. Nonlinearly elastic multibody systems

In this thesis only the first two are considered as the nonlinear materials are often rubbers and soft plastics which do not appear in the course of the two analyses later on.

Ahmed A. Shabana's [34] notation style is used throughout the text. Bold variables refer to vectors or matrices. Variables marked with tilde are skew symmetric matrices (antisymmetric matrices) that satisfy the equation $A = -A^T$. A bar refers to local coordinate system. Variables without a bar are defined in the inertial coordinate system. Plain variables are usually scalars. Description of variable types, for example:

Variable	Type
x_i, θ, t	Scalars
\mathbf{r}_i	Vector or matrix, inertial coordinate system
$\bar{\mathbf{u}}_i$	Vector or matrix, local coordinate system
$\dot{\mathbf{A}}$	Vector or matrix, first time derivative, inertial coordinate system
$\ddot{\bar{\mathbf{r}}}$	Vector or matrix, second time derivative, local coordinate system
$\tilde{\mathbf{v}}$	Skew symmetric matrix, inertial coordinate system
$\tilde{\bar{\mathbf{v}}}$	Skew symmetric matrix, local coordinate system

3.1 Topologies

Multibody systems may be represented with simple graphs – topology diagrams – which show the bodies and their joints without being too intricate on the details. There are three elemental topologies: *chain*, *tree* and *loop*. These options are shown in Fig 7. [11]

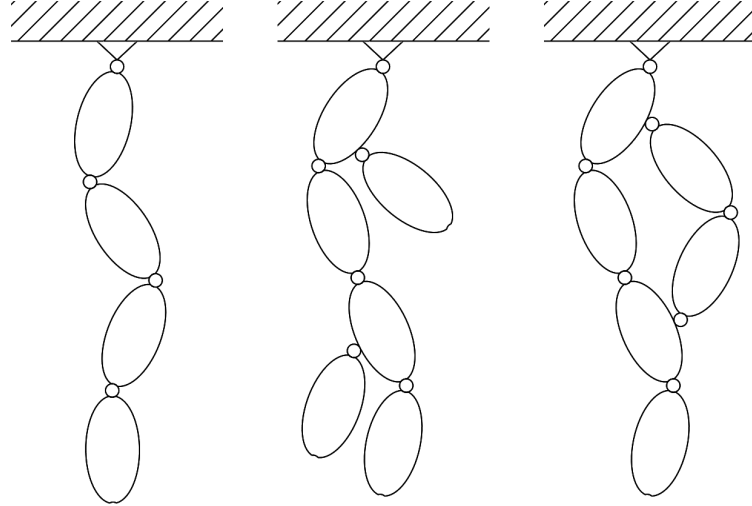


Figure 7: Three different topologies of multibody systems: chain, tree and loop

The chain topology consists of bodies that are linked only to adjacent bodies via a single link. In Fig 7 there are four bodies connected to each other with three links. Additionally, the first body is linked to ground. The chain topology is the simplest to calculate and analyse since each body is only connected to one or two others. [11]

The tree topology resembles the chain but with few additions. There are so called branches of bodies that connect to the bodies in the chain. The result is that some bodies have more than one other bodies connected to them. Fig 7 shows a chain of four bodies with two branching bodies connected to the first and third body with links. A tree complicates the analysing of the system somewhat but since the branched bodies form a chain of their own, the calculating does not differ greatly from the chain topology. [11]

In the loop topology some of the bodies form a loop. Again, there is a chain of four bodies but two extra bodies form a loop with the first and third body on the chain. A loop hinders the calculations even quite considerably in some cases. In a way, the loop creates a feedback mechanism in to the system and thus some recursive formalism operations are not possible leading to time consuming matrix inversions. [11]

3.2 Basic Kinematics

Kinematics is the study of motion regardless of forces and motions acting on the system. That is, the kinematics of a system are defined by the system's geometrical

aspects. *Kinetics* study the motion of the system including the forces. *Dynamics* is kinematics and kinetics combined. [34]

3.2.1 Frame of Reference and Coordinates

A Cartesian coordinate system consists of an origo O and three orthogonal axes, traditionally labelled as \mathbf{X}_1 , \mathbf{X}_2 and \mathbf{X}_3 , or shortly $\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3$. At least two coordinate systems are often required. The first one, often called a *global* or *inertial frame of reference*, is fixed in space and time. The following coordinate systems are attached to each part in the multibody system. These are referred as *body references*. The body reference frame translates and rotates with the body and thus its position is changed in relation to the inertial frame of reference with time. [34]

All the equations and theorems in this thesis are expressed in Cartesian coordinates. However, *generalized coordinates* are more commonly used, for example as in Section 3.3.2. Consider a body in 2D space. To fully position the body three coordinates are required: two for translations along the global axis and one for the orientation. This can be expressed as $\mathbf{q} = [x \ y \ \alpha]$. This position vector is said to consist of three generalized coordinates, that is, the types of these coordinates are ignored. The generalized coordinate representation of this vector would be $\mathbf{q} = [q_1 \ q_2 \ q_3]$. [34]

The concept of generalized coordinates is especially useful in situations where the position vector \mathbf{q} is exceedingly large. This happens very easily as the number of bodies in the system increase. Typically, the position of every body in the system is stored in the generalized coordinate vector \mathbf{q} . [34]

3.2.2 Positions and Translations

An arbitrary point or particle in space can be positioned with three coordinates, x_1 , x_2 and x_3 . The position vector of this particle is thereby

$$\mathbf{r} = x_1\mathbf{i}_1 + x_2\mathbf{i}_2 + x_3\mathbf{i}_3 \quad (3.1)$$

where \mathbf{i}_1 , \mathbf{i}_2 and \mathbf{i}_3 are unit vectors along the X_1 , X_2 and X_3 axes. The velocity vector is the time derivative

$$\mathbf{v} = \dot{\mathbf{r}} = \frac{d}{dt}(\mathbf{r}) = \dot{x}_1\mathbf{i}_1 + \dot{x}_2\mathbf{i}_2 + \dot{x}_3\mathbf{i}_3 \quad (3.2)$$

And acceleration

$$\mathbf{a} = \dot{\mathbf{v}} = \frac{d}{dt}(\mathbf{v}) = \frac{d^2}{dt^2}(\mathbf{x}) = \ddot{x}_1\mathbf{i}_1 + \ddot{x}_2\mathbf{i}_2 + \ddot{x}_3\mathbf{i}_3 \quad (3.3)$$

These can be also represented in matrix form

$$\mathbf{r} = [x_1 \ x_2 \ x_3]^T \quad (3.4)$$

$$\mathbf{v} = [\dot{x}_1 \ \dot{x}_2 \ \dot{x}_3]^T \quad (3.5)$$

$$\mathbf{a} = [\ddot{x}_1 \quad \ddot{x}_2 \quad \ddot{x}_3]^T \quad (3.6)$$

A vector \mathbf{u} in space can be similarly formulated

$$\mathbf{u} = u_1 \mathbf{i}_1 + u_2 \mathbf{i}_2 + u_3 \mathbf{i}_3 \quad (3.7)$$

And in matrix form

$$\mathbf{u} = [u_1 \quad u_2 \quad u_3]^T \quad (3.8)$$

When a body is present in the system, it is denoted as body i . The body coordinate system is correspondingly $\mathbf{X}_1^i \mathbf{X}_2^i \mathbf{X}_3^i$. This coordinate system's unit vectors are indicated similarly as \mathbf{i}_1^i , \mathbf{i}_2^i and \mathbf{i}_3^i .

In 3D space six variables are needed to accurately specify the location and orientation of a body, three for coordinates

$$\mathbf{r}_i = [r_1^i \quad r_2^i \quad r_3^i]^T \quad (3.9)$$

and three for rotations

$$\mathbf{S}_i = [\alpha^i \quad \beta^i \quad \gamma^i]^T \quad (3.10)$$

where the superscript indicates the index of the body in question, such as body i . The location coordinates and rotation angles can be expressed in a single 6×1 position vector

$$\mathbf{q}^i = [r_1^i \quad r_2^i \quad r_3^i \quad \alpha^i \quad \beta^i \quad \gamma^i]^T \quad (3.11)$$

All point, vector and body translation can be done utilizing simple Cartesian vectors such as Eq 3.7. This position vector is commonly denoted as

$$\mathbf{R}^i = [R_1^i \quad R_2^i \quad R_3^i] \quad (3.12)$$

A position vector for an arbitrary point on a body can be expressed as

$$\mathbf{r}^i = \mathbf{R}^i + \mathbf{u}_0^i \quad (3.13)$$

where \mathbf{R}^i is the coordinate vector from the global coordinate system zero to that of the body and \mathbf{u}_0^i is the vector from the body zero to the point itself.

3.2.3 Rotations

Body rotations can be accurately determined by specifying the angle of rotation θ and a unit vector along the axis of rotation \mathbf{v} . The unit vector of rotation is referenced to the inertial coordinate system

$$\mathbf{v} = v_1 \mathbf{i}_1 + v_2 \mathbf{i}_2 + v_3 \mathbf{i}_3 \quad (3.14)$$

3×3 *rotation matrix* presents one body rotation conveniently in a single matrix. The rotation matrix is also know as the *Rodriguez formula* or *transformation matrix*. The equation is

$$\mathbf{A} = [\mathbf{I} + \tilde{\mathbf{v}} \sin \theta + 2(\tilde{\mathbf{v}})^2 \sin^2 \frac{\theta}{2}] \quad (3.15)$$

Where $\tilde{\mathbf{v}}$ is a skew symmetric matrix of the unit vector of rotation. It is defined as

$$\tilde{\mathbf{v}} = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \quad (3.16)$$

Four *Euler parameters* are needed in order to write the rotation matrix completely. The parameters can be acquired from the angle of rotation and the unit vector of rotation (Eq 3.14)

$$\theta_0 = \cos \frac{\theta}{2}, \quad \theta_1 = v_1 \sin \frac{\theta}{2}, \quad \theta_2 = v_2 \sin \frac{\theta}{2}, \quad \theta_3 = v_3 \sin \frac{\theta}{2} \quad (3.17)$$

The transformation matrix can now be expressed with the Euler parameters as

$$\mathbf{A} = \begin{bmatrix} 1 - 2(\theta_2)^2 - 2(\theta_3)^2 & 2(\theta_1\theta_2 - \theta_0\theta_3) & 2(\theta_1\theta_3 + \theta_0\theta_2) \\ 2(\theta_1\theta_2 + \theta_0\theta_3) & 1 - 2(\theta_1)^2 - 2(\theta_3)^2 & 2(\theta_2\theta_3 - \theta_0\theta_1) \\ 2(\theta_1\theta_3 - \theta_0\theta_2) & 2(\theta_2\theta_3 - \theta_0\theta_1) & 1 - 2(\theta_1)^2 - 2(\theta_2)^2 \end{bmatrix} \quad (3.18)$$

The above matrix is used to convert the orientation of a vector within a free body from local coordinates to inertial reference of frame.

The position of a vector \mathbf{u}^i can now be resolved when the matrix and the vector $\bar{\mathbf{u}}^i$ in local coordinates are known

$$\mathbf{u}^i = \mathbf{A}^i \bar{\mathbf{u}}^i \quad (3.19)$$

Additionally, if the position vector \mathbf{R}^i of the local coordinate system in relation to global reference of frame is known, the position of a point at the end of vector $\bar{\mathbf{u}}^i$ in the local system can be calculated with

$$\mathbf{r}^i = \mathbf{R}^i + \mathbf{A}^i \bar{\mathbf{u}}^i \quad (3.20)$$

Successive infinitesimal rotations can easily be determined with

$$\mathbf{A} = \mathbf{e}^{\theta\tilde{\mathbf{v}}} = \exp(\theta\tilde{\mathbf{v}}) \quad (3.21)$$

Both rotation and translation can be expressed in one 4×4 *transformation matrix*. This is sometimes referred to as the *homogeneous transform* and it is of the form

$$\mathbf{A}_4^i = \begin{bmatrix} \mathbf{A}^i & \mathbf{R}^i \\ \mathbf{0}_3^i & 1 \end{bmatrix} \quad (3.22)$$

The equations above are built around the four Euler parameters of rotation. However, there are three degrees of freedom for rotations. An alternative representation uses only three parameters, the *Rodriguez parameters*. Parameter γ is

$$\gamma = \mathbf{v} \tan \frac{\theta}{2} \quad (3.23)$$

And its skew symmetric matrix

$$\tilde{\boldsymbol{\gamma}} = \begin{bmatrix} 0 & -\gamma_3 & \gamma_2 \\ \gamma_3 & 0 & -\gamma_1 \\ -\gamma_2 & \gamma_1 & 0 \end{bmatrix} \quad (3.24)$$

Transformation matrix can now be written as

$$\mathbf{A} = \mathbf{I} + \frac{2}{1 + \gamma^2}(\tilde{\boldsymbol{\gamma}} + \tilde{\boldsymbol{\gamma}}^2) \quad (3.25)$$

The relation to the four Euler parameters $(\theta_0, \theta_1, \theta_2, \theta_3)$ is

$$\gamma_i = \frac{\theta_i}{\theta_0}, \quad i = 1, 2, 3 \quad (3.26)$$

3.2.4 Velocities

The velocity of a point when the angular velocity is known is

$$\begin{aligned} \dot{\mathbf{r}}^i &= \mathbf{v}^i \\ &= \dot{\mathbf{R}}^i + \boldsymbol{\omega}^i \times \mathbf{u}^i \end{aligned} \quad (3.27)$$

$$= \dot{\mathbf{R}}^i + \mathbf{A}^i(\bar{\boldsymbol{\omega}}^i \times \bar{\mathbf{u}}^i) \quad (3.28)$$

$$= \mathbf{J}_{T_i}(\mathbf{y}, t)\dot{\mathbf{y}} + \bar{\mathbf{v}}^i(\mathbf{y}, t) \quad (3.29)$$

where $\boldsymbol{\omega}$ is called the *angular velocity vector* and it is defined in the inertial coordinate system. $\dot{\mathbf{R}}^i$ is the velocity of the global origin. The superscript i once again indicates the body index. Eq 3.29 is according to W. Schielen [31]. \mathbf{J}_{T_i} is a $3 \times n$ Jacobian matrix of virtual translational displacement, n is the number of generalized coordinates and \mathbf{y} is a $1 \times n$ vector of these generalized coordinates.

The angular velocity vector itself in body frame of reference can be determined with

$$\bar{\boldsymbol{\omega}} = 2\dot{\mathbf{v}} \times \mathbf{v} \sin^2 \frac{\theta}{2} + \dot{\mathbf{v}} \sin \theta + \mathbf{v}\dot{\theta} \quad (3.30)$$

Where \mathbf{v} is the unit vector of rotation, size 3×1 . The same velocity but with reference to inertial frame of reference is

$$\boldsymbol{\omega} = 2\mathbf{v} \times \bar{\mathbf{v}} \sin^2 \frac{\theta}{2} + \dot{\mathbf{v}} \sin \theta + \mathbf{v}\dot{\theta} \quad (3.31)$$

$$= \mathbf{J}_{T_i}(\mathbf{y}, t)\dot{\mathbf{y}} + \bar{\boldsymbol{\omega}}^i(\mathbf{y}, t) \quad (3.32)$$

where Eq 3.32 is again from W. Scheilen's article [31]. Now \mathbf{J}_{T_i} refers to a $3 \times n$ Jacobian matrix of virtual rotational displacement.

The velocity of an arbitrary point can likewise be determined with the transformation matrix. Position vector from Eq 3.20 is

$$\mathbf{r}^i = \mathbf{R}^i + \mathbf{A}^i \bar{\mathbf{u}}^i \quad (3.33)$$

Time derivative of this equation is

$$\dot{\mathbf{r}}^i = \dot{\mathbf{R}}^i + \dot{\mathbf{A}}^i \bar{\mathbf{u}}^i \quad (3.34)$$

3.2.5 Accelerations

The velocity of a point is

$$\begin{aligned}\mathbf{a}^i &= \ddot{x}^i \\ &= \ddot{\mathbf{R}}^i + \dot{\boldsymbol{\omega}}^i \times \mathbf{u}^i + \boldsymbol{\omega}^i \times \dot{\mathbf{u}}^i\end{aligned}\quad (3.35)$$

$$= \mathbf{J}_{Ti}(\mathbf{y}, t)\ddot{\mathbf{y}} + \frac{\partial \mathbf{v}^i}{\partial \mathbf{y}^T} \dot{\mathbf{y}} + \frac{\partial \mathbf{v}^i}{\partial t}\quad (3.36)$$

The velocity with the transformation matrix is

$$\dot{\mathbf{r}} = \mathbf{A}\dot{\bar{\mathbf{r}}} + \tilde{\boldsymbol{\omega}}\mathbf{A}\bar{\mathbf{r}}\quad (3.37)$$

And it's first derivative accordingly

$$\ddot{\mathbf{r}} = \dot{\mathbf{A}}\dot{\bar{\mathbf{r}}} + \mathbf{A}\ddot{\bar{\mathbf{r}}} + \dot{\tilde{\boldsymbol{\omega}}}\mathbf{A}\bar{\mathbf{r}} + \tilde{\boldsymbol{\omega}}\dot{\mathbf{A}}\bar{\mathbf{r}} + \tilde{\boldsymbol{\omega}}\mathbf{A}\dot{\bar{\mathbf{r}}}\quad (3.38)$$

Rearranging terms and substituting some of the terms yields

$$\begin{aligned}\ddot{\mathbf{r}} &= \mathbf{A}\ddot{\bar{\mathbf{r}}} + 2\boldsymbol{\omega} \times \mathbf{v}_g + \boldsymbol{\alpha} \times \mathbf{r} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) \\ &= \mathbf{a}_l + \mathbf{a}_c + \mathbf{a}_t + \mathbf{a}_n\end{aligned}\quad (3.39)$$

where $\mathbf{v}_g = \dot{\mathbf{A}}\bar{\mathbf{r}}$ is the time derivative of the vector $\dot{\mathbf{r}}$ in inertial coordinate system. Eq 3.39 consists of four different variables: \mathbf{a}_l , \mathbf{a}_c , \mathbf{a}_t and \mathbf{a}_n . Here, \mathbf{a}_l represents the acceleration projected to the point by the *acceleration of the global coordinate system*. The other three acceleration components are: *Coriolis* \mathbf{a}_c , *tangential* \mathbf{a}_t and *normal* \mathbf{a}_n . Furthermore, the *angular acceleration vector*

$$\begin{aligned}\boldsymbol{\alpha} &= \dot{\boldsymbol{\omega}} \\ &= 2 \begin{bmatrix} \ddot{\theta}_3\theta_2 & - & \ddot{\theta}_2\theta_3 & + & \ddot{\theta}_1\theta_0 & - & \ddot{\theta}_0\theta_1 \\ \ddot{\theta}_1\theta_3 & - & \ddot{\theta}_0\theta_2 & - & \ddot{\theta}_3\theta_1 & + & \ddot{\theta}_2\theta_0 \\ \ddot{\theta}_2\theta_1 & + & \ddot{\theta}_3\theta_0 & - & \ddot{\theta}_0\theta_3 & - & \ddot{\theta}_1\theta_2 \end{bmatrix}\end{aligned}\quad (3.40)$$

$$= \mathbf{J}_{Ri}(\mathbf{y}, t)\ddot{\mathbf{y}} + \frac{\partial \boldsymbol{\omega}^i}{\partial \mathbf{y}^T} \dot{\mathbf{y}} + \frac{\partial \boldsymbol{\omega}^i}{\partial t}\quad (3.41)$$

3.3 Dynamics

Solving problems where there are forces present in moving mechanisms or systems is called *kinetics* or *dynamic force analysis*. Robert L. Norton introduces the concepts of two subclasses of dynamics problems [26]:

1. *The forward dynamics* problems, where the forces and masses are known and velocities or accelerations must be solved.
2. *The inverse dynamics* deal with problems, where accelerations, velocities or displacements are known and forces required to achieve these motions need to be solved. This is sometimes called *kinetostatics*.

Dynamics problems can be solved in three different methods:

1. Classical Newtonian dynamics
2. Virtual work
3. Lagrangian dynamics

The classical Newtonian method – which is a so called *Vector Variational Principle* – is by far the easiest to use when calculating simple problems by hand. The concepts of virtual work and Lagrangian dynamics are more complex but much more practical to work with when solving complicated systems. Simple problems can nevertheless also be solved by hand with these two methods. Virtual work and Lagrangian dynamics are so called *Scalar Variational Principles*, that is they function by scalars such as work and energy as opposed to Newtonian method which operates with vectors such as velocity and force. [3]

3.3.1 Newtonian Method

The Newton's three *laws of motion* are the basis of any dynamics system [24]:

1. When no external forces act on a body, it remains at rest or in motion at constant velocity
2. The momentum of a body changes according to the magnitude of the applied force and its direction
3. For every action force there is an equal and opposite reaction force

Newton-Euler equations are

$$\begin{cases} \mathbf{F} = m\mathbf{a} \\ \mathbf{T} = \mathbf{I}\alpha \end{cases} \quad (3.42)$$

where

- \mathbf{F} is the force acting on the center of mass of the body, N
- m is the mass of the body, kg
- \mathbf{a} is the acceleration of the center of mass of the body, m/s^2
- \mathbf{T} is the torque acting on the body, Nm
- \mathbf{I} is the mass moment of inertia around the center of mass of the body, kgm^2
- α is the angular acceleration of the body, rad/s^2

The mass moment of inertia, \mathbf{I} can be either a scalar or a matrix. As a 3×3 matrix it is known as *inertia tensor*

$$\mathbf{I} = \begin{bmatrix} I_{XX} & I_{XY} & I_{XZ} \\ I_{YX} & I_{YY} & I_{YZ} \\ I_{ZX} & I_{ZY} & I_{ZZ} \end{bmatrix} \quad (3.43)$$

where the elements I_{ij} are known as *mass moments of inertia* and the non-diagonal elements *products of inertia*.

3.3.2 Equations of Motion

The principle equation of motion is Eq 3.42

$$\mathbf{F} = m\mathbf{a} \quad (3.44)$$

In literature this equation takes the form [31] [11]

$$\mathbf{M}(\mathbf{y}, t)\ddot{\mathbf{y}} + \mathbf{k}(\mathbf{y}, \dot{\mathbf{y}}, t) = \mathbf{q}(\mathbf{y}, \dot{\mathbf{y}}, t) \quad (3.45)$$

where \mathbf{y} is a $1 \times n$ vector of generalized coordinates and n the number of these coordinates. \mathbf{M} is a $n \times n$ symmetric inertia matrix. The $n \times 1$ vector \mathbf{k} describes the generalized Coriolis forces and the $n \times 1$ vector \mathbf{q} the generalized applied forces.

Eq 3.45 can further be linearized in case of small vibrations resulting in [11]

$$\mathbf{M}(t)\ddot{\mathbf{y}} + \mathbf{P}(t)\dot{\mathbf{y}} + \mathbf{Q}(t)\mathbf{y} = \mathbf{h}(t) \quad (3.46)$$

this equation is appears very familiar. It is essentially the same as the equation for a simple spring-damper system [30]

$$m\ddot{x} + c\dot{x} + kx = 0 \quad (3.47)$$

except the mass, damping coefficient and spring stiffness are replaced with corresponding matrices \mathbf{M} , \mathbf{P} and \mathbf{Q} . Additionally, Eq 3.46 is presented in generalized coordinates.

Differential equations as above may – in simple cases – be solved symbolically. Another way of solving them is by using integrators and raw computing power. See Section 3.7 for more information about the solvers in MSC ADAMS.

3.3.3 Virtual Work

The concept of virtual work is constructed around the idea that: "if a system would move an arbitrary virtual displacement of $\delta\mathbf{r}^i$, δW amount of virtual work would be done." Then, by setting the virtual displacement to zero, force equations are gained and the state of the system successfully determined. [34]

For a particle or body in static equilibrium it holds that the resultant vector

$$\mathbf{F}^i = 0 \quad (3.48)$$

And thus for any virtual displacement

$$\sum \mathbf{F}^i \cdot \delta\mathbf{r}^i = \delta W = 0 \quad (3.49)$$

Furthermore, if we split the resultant force and virtual work to two parts, one marking the constraint forces and the other external forces we get

$$\mathbf{F}^i = \mathbf{F}_c^i + \mathbf{F}_e^i \quad (3.50)$$

$$\delta W = \delta W_c + \delta W_e = 0 \quad (3.51)$$

If we assume the constraints do no work, i.e. they are *workless constraints*, the virtual work is zero

$$\delta W_c = \sum \mathbf{F}_c^i \cdot \delta \mathbf{r}^i = 0 \quad (3.52)$$

For the external forces we get

$$\delta W = \delta W_e = \sum \mathbf{F}_e^i \cdot \delta \mathbf{r}^i = 0 \quad (3.53)$$

This is known as *the principle of virtual work in static equilibrium*. For dynamic systems we must include the momentum's rate of change in the equations

$$\mathbf{F}^i - \dot{\mathbf{P}}^i = 0 \quad (3.54)$$

If the constraints are workless, the Eq 3.52 remains the same. For Eq 3.53 we now obtain

$$\sum (\mathbf{F}^i - \dot{\mathbf{P}}^i) \cdot \delta \mathbf{r}^i = 0 \quad (3.55)$$

This equation is also known as *D'Alembert's principle*. [10]

3.3.4 Lagrangian and Hamiltonian Dynamics

The Lagrangian mechanics or dynamics is another way of solving the equations of motion for different kinds of systems. It is a principle of least action, that is, the system moves where it requires or uses the least energy or work. The Lagrangian method is widely used in different areas of analyses in mechanical and structural engineering. For example, static, dynamic and vibration problems are commonly solved using the tools presented in this section.

The *Lagrangian* is

$$L = T - V \quad (3.56)$$

where T is the kinetic energy and V the potential energy of the system. Once these energies have been calculated and L determined, the equation of motion is

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} - Q_j = 0 \quad (3.57)$$

where q_j is generalized coordinate and Q_j corresponding generalized force. The most common equations for kinetic and potential energies, for example, are $1/2mv^2$ and mgh . [19]

From the above equations another technique of calculating dynamics can be obtained, the *Hamiltonian* method. It is defined as

$$H = \dot{\mathbf{q}} \cdot \mathbf{p} - L \quad (3.58)$$

The equation above is the *Legendre transform* of the Lagrangian [19]. Noting that $p = \frac{\partial L}{\partial \dot{q}_j}$ we obtain

$$H = \sum \frac{\partial L}{\partial \dot{q}_k} \dot{q}_k - L \quad (3.59)$$

3.4 Flexible Bodies

Sometimes it is useful to model bodies as flexible entities instead of rigid ones. Rigid bodies, by definition, do not flex or deform at all. That is, the distance between two arbitrary points remains constant. Consequently, a force acting on a rigid body creates a linear acceleration throughout the body with an angular acceleration about its center of mass. [34]

Deformable bodies, on the other hand, are not rigid. They flex and twist according to the forces acting on them. This branch of mechanics is referred to as *Continuum mechanics*. Continuum mechanics focuses on the motion of flexible bodies. Introducing flexibility to dynamics causes new problems and possibilities. For instance, applying a point force on a deformable body does not cause a linear acceleration throughout the body. Instead, the other points on the body are not set in motion instantaneously but in relation to the rigidity of the body. [34]

3.4.1 Kinematics

The position of points can now be expressed with an additional vector \mathbf{u}_f^i which indicates the vector from the original point on a body to the point after the body has deformed. Consider Eq 3.13. It now becomes

$$\mathbf{r}^i = \mathbf{R}^i + \mathbf{u}_0^i + \mathbf{u}_f^i \quad (3.60)$$

Similarly the velocity, Eq 3.28

$$\mathbf{v}^i = \dot{\mathbf{R}}^i + \boldsymbol{\omega}^i \times (\mathbf{u}_0^i + \mathbf{u}_f^i) + \dot{\mathbf{u}}_f^i \quad (3.61)$$

And acceleration, Eq 3.35

$$\mathbf{a}^i = \ddot{\mathbf{R}}^i + \boldsymbol{\omega}^i \times (\boldsymbol{\omega}^i \times \mathbf{u}^i) + \boldsymbol{\alpha}^i \times \mathbf{u}^i + 2\boldsymbol{\omega}^i \times (\dot{\mathbf{u}}_f^i) + \ddot{\mathbf{u}}_f^i \quad (3.62)$$

3.4.2 Strain

Strain is the relative displacement of a point on a body from an undeformed state to a deformed one [29]. Consider the distance between two points when the body is rigid as δl_0 . On a deformed body this same distance is δl . The strain in this case is

$$\varepsilon = \frac{\delta l}{\delta l_0} - 1 \quad (3.63)$$

In 3D space there are nine different *strain components* forming a 3 x 3 symmetric matrix called the *Lagrangian strain tensor*

$$\begin{aligned} \boldsymbol{\varepsilon}_m &= \frac{1}{2}([\bar{\mathbf{J}}^T + \bar{\mathbf{J}}] + \bar{\mathbf{J}}^T \bar{\mathbf{J}}) = \frac{1}{2}(\bar{\mathbf{J}}^T \bar{\mathbf{J}} - \mathbf{I}) \\ &= \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{21} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & \varepsilon_{33} \end{bmatrix} \end{aligned} \quad (3.64)$$

where $\bar{\mathbf{J}}$ is the *gradient* of the displacement vector \mathbf{u} which itself is defined as

$$\bar{\mathbf{J}} = \begin{bmatrix} \partial u_1 / \partial x_1 & \partial u_1 / \partial x_2 & \partial u_1 / \partial x_3 \\ \partial u_2 / \partial x_1 & \partial u_2 / \partial x_2 & \partial u_2 / \partial x_3 \\ \partial u_3 / \partial x_1 & \partial u_3 / \partial x_2 & \partial u_3 / \partial x_3 \end{bmatrix} \quad (3.65)$$

As the Lagrangian strain tensor is a symmetric matrix, its contents can be sufficiently determined with six components. These components form the *strain vector*

$$\boldsymbol{\varepsilon} = [\varepsilon_{11} \quad \varepsilon_{22} \quad \varepsilon_{33} \quad \varepsilon_{12} \quad \varepsilon_{13} \quad \varepsilon_{23}]^T \quad (3.66)$$

Alternatively, the strain components can be calculated with the equation

$$\varepsilon_{ij} = \frac{1}{2} \left(u_{i,j} + u_{j,i} + \sum_k u_{k,i} u_{k,j} \right), \quad i, j = 1, 2, 3 \quad (3.67)$$

When the strains and rotations are small, such as in engineering applications, the product $\bar{\mathbf{J}}^T \bar{\mathbf{J}}$ can be neglected, that is, $\bar{\mathbf{J}}^T \bar{\mathbf{J}} \approx 0$. Eq 3.64 now becomes

$$\boldsymbol{\varepsilon}_m \approx \frac{1}{2} (\bar{\mathbf{J}}^T + \bar{\mathbf{J}}) \quad (3.68)$$

3.4.3 Stress

Stress is the pressure present within a deforming part. It depends on strain affecting the part at the moment and material properties. [29]

Similarly to the strain tensor, a 3 x 3 matrix for the *stress tensor* can be defined as

$$\boldsymbol{\sigma}_m = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \quad (3.69)$$

where σ_{ij} are the *stress components*. Correspondingly, a similar *stress vector* is

$$\boldsymbol{\sigma} = [\sigma_{11} \quad \sigma_{22} \quad \sigma_{33} \quad \sigma_{12} \quad \sigma_{13} \quad \sigma_{23}]^T \quad (3.70)$$

The Eqs 3.66 and 3.70 together are the *constitutive equations* and essentially define the behaviour of the material. *Hooke's law* determines the relation between strain and stress. It is [29]

$$\boldsymbol{\sigma} = \mathbf{E} \boldsymbol{\varepsilon} \quad (3.71)$$

where \mathbf{E} is a 6 x 6 matrix of *elastic constants*. Hooke's law is only valid for linearly behaving materials, such as metals under the elastic limit. Plastic or viscoelastic deformations are not covered by Eq. 3.71.

3.4.4 Discrete Flexible Link

One of the primary flexible elements is a beam, called a *Discrete Flexible Link* in MSC ADAMS. A typical I-beam is shown in Fig 8. The properties of beam depend

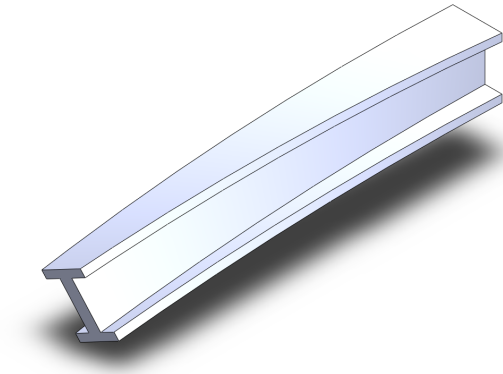


Figure 8: A bent I-beam

on three parameters: length, moment of inertia and elastic modulus. For example the displacement at the end of a beam fixed on one end is [29]:

$$v = \frac{FL^3}{3EI} \quad (3.72)$$

where F is the force at the end of the beam, L is the length of the beam, E is the beam's material's elastic modulus and I is the moment of inertia.

A discrete flexible link is a beam divided into a number of segments attached together. The segments have a mass and geometry according to the original beam but they are connected together with springs of the Eq 3.72. An eight segment beam in ADAMS is shown in Fig 9. The beam is in yellow and the red shapes between the segments are springs connecting the segments together. The characteristics of the

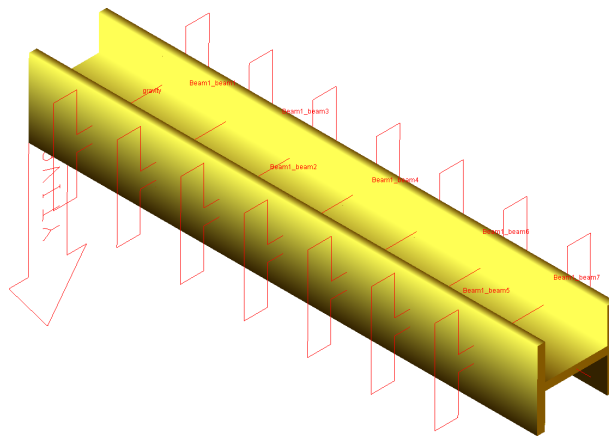


Figure 9: A discrete flexible link in ADAMS

beam, such as the geometry and material properties, are entered to the link tool. ADAMS then calculates appropriate forces and masses for the beam automatically.

The properties of the beam may be later modified. The more the segments there are in the discrete flexible link, the closer it behaves to an actual beam.

3.4.5 Area Moments of Inertia

Area moments of inertia are geometrical properties of beams used to calculate beam displacements, for example. In Fig 10 is shown necessary needed dimensions to determine area moments of inertia for rectangular and circular profiles. [29]

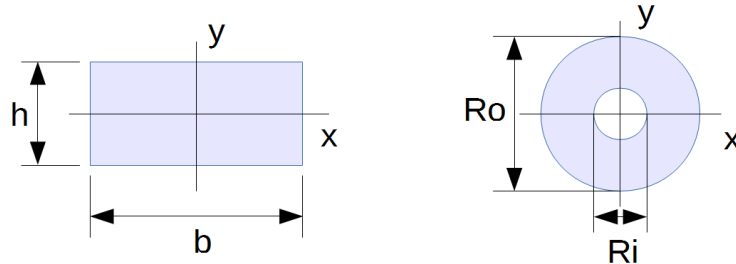


Figure 10: Dimension for rectangular and circular cross sections

The general equations for calculating area moments of inertia are [29]

$$I_x = \int \int y^2 dA \quad (3.73)$$

$$I_y = \int \int x^2 dA \quad (3.74)$$

The equations for rectangles whose centroid is at origin are

$$I_x = \frac{b \cdot h^3}{12} \quad (3.75)$$

$$I_y = \frac{b^3 \cdot h}{12} \quad (3.76)$$

And for hollow cylinders

$$I_x = \frac{\pi}{4}(R_o^4 - R_i^4) \quad (3.77)$$

$$= I_y \quad (3.78)$$

3.5 Degrees of Freedom and Joints

In 3D space an unconstrained body has six *degrees of freedom*: three for translations along the three coordinate axes and three for rotations around the same axes. Constraints reduce the number of degrees of freedom by limiting the bodies' movement. The equation for calculating the number is sometimes called the *Kutzbach criterion* and it is [34]

$$m = 6 \cdot n_b - n_c \quad (3.79)$$

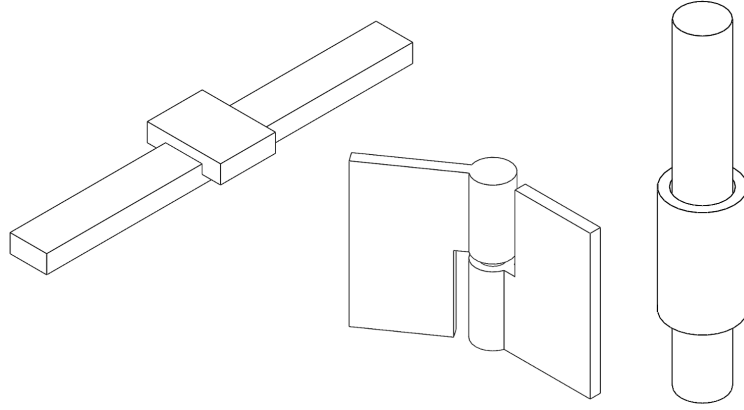


Figure 11: Three different joints: translational, revolute and cylindrical

where n_b is the number of bodies in the system and n_c the number of independent constrain equations. Another – perhaps more commonly used – equation is the *Gruebler count* [5]:

$$m = 6 \cdot (n_b - 1) - n_c \quad (3.80)$$

The difference between the Kutzbach criterion and Gruebler count is that the latter considers the global ground as one body whereas Kutzbach does not. The end result is the same regardless of which method is utilized. The number of free DOFs directly relates to how the mechanism is able to manoeuvre. If the number of free degrees of freedom is zero, the mechanism is locked and cannot move except in the case where one of the constrains is a motion. If there are one or more free DOFs the mechanism is able to move in one or more direction. Usually, the desired number of free DOFs is zero or one. It is also possible for a system to have a negative number of freedoms, it is then called *overconstrained*. Typically in this case, there are redundant constrains in the system and the mechanism itself might still be able to move. For example, if all four joints in Fig ?? are revolute type, the joints remove the same DOFs four times. One of these freedoms are translation and rotation along axis perpendicular to the viewer.

In Fig 11 there are shown three different joints: translational, revolute and cylindrical. These are among the most common types of joints. In Table 1 are listed all types of joints MSC ADAMS supports and how many numbers of freedom they limit. For a translational joint, for example, the table shows that two translations and all three rotations are limited leaving only one translational direction free. As such, it constraints five DOFs. The revolute joint is similar to translational since it also allows one degree of freedom. The cylindrical joint restricts in total four DOFs: two translational and two rotational.

Joints that have a surface contact are called *lower pairs*. *Higher pair* joints have a point or a line contact. For example, spherical constraint has a point contact and thus is a higher pair joint. Revolute joint has a line contact since it rotates around an axis. Planar joint is an example of a lower pair constraint.

Further classification of constraints [34]:

Table 1: Types of joints used MSC ADAMS and the number of freedoms they remove [5]

Constrain element	Translational constraints	Rotational constraints	Coupled constraints	Total constraints
Joints				
Fixed	3	3	0	6
Translational	2	3	0	5
Revolute	3	2	0	5
Cylindrical	2	2	0	4
Universal	3	1	0	4
Planar	1	2	0	3
Spherical	3	0	0	3
Rack-and-pinion	0	0	1	1
Joint primitives				
Atpoint	3	0	0	3
Orientation	0	3	0	3
Inline	2	0	0	2
Parallel	0	2	0	2
Inplane	1	0	0	1
Perpendicular	0	1	0	1
Motions				
Translational	1	0	0	1
Rotational	0	1	0	1
Coupler	0	0	1	1

- *Holonomic* constraints restrict the position of a system. The constrain equation can be written as

$$\mathbf{C}(q_1, q_2, \dots, q_n, t) = \mathbf{C}(\mathbf{q}, t) = 0 \quad (3.81)$$

- *Rheonomic* constraints depend on time, that is, time appears explicitly in the above equation
- *Scleronomic* constraints do not depend on time

where C stands for constraint equation. A joint may be holonomic and rheonomic or holonomic and scleronomic. If a constraint equation cannot be expressed in the form of Eq 3.81 it is called a *nonholonomic* constraint. For example, if the constraint equation involves a coefficient matrix it cannot be integrated and expressed in the form of Eq 3.81 and thus is nonholonomic. An example of such a constraint could be a rubber shock absorber whose properties are strongly nonlinear and must be given in a matrix.

If a constraint contains an equality, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}, t) = 0$, it is called *two-sided* and *restrictive* or *limiting*. In the event of inequality, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}, t) \geq 0$, a constraint is referred to as *one-sided* and *nonrestrictive* or *nonlimiting*. [34]

3.6 Vibrations and Damping

Vibration theory is closely related to dynamical analysis of multibody systems [11]. The same dynamic equations and matrices function in the background of both fields. As such, both areas of problems can just as easily be solved with multibody dynamics software.

The equation of motion for a simple spring-damper system is [13]

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f}(t) \quad (3.82)$$

here, the \mathbf{x} is generalized coordinate, \mathbf{M} is mass matrix, \mathbf{C} is damping matrix, \mathbf{K} is spring constant matrix and \mathbf{f} is external force. The equation is of differential type and of second degree. Differential equations can be solved directly with integrators or symbolically. Symbolic calculation is easy when the system is simple. Some symbolic equations are presented next.

The *characteristic equation* for this kind of DE is

$$m\lambda^2 + c\lambda + k = 0 \quad (3.83)$$

From this equation few additional parameters can be determined. Inspecting a critically damped system in which response does not involve oscillation, the *natural frequency* is defined as

$$\omega_n = \sqrt{\frac{k}{m}} \quad (3.84)$$

where k is the spring constant and m the mass. For systems with damping, the *damping ratio* is

$$\zeta = \frac{c}{2m\omega_n} \quad (3.85)$$

where c is the damping factor. Damping also affects to system's natural frequency. The natural frequency of a damped system is

$$\omega_d = \omega_n \sqrt{1 - \zeta^2} \quad (3.86)$$

3.6.1 Mass-spring Absorber

A simple vibration absorber is constructed of a mass and a spring by choosing their values carefully. When properly adjusted, the absorber suppresses all vibration in a system with a harmonic excitation. [30] Consider Fig 12. A harmonic excitation of $F(t) = F_0 \sin(\omega t)$ affects the mass m . The mass is on two springs each with a stiffness of $k/2$. The absorber consists of mass m_a and spring k_a . The amplitude of the excitation X is [30]

$$X = \frac{(k_a - m_a \omega^2) F_0}{(k + k_a - m \omega^2)(k_a - m_a \omega^2) - k_a^2} \quad (3.87)$$

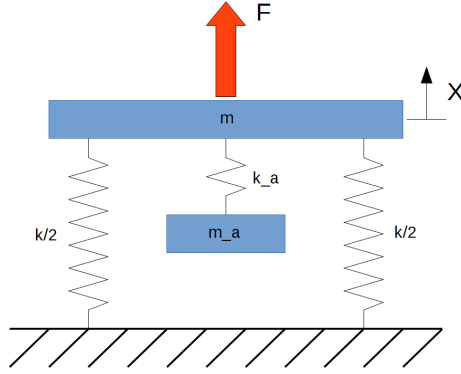


Figure 12: A mass-spring absorber suppresses harmonic excitation on a mass-spring system

If the amplitude is set to zero the vibrations are completely damped. From above

$$X = (k_a - m_a \omega^2) F_0 = 0 \quad (3.88)$$

$$\Rightarrow k_a - m_a \omega^2 = 0 \quad (3.89)$$

$$\Rightarrow \frac{k_a}{m_a} = \omega^2 \quad (3.90)$$

Now the amplitude of the absorber mass is

$$X_a = -\frac{F_0}{k_a} \quad (3.91)$$

We can see the force F_0 is directly affecting only to the absorber, fully negating its effect to the main mass m .

The Eq 3.90 can be easily converted from linear motion to rotation motion by changing the masses to inertias and stiffnesses to rotational ones [30]

$$\frac{k_a}{m_a} = \omega^2 \quad (3.92)$$

$$\Rightarrow \frac{k}{J} = \omega^2 \quad (3.93)$$

3.6.2 Modal Analysis

Modal analysis aims to determine and solve the vibrational modes and frequencies of parts and structures. The vibrational modes are resonance points where the part starts to vibrate with excessive oscillatory motion. These modes are mainly the result of the inertial and elastic properties of the materials in the part or system. Modal analysis is typically done for structures or parts to assess their dynamic behaviour and resonance frequencies, see Fig 13. [14]

In analytical modal analysis, physical coordinates, \mathbf{x} , are replaced by modal coordinates, \mathbf{r} . This is allowed by eigenvectors \mathbf{v} . With physical coordinates the differential equations describing the motion are coupled, that is, they must be solved

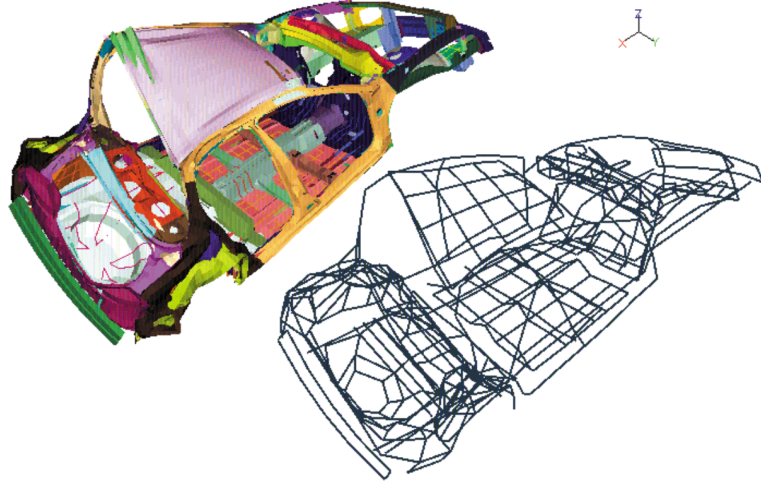


Figure 13: Modal analysis of a car body [14]

together. This is potentially problematic and slows down the solving. When expressed with nodal coordinates these equations are uncoupled and they can be solved independently. [30]

This change of variables is done by a matrix of mode shapes S [30]

$$\mathbf{r} = S^{-1}\mathbf{x} \quad (3.94)$$

where $S = M^{-1/2}P\mathbf{r}$. For example a simple mass-spring system

$$M\ddot{\mathbf{x}} + K\mathbf{x} = \mathbf{0} \quad (3.95)$$

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{x}}_1 \\ \ddot{\mathbf{x}}_2 \end{Bmatrix} + \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{Bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad (3.96)$$

Becomes [30]

$$\begin{Bmatrix} \ddot{r}_1 \\ \ddot{r}_2 \end{Bmatrix} + \begin{bmatrix} \omega_1^2 & 0 \\ 0 & \omega_2^2 \end{bmatrix} \begin{Bmatrix} r_1 \\ r_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad (3.97)$$

3.6.3 Modal Neutral Files

Modal Neutral Files (MNFs) are, in essence, modal descriptions of flexible bodies. They are the primary method of handling flexible bodies in MSC ADAMS and they are generated in finite element analysis programs such as MSC NASTRAN. One MNF file contains the following information [9]:

- Geometry including node location and connectivity. The nodes are generated from the mesh the FEA program creates when constructing the MNF file.
- Nodal mass and inertia. Each node is assigned to a mass according to material parameters. Accordingly, the inertia is also calculated.

- Mode shapes. A flexible body has numerous modes of vibration, the so called normal modes. The frequencies and shapes of these modes are included in the MNF file. The flexible body deformations are, basically, a linear combination of these deformation shapes.
- Generalized mass and stiffness for mode shapes. The stiffness of the body is essential when working with flexible bodies. When the stiffness and external forces are known, the strains can be calculated as well as stresses.

Generally, a MNF file includes the data of the first ten normal modes as they are often the most important. The first normal mode has the lowest frequency and the following modes have increasingly higher. The higher the excitation frequency equals to lower vibration amplitudes, rendering them improbable to have a significant impact. Therefore, usually only the first ten or so modes are included in the MNF file.

3.6.4 Natural Frequencies of Beams

The natural frequency of a beam can be calculated with equation [27]

$$\omega_n = A \sqrt{\frac{EI}{\mu L^4}} \quad (3.98)$$

where A is the area of the beam's cross section, E is elastic modulus, I is area moment of inertia, μ is the mass per unit length of beam and L the length of the beam.

3.7 MSC ADAMS

MSC ADAMS (Automatic Dynamic Analysis of Mechanical Systems) is a widely used multibody dynamics simulation software produced by MSC Software Corporation. According to MSC [21]:

Adams lets you build and test virtual prototypes, realistically simulating on your computer, both visually and mathematically, the full motion behaviour of your complex mechanical system designs.

Adams offers multiple types of simulations and many tools to help build models as accurately as possible. In this section Adams is introduced and some important features and functionalities presented.

3.7.1 Simulation Types

In MSC ADAMS, there are three different simulation types from which the user can choose the most appropriate one. These are: *kinematic*, *static* and *dynamic*. [9]

The kinematic mode is applicable when there are zero degrees of freedom. Any movement in the system is done by forced motions at joints and there are no freely

moving parts. In this mode, ADAMS uncouples the motion and force equations and first solves positions, then velocities, accelerations and forces algebraically. [5] [9]

In static or *quasi-static* mode reaction forces are determined so that they balance out the external forces and loads and the whole system is in equilibrium independently at each time step. From equilibrium it follows that all velocities and accelerations are zero and that there is no movement in the system. This mode is useful if inertial effects need not be taken to account. Static equilibrium can also be determined before starting a dynamic simulation. For example, in vehicle simulation it is important to place the vehicle on the ground before attempting a dynamic simulation. [5]

When the system has at least one degree of freedom, a *dynamic* analysis is required. In dynamic simulation, the differential equations are automatically formulated and numerically solved to determine the system's components' positions, velocities, accelerations and forces. [5]

The simulations are all done at predetermined time steps, known as integration time steps. The step size is free for user to select though the program may compute additional time points between the steps. The user also sets a simulation stop time, that is, the time limit how long the simulation is run. [5]

3.7.2 Solvers

There are three types of equations ADAMS must solve depending on the model and simulation type: *linear equations*, *non-linear equations* and *integration*. [5]

Linear equations are equation sets that may be expressed as

$$\mathbf{A} \mathbf{x} = \mathbf{b} \quad (3.99)$$

where \mathbf{A} is square matrix of constants, \mathbf{x} a vector of unknowns and \mathbf{b} a vector of constants. Typically, matrix \mathbf{A} is very sparse, meaning there is a great number of zeros compared to non-zero values [5]. Eqs such as 3.99 can be solved using *LU Decomposition* where the matrix \mathbf{A} is decomposed or factorized into the product of two new matrices: lower triangular \mathbf{L} and upper triangular \mathbf{U}

$$\mathbf{A} = \mathbf{L} \mathbf{U} \quad (3.100)$$

First is solved equation $\mathbf{L} \mathbf{y} = \mathbf{b}$ for \mathbf{y} by forward substitution. Then, $\mathbf{U} \mathbf{x} = \mathbf{y}$ for \mathbf{x} by back substitution. [2]

Non-linear equations are more difficult to solve and require a numerical approach. A set of non-linear equations are of the form

$$\mathbf{G} \mathbf{x} = 0 \quad (3.101)$$

where \mathbf{G} is a set of implicit equations depending on \mathbf{x} . ADAMS utilizes a method called *Newton-Raphson iteration* [9]. This technique is based on the hypothesis that as the non-linear equation approaches zero, $G = 0$, the curve can be approximated

to a straight line. The derivative of this curve at n th iteration at point (x_n, G_n) is $(\partial G/\partial x)$. If the presumed value of G at point $G = 0$ is (x_0, G_0) , it follows that

$$\frac{\partial G}{\partial x} = \frac{G_0 - G_n}{x_0 - x_n} = \frac{\Delta G}{\Delta x} \quad (3.102)$$

and rearranging

$$\Delta G = \frac{\partial G}{\partial x} \Delta x \quad (3.103)$$

Now ΔG is a vector of current error of each equation. The solution has been reached when this error approaches zero. A commonly used convergence criteria is $-10^{-4} \leq \Delta G \leq 10^4$. [5] [9]

Integration methods are required for dynamic analyses as they cannot be determined algebraically alone. These simulations include a number of both differential and algebraic equations (DAEs). ADAMS has two main types of integrators: *stiff* and *non-stiff*. Stiff-solvers utilize an implicit, backward difference formulations (BDFs) to solve DAEs whereas Non-stiff ones use explicit formulations ordinary differential equations determined from the BDFs. [9]

There are four stiff integrators in ADAMS. These are: *GSTIFF*, *WSTIFF*, *Constant BDF* and *RKF45*. There is also one non-stiff integrator, *Adams-Bashforth-Adams-Moulton (ABAM)*. Additionally, there are two newer integrators which are neither stiff or non-stiff. They are: *Hilber-Hughes-Taylor (HHT)* and *Newmark*. In Fig 14 the different integrators are listed as well as their characteristics. [9]

Another substantial influence regarding the solving of DAEs and ODEs is the formulation. This formulates the equations from the model and chooses which variables to solve first. In ADAMS there are three formulations: *I3*, *SI2*, *SI1*. These are listed in Fig 15. [9]

3.7.3 Optimization

Optimization is a sophisticated tool used to improve the design or model and to shorten the design cycle. Usually, designs have certain goals or design objectives which indicate how well the model performs in its task. Such a goal might be to minimize forces affecting to a part during a predefined movement or action. The design process involves choosing the model parameters correctly so that the design fulfills all goals and satisfies possible restrictions. Optimization is meant to analytically reach the theoretically best solution. [9]

The model parameters are expressed as *Design Variables*. These are simple numerical or boolean variables which are used to define the model and its geometry. Design variables may be used to define part's length or velocity, for example. Design goals are called *Design Objectives*. The objective is a user defined numerically calculated value what the user wishes the outcome of the simulation to be, such as forces affecting a part or its weight cannot be above a certain point or the part's final position after a simulation must be a certain value. Restrictions are referred to as *Design constraints*. These constraints are user defined to limit the model in some way, for example, due to manufacturing restraints or maximum dimensions. [9]

The integrator:	Has the following characteristics:
GSTIFF	<ul style="list-style-type: none"> • Uses backward differentiation formulas. • Uses fixed coefficients for prediction and correction. • Is the default method.
WSTIFF	<ul style="list-style-type: none"> • Uses backward differentiation formulas. • Uses variable coefficients for prediction and correction.
Constant BDF	<ul style="list-style-type: none"> • Uses backward differentiation formulas. • Uses fixed coefficients for prediction and correction. • The maximum allowed step size controls the integrator accuracy. • Does not calculate local integration error at each step.
ABAM	<ul style="list-style-type: none"> • Uses coordinate partitioning to reduce the full set of differential and algebraic equations (DAEs) to a smaller set of ordinary differential equations (ODEs). • Uses Adams-Bashforth for prediction; uses Adams-Moulton for correction.
RKF45	<ul style="list-style-type: none"> • Single-step method. • Primarily designed to solve non-stiff and mildly stiff differential equations when derivative evaluations are not expensive. • Not highly accurate.
HHT	<ul style="list-style-type: none"> • Adams/Solver (C++) only • Expected to result in a smaller number of Jacobian evaluations. • Unlike BDF-type formulas, it behaves like a low pass filter; it cuts high frequency spurious oscillations, while accurately preserving low frequency oscillations. • The cutoff frequency can be controlled by adjusting the option Alpha; the smaller value, the lower the cutoff threshold. • Stable at small value of the integration step size.
Newmark	<ul style="list-style-type: none"> • Adams/Solver (C++) only • Behavior similar to HHT. • Lower order. • Has two control parameters.

Figure 14: Comparison of Integrators. Courtesy of MSC Corporation [9]

In Fig 16 a simple optimization process is shown. First, it is often a good idea approach to define the design variables and build the model based on them. Then, after the model is ready, possible design constraints are specified. This step is optional. Finally, design objectives must be chosen. Now an optimization can be run.

The optimization starts running a single simulation. Based on the simulation and its outcome the design objective is calculated. If the objective is reached the optimization is complete. If not, the design variables are adjusted based on the selected algorithm. After adjusting the simulation is ran again and results checked. This loop continues as long as the user has specified in the optimizer settings.

The equation formulation:	Has the following characteristics:
I3	<ul style="list-style-type: none"> • Ensures that the solution satisfies all constraints. • Does not ensure that the the velocities and accelerations calculated satisfy all first- and second-time derivatives. • Monitors integration error only in system displacements, not in velocities. • Is fast. • The Jacobian matrix can become ill-conditioned at small step sizes.
SI2	<ul style="list-style-type: none"> • Takes into account constraint derivatives when solving for equations of motion. This process enables the GSTIFF integrator to monitor the integration error of velocity variables, and, therefore, renders highly accurate simulations. • Jacobian matrix remains stable at small step sizes, which in turn increases the stability and robustness of the corrector at small step sizes.
SI1	<ul style="list-style-type: none"> • Takes into account constraint derivatives when solving for equations of motion. • Monitors the integration error on the impulse of the Lagrange Multipliers in the system. These additional safeguards enable the integrators to monitor the integrator error in velocity variables and the impulse of the Lagrange Multipliers. • Very accurate. • Jacobian matrix remains stable at small step sizes, which in turn increases the stability and robustness of the corrector at small step sizes.

Figure 15: Equation formulation comparison. Courtesy of MSC Corporation [9]

Some commonly used settings for optimization are presented in Table 2. There are two different optimization algorithms supplied with ADAMS and they are: [9] [20]

- OPTDES-GRG (Generalized Reduced Gradient)
- OPTDES-SQP (Sequential Quadratic Programming)

In addition to these algorithms, more can be purchased from 3rd parties. It is also possible to create your own algorithms.

MSC ADAMS also offers design evaluation tools similar to the optimization tool. *A Design Study* helps to investigate the influence of a single particular design variable to the performance of the whole model. During a design study ADAMS varies the value of this variable according to a predefined *range* and simulates & records the results. [9]

A Design Experiment is similar to a design study but it analyses the effect multiple variables at once. It is a useful tool to understand how different variables interact and how it affects the model. [9]

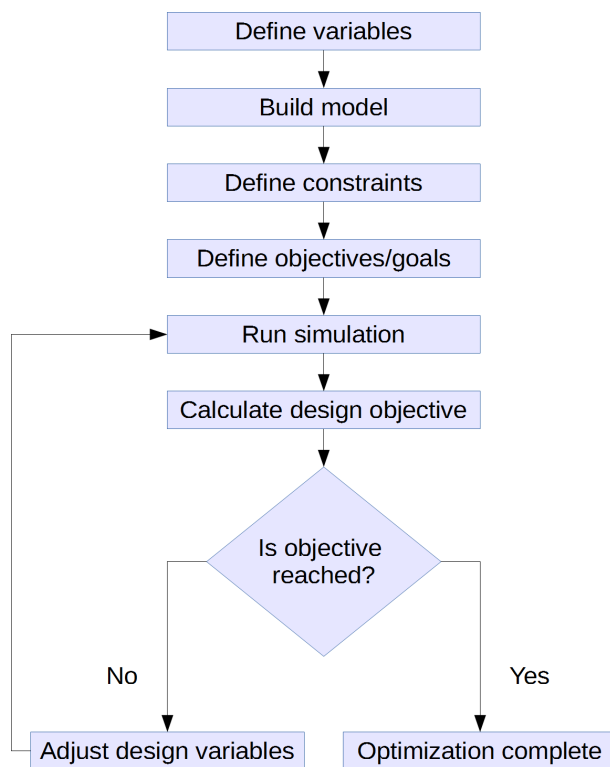


Figure 16: A simple representation of an optimization process

Table 2: A list of measures in the model [9]

Setting	Description
Algorithm	Specifies the mathematical algorithm used to perform the optimization
Convergence Tolerance	The limit below which subsequent differences of the objective must fall before an optimization is considered successful
Maximum Iterations	How many iterations the algorithm should take before it admits failure
Rescale Iterations	The number of iterations after which the design variable values are rescaled
Differencing Technique	Controls how the optimizer computes gradients for the design functions
Differencing Increment	Specifies the size of increment to use when performing finite differencing to compute gradients

4 Analysis

Two analyses are done in this thesis: the first is designing and dimensioning a lifting aid for a hinged main cover with a gas spring. The second one is a vibrational analysis of a dispensing robot unit and its needle. These two problems were chosen to demonstrate the abilities of MSC ADAMS from general designing aid point of view to a frequency and amplitude analysis. The problems were selected with TFS's R & D personnel to reflect real life situations.

4.1 Problem 1: Lifting aid for main cover

The first problem is a classical mechanism analysis including dimensioning and choosing a suitable spring to lighten the load to open a main cover of a laboratory instrument.

4.1.1 Background

In Thermo Fisher Scientific Oy a subdivision called Analyzers & Automation design and manufacture modular automatic laboratory sample transfer and handling systems. These systems are comprised of multiple different modules each performing a specific task. Together they make up a fully automatic conveyor belt style sample handling assemblies. For one such module, a new main cover has been designed but it is too heavy to lift and thus a lifting aid must be added the design. This is the focus of this section.

In Fig 17 the module is shown simplified with the main cover visible in blue colour. The outer dimensions of the module are approximately 1400 x 800 x 1400 mm (W x D x H). The main cover is transparent allowing the user to see inside. The cover is opened during maintenance and in error situations where good reach inside the module is important. The design requirements for the main cover are:

1. The gas spring must be able to be installed fully extended
2. The cover must be able to be lifted with a force of few dozen newtons
3. The cover should stay open by itself

The first item in the list above is related to the assembly of the module. It is required that the gas spring can be installed fully extended as it is exceedingly impractical to compress the spring inside the instrument.

The cover also has to be easy to lift with a force of only few dozen newtons equalling few kilograms. The module is used by different laboratory personnel with various lifting capabilities and therefore the mechanism must be light. Additionally, the cover should stay open by itself in the upright position to ensure safe operation inside the module for the user. In the case this cannot be achieved with the gas spring, it can also be accomplished with an additional mechanism incorporated into the design of the hinges.

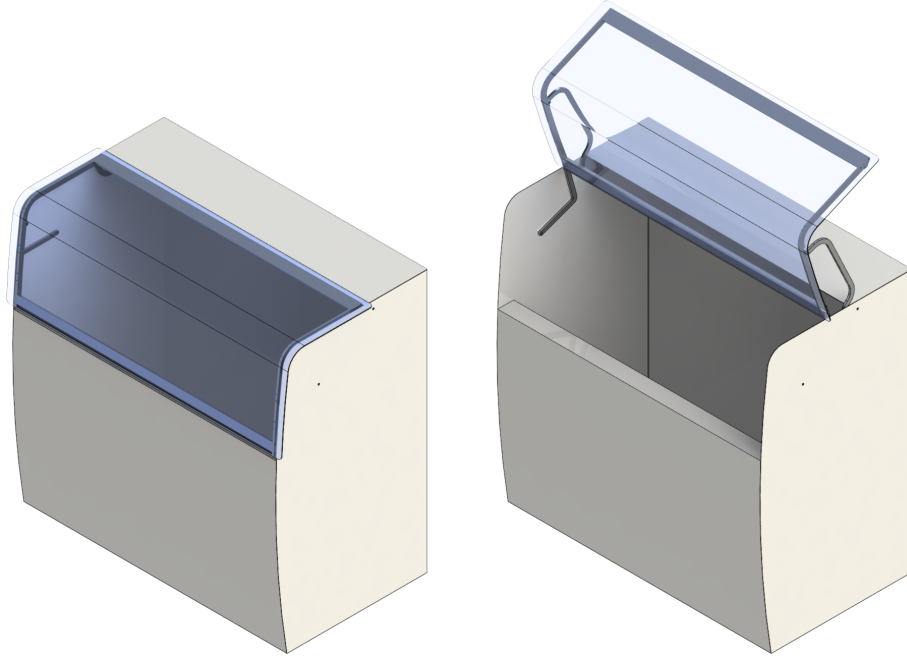


Figure 17: The module with its main cover closed and open

4.1.2 Model

An exploded assembly picture of the main cover is shown in Fig 18. Part names are accordingly listed in Table 4. All other parts are aluminium except the window which is polycarbonate.

The cover mechanism can be simplified to a four bar linkage as shown in Fig 19. The dimensions of the linkage are given in the Table 3. In the open position, the lower link a rotates 160 degrees clockwise while the upper link c approximately 74 degrees.

Table 3: Dimensions of the four bar linkage according to Fig 19

Constant	Name	Length
a	Link lower	299.1 mm
b	Main cover	346.4 mm
c	Link upper	475.2 mm
d	Ground	360.6 mm

The main cover, b in Fig 19, consists of five parts: left and right frame, lower and upper rail and the window itself. These parts are fixed together to simplify the calculations. The masses of the parts in the model are listed in table

A typical gas spring consists of five components: a piston rod, guide, seal, piston and tube. In Fig 20 a gas spring is shown. The tube is filled with an inert gas at high pressure, typically nitrogen. This pressure is equal in both sides of the tube.

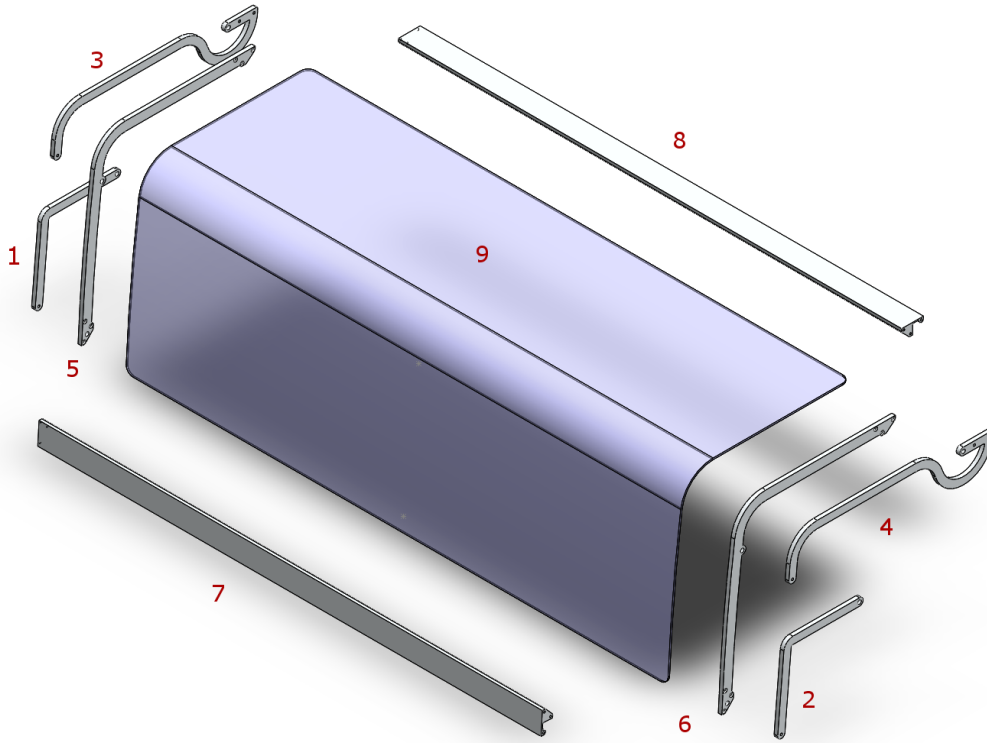


Figure 18: The main cover model assembly exploded

Table 4: Weights of the components in the mechanism according to 3D models. Numbering in relation to Fig 18.

Number	Name	Weight
1	Link lower left	0.22 kg
2	Link lower right	0.22 kg
3	Link upper left	0.41 kg
4	Link upper right	0.41 kg
5	Frame left	0.57 kg
6	Frame right	0.57 kg
7	Rail lower	3.23 kg
8	Rail upper	3.23 kg
9	Window	6.86 kg
	Combined	15.72 kg

The pressure exerts a force on the piston that is greater in the right side in Fig 20. This produces a pushing force at the end of the piston rod. A guide is installed at the end of the tube to keep the rod moving linearly. A seal keeps the pressurized gas inside the tube. Mounting points are provided at each end of the gas spring. [18]

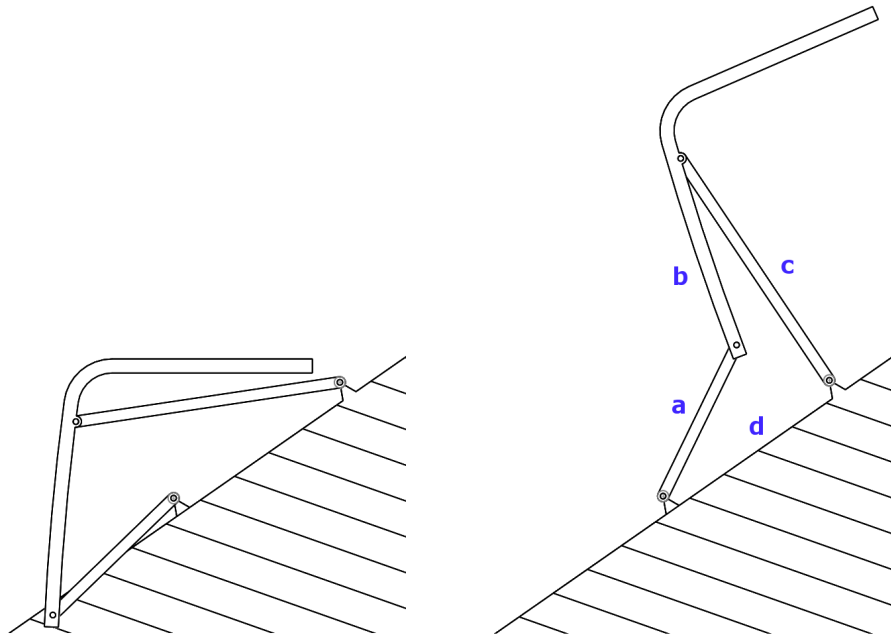


Figure 19: The cover mechanism closed and open as a simplified four bar linkage.

Typical applications of gas springs include lifting and unloading heavy objects, such as window frames, robot actuators and doors. They have dampening characteristics which is useful to control the movement. [18]

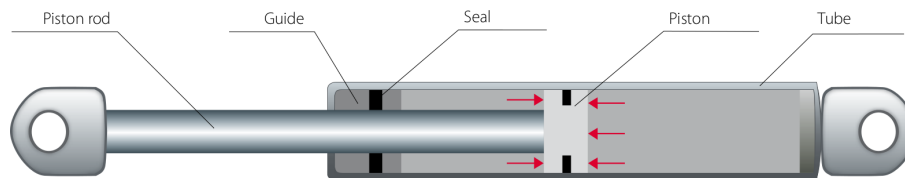


Figure 20: Gas spring, courtesy of Lesjofors AB [18]

Unlike traditional coil springs, gas springs produce force throughout the stroke. Typically, the manufacturer gives gas spring's force when the spring is completely extended. As the spring is compressed, the force increases due to progression and the gas inside the spring compressing. In Fig 21 is shown a typical stroke-force-graph. When stroke is at 100 %, i.e. the spring fully extended, the force is equally at 100 %. As the spring is compressed and the stroke decreases, the force increases. At stroke 0 % the force is 120 % indicating a twenty percent increase. The spring is thus said to have a progression of 20 %.

The model used in ADAMS is shown in Fig 22. The spring is constructed of two parts to aid visualize its movement, a cylinder and a shaft. Additionally in the figure is shown three construction points, PNT_1 , PNT_2 and PNT_3 , as well as two design variables, DV_spring_length and DV_spring_stroke . The points are

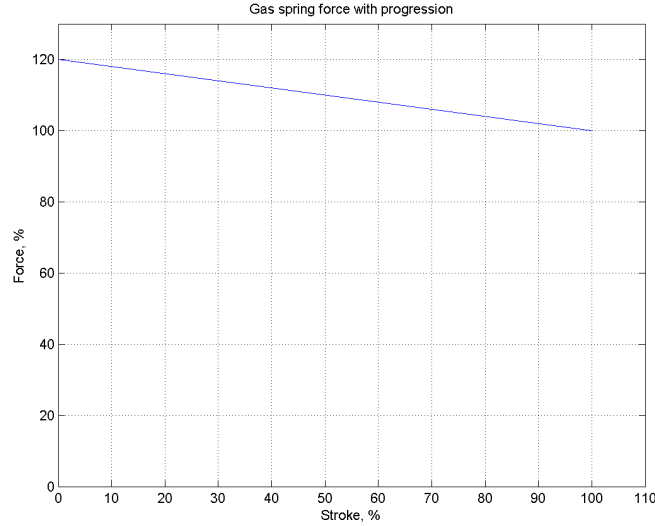


Figure 21: Gas spring force with progression

utilized to measure spring length and stroke. The variables were created to enable easy changing of the values during testing.

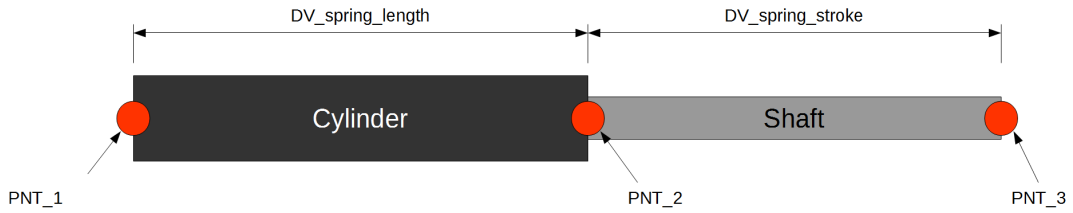


Figure 22: Gas spring model in ADAMS

Besides to the two variables above, two more were needed: $DV_spring_preload$ and $DV_spring_progression$. The preload variable determines the force the spring exerts at fully extended state. The progression variable declares the amount of force increase as the spring compresses according to Fig 21.

The equation to calculate the force of the spring is:

$$F_s = Preload \left(1 + \frac{Progression}{100} \cdot \frac{-DM(PNT_1, PNT_3) + Stroke + Length}{Stroke} \right) \quad (4.1)$$

where DM is ADAMS function to calculate the distance of two points. First, the negative distance of the end points of Fig 22 is determined. Then, the stroke and length are added and the sum divided with the stroke of the spring. As a result, a number from 0 to 1 is obtained. This is multiplied with the amount of progression and summed with the preload.

The model was built according to the "Crawl, walk, run" philosophy introduced in Section 2.3. Topology is shown in Fig 23 with appropriate measures. A full list

of measures is shown in Table 5. The topology is a *closed loop* type described in Section 3.1. Only the left side links and hinges were modelled since the other side's links would have overdefined the system.

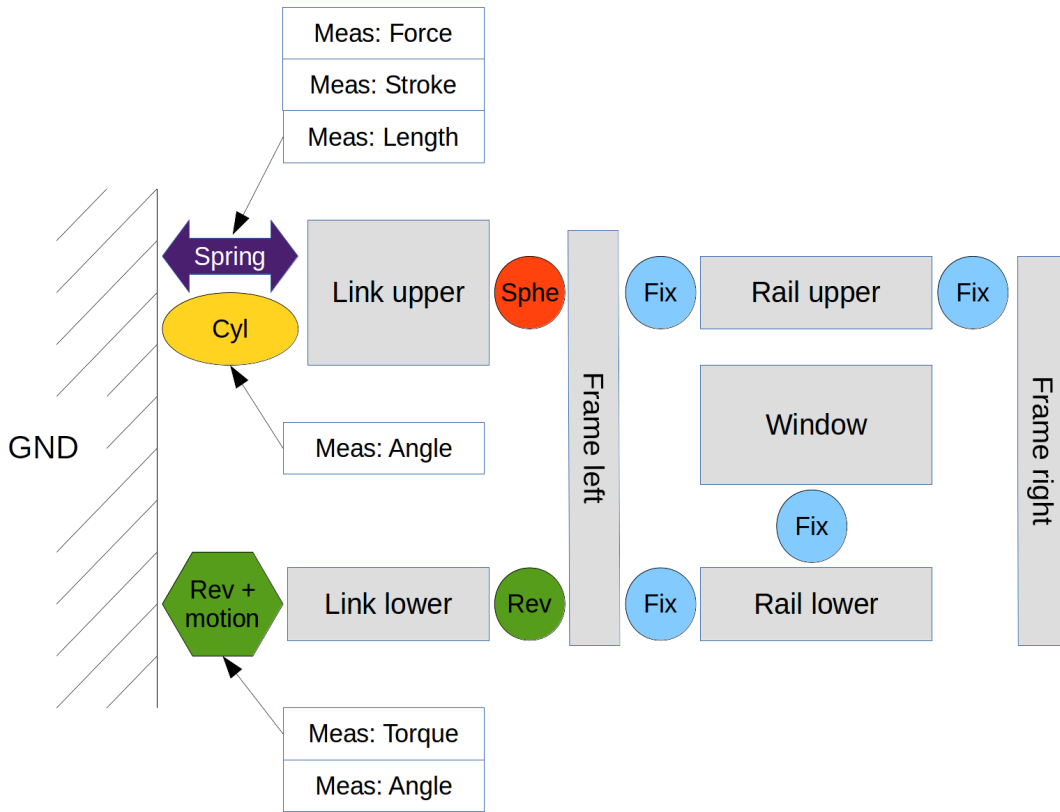


Figure 23: Model topology

The lower link in Figs 18 and 23 is actuating the mechanism during the simulations. There is a motion in the revolution joint between the link and ground. There was another revolution joint connecting the lower link to the left frame. The upper link was connected with a cylindrical joint to ground and with a spherical joint to the left frame instead of revolute joints to avoid redundant constraints. The other parts of the assembly were fixed to the left frame.

The *Gruebler count* including the motion is

$$\begin{aligned}
 m &= 6 \cdot (n_b - 1) - n_c \\
 &= 6 \cdot (8 - 1) - (4 \cdot 6 + 2 \cdot 5 + 4 + 3 + 1) \\
 &= 6 \cdot 7 - (24 + 10 + 4 + 3 + 1) \\
 &= 42 - 42 \\
 &= 0
 \end{aligned}$$

which means there are no degrees of freedom and the movement of the mechanism is completely determined. This is as intended and is required to perform a static simulation.

Table 5: A list of measures in the model

Type	Name	Description
Torque	Motion_open_torque	The torque in <i>Motion_open</i> needed to open and close the main cover around the joint between <i>Link_lower</i> and <i>GND</i> , see Fig 23
Force	Motion_open_force	The force needed to open and close the main cover at the end of <i>Link_lower</i> , calculated from Motion_open_torque
Angle	Link_upper_angle	The angle of the cylindrical joint between <i>Link_upper</i> and <i>GND</i>
Angle	Link_lower_angle	The angle of the revolution joint between <i>Link_lower</i> and <i>GND</i>
Length	Spring_length	The length of the gas spring, see Fig 22
Length	Spring_stroke	The stroke of the gas spring, see Fig 22
Force	Spring_force	The force of the gas spring

4.1.3 Analysis

The analysis was begun by determining the motion to open and close the main cover. The cover is closed when the lower link is horizontal and open when it has rotated 160 degrees. The motion to rotate the link was determined with *STEP* functions as

$$Displacement = STEP(time, 0, 0, 10, 160d) - STEP(time, 10, 0, 20, 160d) \quad (4.2)$$

The cover now opens and closes 160 degrees in 10 seconds. The whole movement takes 20 seconds which was also the simulation time used. Since no inertial loads had to be considered, the simulations were ran as static. The duration of the movement does not affect the results since a static simulation was run and it does not take in to account the inertial properties of the components. The resulting angles are shown in Fig 24. Contrary to the figure, the links rotate to the same direction. The lower link rotates from zero to 160 deg to zero as intended. The lower link rotates from zero to approximately 74 degrees, then back to 65 degrees at top dead center.

The force required to open and close the main cover is shown in Fig 25. Note that the x-axis is in lower link angle. The maximum force required is about 125 newtons. This is the value the gas spring aims to diminish. At the end of the movement the mechanism locks in to place as indicated in the graph when the line approaches and reaches zero and then changes sign.

Now, when the movement of the upper link has been calculated, the spring can be preliminary placed for testing purposes. In Fig 26 a sketch is shown to indicate the placement of the spring. In the figure there is the end of upper link that connects to the ground via a cylindrical joint. The link has a 'J' shaped end to accommodate

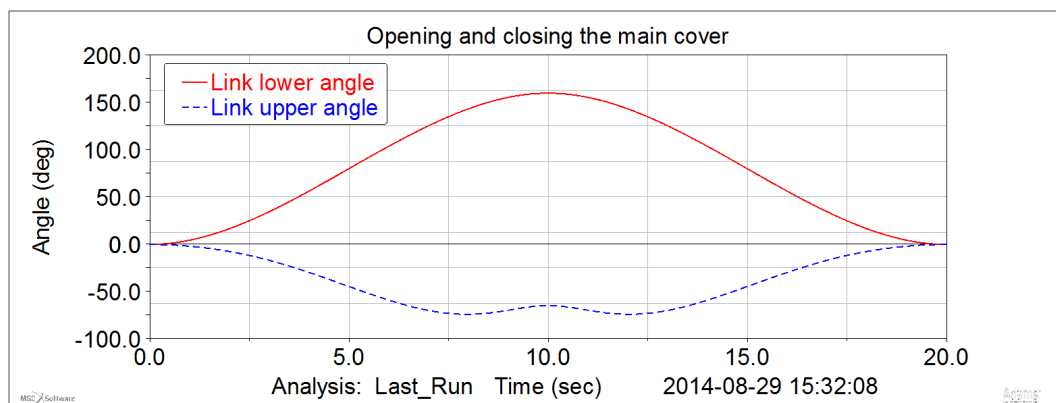


Figure 24: The angles of lower and upper links during opening and closing motion

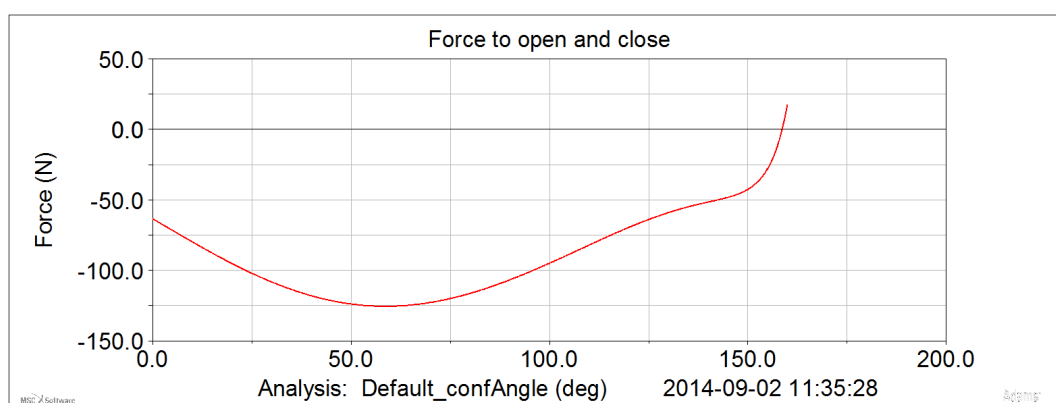


Figure 25: The required force to open cover as a function of lower link angle

the module's roof when fully opened. The spring can be mounted on to a circle with diameter of 140 mm shown as dotted line. At a distance of 285 mm from the link's joint there is a beam in the frame of the instrument to which the spring may be securely attached. A vertical line is drawn to resemble this beam. The other end of the spring should therefore lie on this line.

Next, a simple test was conducted to get a rough view of what could be the optimal point to attach the spring to the upper link. The end of the spring that connects to the module frame – ground in ADAMS – was positioned to the aforementioned line and horizontally to the link's joint. Then, the spring's other end was attached to the circle with the diameter of 140 mm. The figure shows the two positions, compressed and extended, of the gas spring. The line indicating the compressed spring's length starts from this circle at a 48 degree angle from horizontal line. Accordingly, the extended spring starts from a 74 degree angle from the compressed spring's starting point as determined from Fig 24.

For the test the 48 degree angle – later referred to as β or β , was altered, i.e. the angle of the attaching point of the spring in its compressed state. The tested

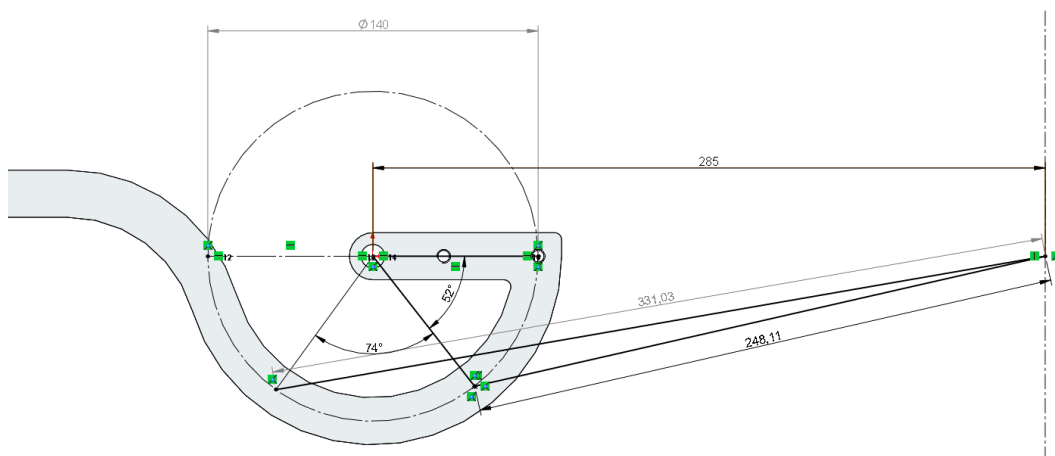


Figure 26: The sketch of the placement of the spring on the upper link

angles were: 0, 26, 52, 78 and 104. Corresponding angles of the spring in its extended state were: 74, 100, 126, 154 and 180. At the minimum angle of 0/74 degrees the spring is horizontal at the beginning of the movement and at the maximum angle of 104/180 the spring is again horizontal but at the end of the movement.

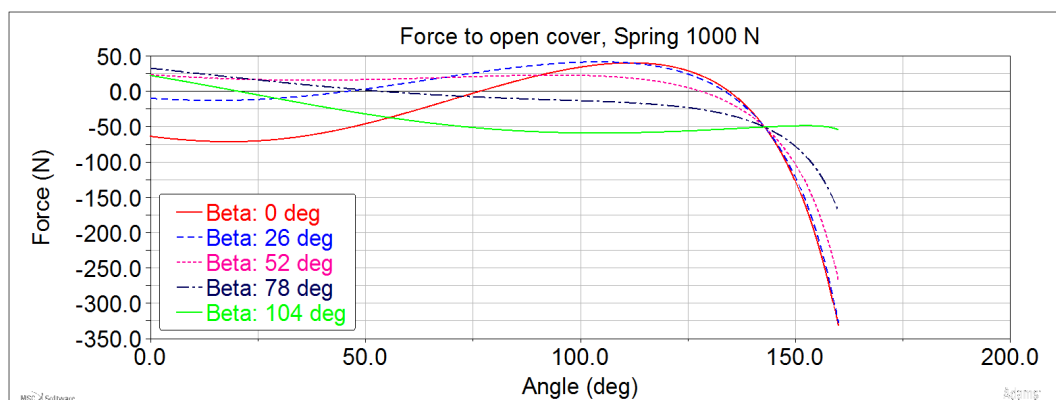


Figure 27: Results of altering the spring attaching point on upper link

The results are shown in Fig 27. Only half of the full movement of the cover was measured, that is, the cover is only opened but not closed. The force on vertical axis is the force required to open the cover according to Table 5. The gas spring was set as constant at 1000 N which proved to be a good guess.

From the figure we can see that with a small β the opening movement requires a great force at beginning but lightens when approaching approx 120 degree angle. Likewise with a large β the cover is first light to open but becomes heavier as the angle increases. The best results are obtained with $\beta = 52$ deg when the force is most constant throughout the movement.

Next, a more intricate analysis of the angle β was performed. A new set of results

is shown in Fig 28. The best compromise is once again achieved with $\beta = 52$ deg. Instead, an angle of $\beta = 48$ deg was chosen to accommodate the spring progression later.

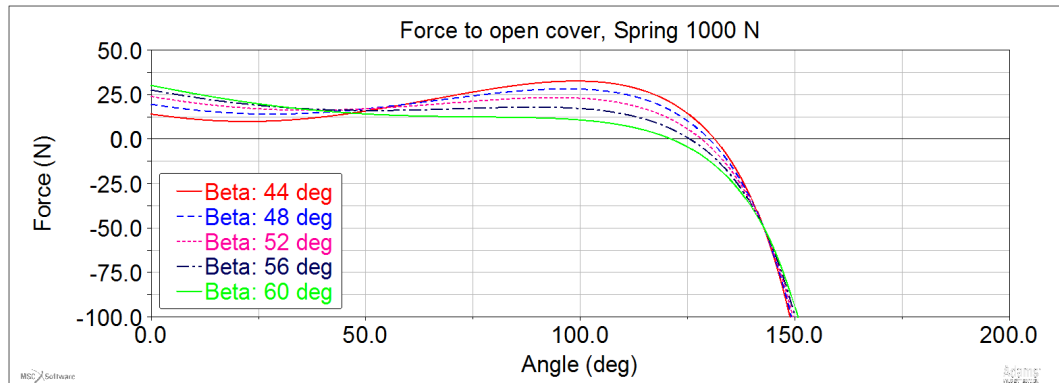


Figure 28: Another set of results of the spring attaching point

A sketch of spring placement with angle $\beta = 48$ deg is shown in Fig 29. The length of the spring extended is 327.52 mm which is the ideal length. Additionally, the angle between the compressed spring line and the line defining the angle β is shown; it is 119.68 degrees. This angle – defined as α – must kept constant so that the results obtained earlier remain valid.

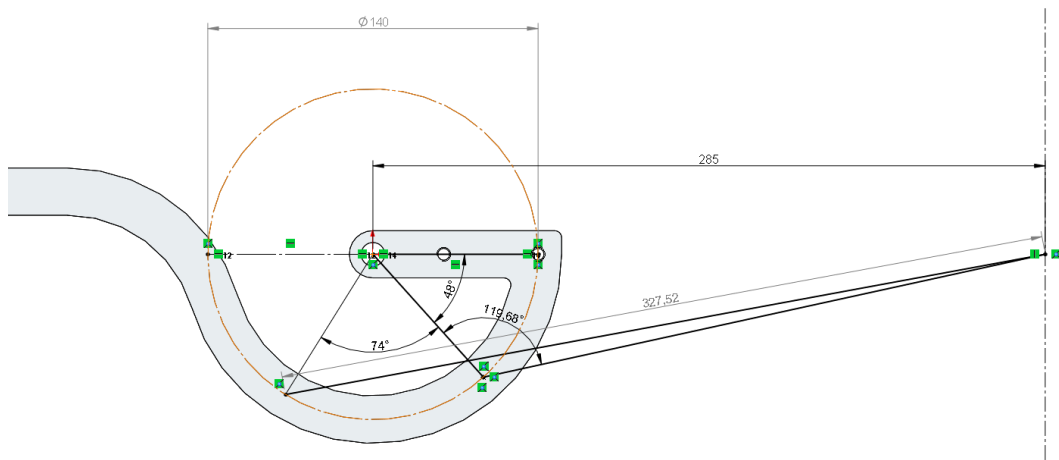


Figure 29: The sketch of the placement of the spring with $\beta = 48$ deg

Now, a suitable gas spring can be found. The ideal length from Fig 29 is 327.52 mm and the force the spring produces must be around 1000 newtons. If two gas springs are installed on both sides of the main cover, half of 1000 N is sufficient. According to Fig 24, the stroke of the gas spring must be at least [36]

$$S = 2 \cdot \frac{140 \text{ mm}}{2} \cdot \sin\left(\frac{74}{2 \cdot 360}\right) \approx 84.3 \text{ mm} \quad (4.3)$$

An applicable spring was found from HAHN Gasfedern. The model is G 10-28 indicating a piston rod diameter of 10 mm and cylinder diameter 28 mm. The specification sheet can be found in Appendix 1. Extension force is rated from 100 to 1200 newtons with a progression of 20 %. Choosing a stroke of 120 mm, the overall length of the spring extended is 310 mm without fittings. With fittings *AU 19* the overall length increases by 38 mm to 348 mm. The geometry with this length is shown in Fig 30. In the figure can be seen that the angle of 119.68 degrees is kept constant but there is no reference to β .

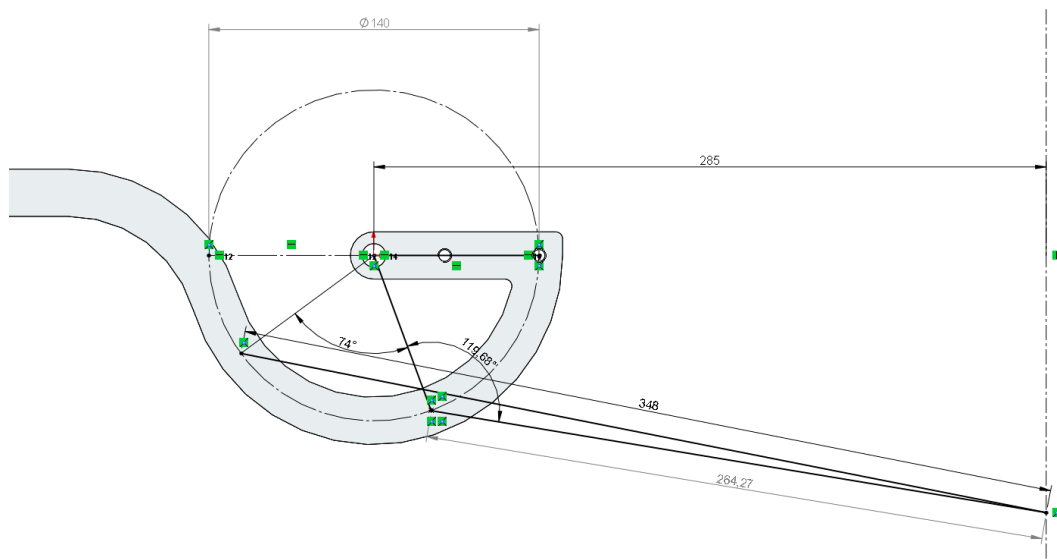


Figure 30: The sketch of the placement of the spring with length 348 mm

Next, different spring forces were tested. The results can be seen in Fig 31. From the graph can be seen that the best solution is somewhere between 750 N and 800 N. A further analysis was performed of this range and the results can be seen in Fig 32. From this graph, the force 780 N seems most suitable and was therefore chosen.

The ADAMS model with the cover closed and open is shown in Fig 33. The gas spring can be seen in the lower left corner of the assembly. Measures for the final configuration are shown in Fig 35. In the 'Force of gas spring' graph there is a new component labelled 'Spring force eff'. This is the effective force the gas spring generates where the angle of the spring relative to the joint is taken into account. The function measure is

$$Spring_force_eff = SIN(INCANG(PNT_1, PNT_3, JOINT)) \cdot Spring_force \quad (4.4)$$

The force graph from the figure is shown again in Fig 34 as a function of lower link angle. From this graph it can be seen, that the cover opens easily to about 120 degrees. After this point, the opening motion starts to compress the spring as seen previously in Fig 24.

Comparing to the requirement list in Section 4.1.1, following observations can be made:

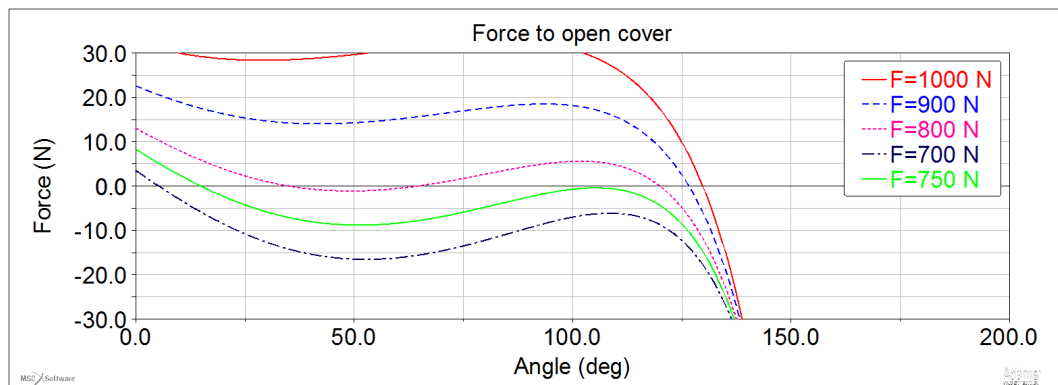


Figure 31: The force needed to open the main cover as a function of different spring forces between 700 N and 1000 N



Figure 32: The force needed to open the main cover as a function of different spring forces between 700 N and 800 N

1. The gas spring design length according to manufacturer is 348 mm. From Fig 35 it can be seen that this dimension is reached when the spring is fully extended
2. The force required to open the cover is less than 20 newtons from 0 to 130 degrees in lower link angles
3. An adjustable friction element is required for the cover to stay open by itself

All the requirements are met and the design should function as intended. The gas springs were successfully ordered and received from the manufacturer. According to the simulations the force required to open the cover is under 20 newtons which equals to two kilograms fulfilling the requirement 2.



Figure 33: The ADAMS model with the gas spring, closed and open configurations

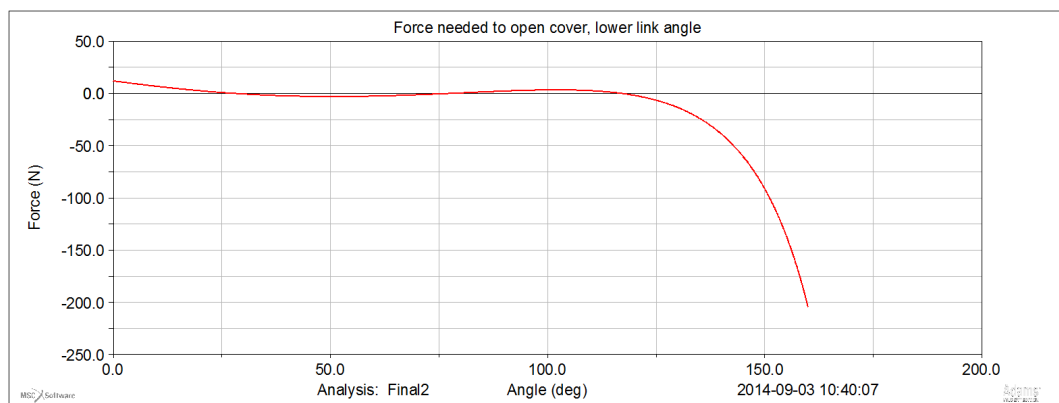


Figure 34: Graph of the force needed to open the cover as a function of lower link angles

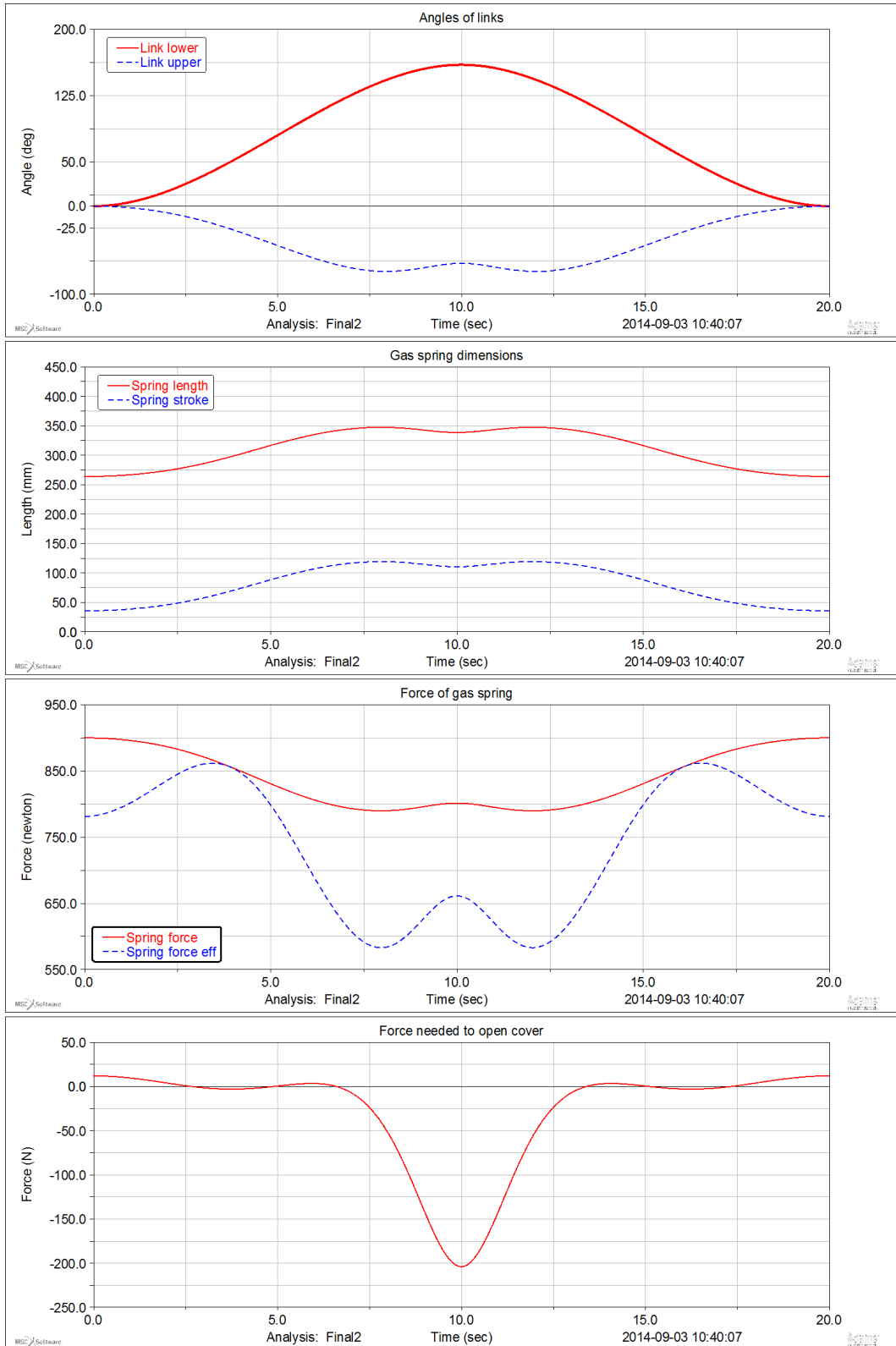


Figure 35: Measures from the final configuration

4.1.4 Results Verification

A real life prototype of the main cover was constructed to check the functionality of the design and whether the design goals listed in Section 4.1.1 were reached. The prototype was built according to description in Section 4.1.2 and the final design obtained from results in Section 4.1.3. A photograph of the gas spring in the prototype is shown in Fig 36. All the parts were attached to one another with screws with built in plain bearing surfaces. The bearing itself was installed between the screw and the part it was being attached to. This eliminates the need for any additional joints such as hinges etc. In the Figure, the screws can be seen in black and the bearing as white beneath them.

Functionality of the main cover was evaluated by hand as it was the most straightforward method and defines the user experience in the end. As intended, the cover proved very light to open and the force required stayed quite constant throughout the manoeuvre. The cover was somewhat more easier to open than close, this may be because of the additional frictions in the joints and screws. It may be necessary to select springs with a slightly lower force for the end product.

A positive surprise was the fact that the gas spring itself has enough internal damping so no external friction element was needed for the cover to effortlessly stay open by itself at any angle. The prototype had perhaps a bit too much damping and it may be worth to try out springs with a bit lower damping. The manufacturer of the gas spring can build the spring with different damping coefficients if there is need to fine tune it.

The installation of the spring was very straightforward. The spring could be quickly put in and without any complicated tools since the attaching points of the gas spring were chosen so that it could be installed fully extended. The end connectors of the spring were also ideal for this purpose and their holes for screws very conveniently sized.

Overall the designer of the cover and mechanism was very satisfied with the result. The prototype was not very sturdy but otherwise it functioned superbly and demonstrated well the idea of virtual prototyping. There were already some minor improvement plans requiring further simulations, such as trying to re-enable the lost locking function as shown earlier in Fig 19.

4.1.5 Conclusions

In this problem, a lifting aid for an instrument's main cover was dimensioned and suitable gas spring chosen. The mechanism and design of the main cover was already given but the attaching points of the gas springs and the springs themselves were determined with a multibody simulation in MSC ADAMS.

The main cover consists of nine parts but the simulation model was reduced to six components by taking into account duplicated parts on the other side of the cover. In the model, the gas spring was modelled as a custom force function representing the actual behaviour of a gas spring such as progression during extending and compressing. Revolution, cylindrical, spherical and fixed joints were used to



Figure 36: Photograph of the prototype and one of its gas springs

construct the model. There were no flexible parts. The system had zero degrees of freedom and thus the simulations were ran as static to ignore the parts' inertias.

In the analysis, first the geometry and the attaching points of the gas spring were chosen. Typically, few simulations were ran at a time varying certain parameters and the best result was chosen to the next step. With this semi-empirical method a suitable geometry was found fulfilling the requirements. After this, the desired stroke and force of the gas spring were known and an appropriate gas spring was searched. HAHN Gasfedern model G 10-28 was selected. Final simulations were successfully ran to confirm the selection and the fulfilment of the original requirements.

A fully functioning prototype was built for result verification purposes. The prototype proved to perform very well and just as intended. The cover was light to open and stayed open at all opening angles. Additionally, the gas spring was easy to install to the mechanism. The designer of the cover and mechanism was very satisfied with the results.

The dimensioning and choosing the gas spring for the cover mechanism was considered a success. The overall time used to solve this problem with multibody simulations was about one week. This is relatively small time and price to pay compared to performing the same iteration with actual prototypes. With prototypes, the problem would most probably not be solved with the first creation but would have to be refined in the next ones each spanning few months apart. As the problem was successful solved with a one week simulation, considerable time and effort was saved. The same model and observations may be very easily transferred and taken

into account in different types and shapes of covers requiring even less time than the one week this problem took.

4.2 Problem 2: Vibration analysis of a needle

The second problem involves analysing deflections of a polar coordinate robot and its needle utilizing flexible parts and ADAMS' inbuilt functions.

4.2.1 Background

In some of Thermo Fisher's laboratory analyzers, polar coordinate robots are utilized for liquid handling among other things. A picture of one such robot can be seen in Fig 37. The unit's main parts are: frame, phi and y motors, bearing assembly, arm, needle holder and needle. The two polar coordinates are y for up/down motion and ϕ for rotations about the y axis, each has its own stepper motor. The motors actuate a spline shaft via belts, the shaft being attached to the frame with the bearing assembly. The shaft is connected to the arm at the other end. The needle holder holds the needle in place, as the name suggests. Through the needle the unit dispenses liquids with the actual syringe located elsewhere in the analyzer. The

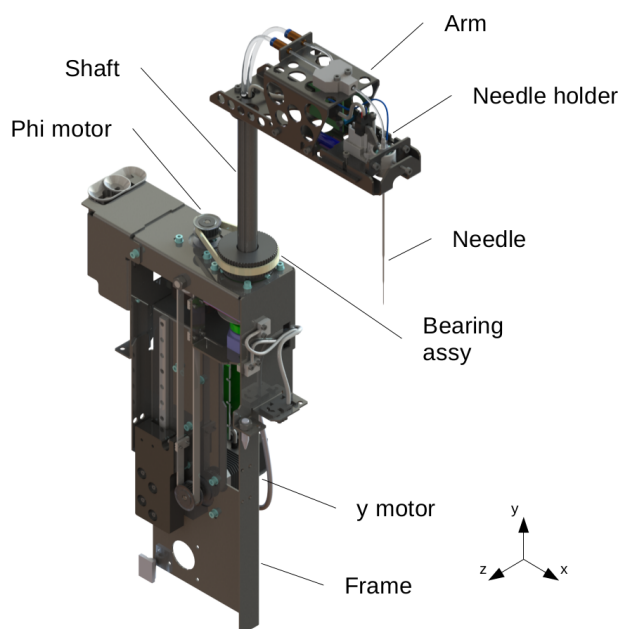


Figure 37: Polar coordinate dispensing robot with main components

shaft in the unit is a hollow spline type, shown in Fig 38. The ball spline functions as a translational joint that is fixed to the unit frame with the bearing assembly allowing rotations about the y axis. Since the tip of the needle hangs freely in the air, vibrations occur during the movement of the unit. This is particularly a problem in the ϕ direction when the movement stops suddenly. The purpose of this analysis is to construct a model resembling the actual unit and measuring how large vibrations occur at the tip of the needle.

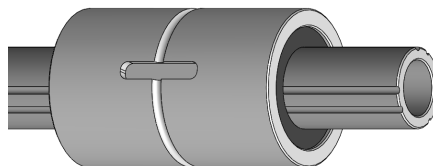


Figure 38: THK LT 16 Hollow ball spline shaft and bearing

4.2.2 Stepper Motor Flexibility

The stepper motor flexibility plays a significant part in the vibrations of the dispensing unit. Although the motor aims to keep as still as possible when movement has stopped, the so called *Holding torque*, it nevertheless has certain spring-like characteristics in it. That is, the motor is prone to oscillate for a short while after any movement has stopped. This is a problem particularly when stopping from high speeds. [15]

This behaviour can be modelled with an additional torsion spring. To do this, first a stiffness has to be determined for the motor. This is done by analysing the oscillation frequency with a high speed camera and tracking software. Software called *Tracker* was chosen since it is open source and academically developed [8]. The high speed camera used shoots 400 frames per second.

In the test the stepper motor was equipped with a heavy disc whose inertial properties are known. The motor was then made stationary with holding current enabled and the disc was lightly rotated from the equilibrium. After the rotation the disc was allowed to freely move, resulting in a oscillating motion. The outcome can be seen in Fig 39.

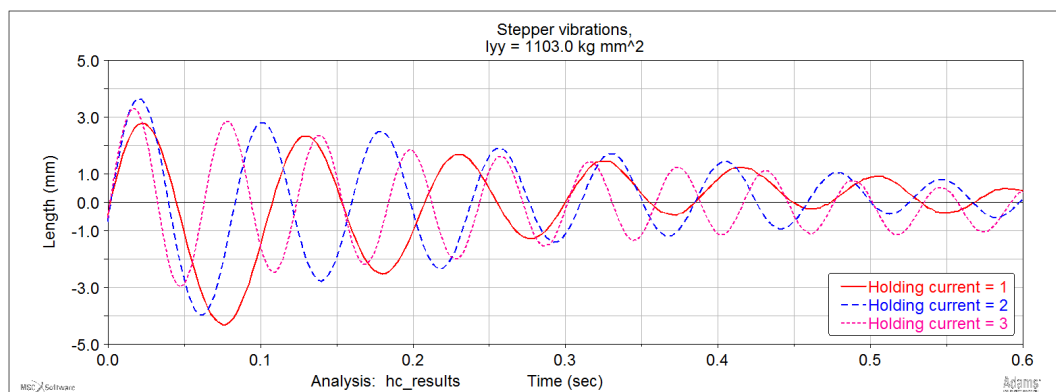


Figure 39: Measured stepper motor vibrations with different holding current settings

In the figure, three graphs are shown. Each graph represents a different holding current setting. The holding current is an electrical current flowing through the stepper when it is stationary, holding it in place and resisting any movement. The

currents are rated from 0 to 3, with 0 = no holding at all, 1 = 25 % current, 2 = 50 % current and 3 = 100 % current from rated maximum for the motor.

To determine the stiffness of the motor at these different settings, the natural frequencies must be known. A FFT transformation was done for all three. The results are in Fig 40.

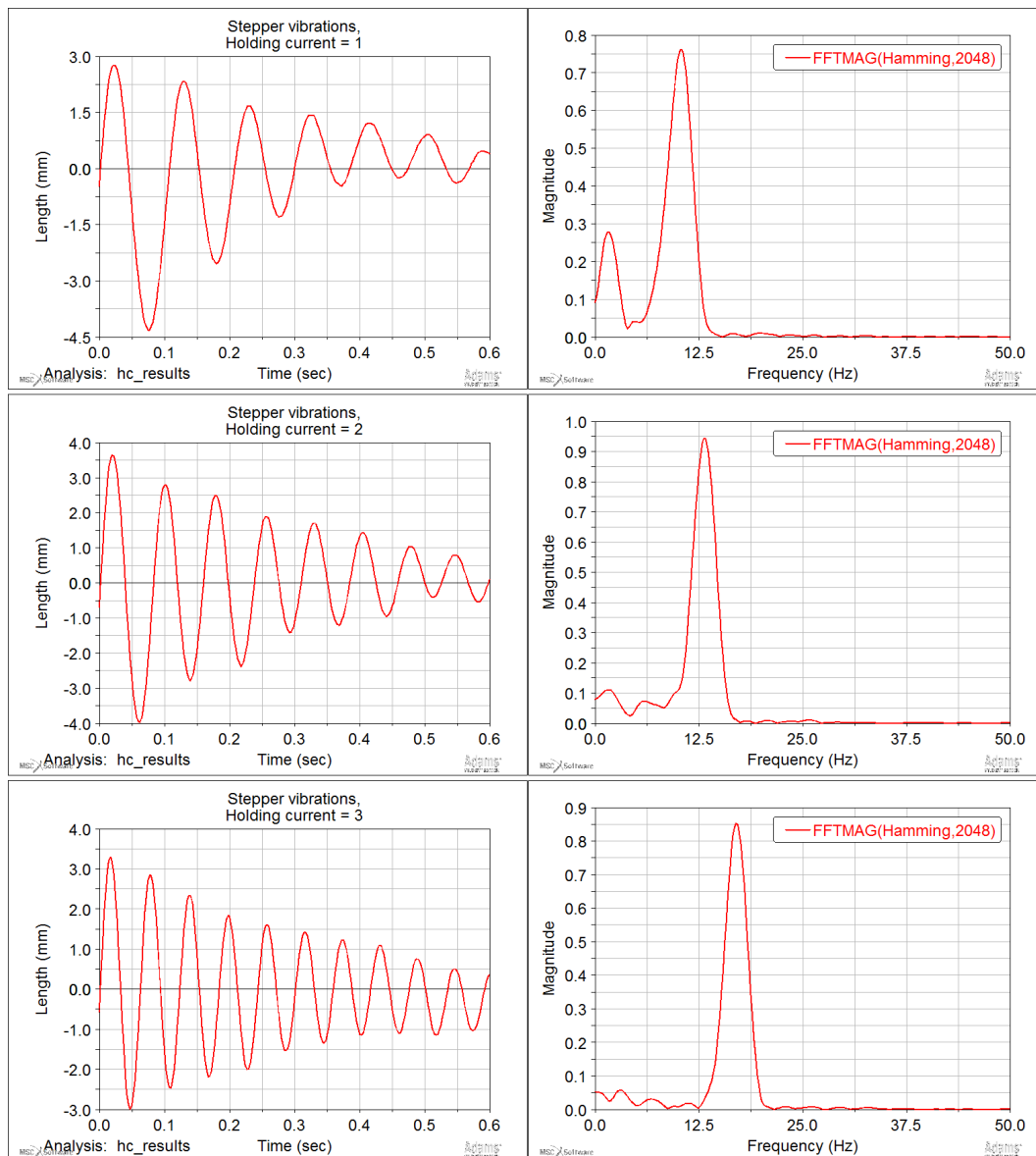


Figure 40: Measured stepper motor vibrations with different holding current settings and FFT graphs

From the figure it can be seen, that with holding current 1 the high frequency peak is 10.9 Hz, with holding current 2 it is 12.5 Hz and with holding current 3 17.2

Hz. From Section 3.6 we have learned that

$$\omega_n = \sqrt{\frac{k}{J}} \quad (4.5)$$

$$\Rightarrow k = J \cdot \omega_n^2 \quad (4.6)$$

Where ω_n is the frequencies above, k is the wanted stiffness and J is the moment of inertia of the disc, which is $J = 1103.0 \text{ kgmm}^2 = 1.103 \cdot 10^{-3} \text{ kgm}^2$, calculated with SolidWorks. We get

$$k_1 = J \cdot \omega_n^2 \quad (4.7)$$

$$= 1.10 \cdot 10^{-3} \text{ kgm}^2 \cdot (10.9 \text{ Hz})^2 \quad (4.8)$$

$$= 0.131 \text{ Nm/1} \cdot \frac{\pi}{180 \text{ deg}} \quad (4.9)$$

$$= 2.286 \text{ Nmm/deg} \quad (4.10)$$

The results are listed in Table 6.

Table 6: List of stepper motor vibration results

Holding current	ω_n	k_n
1	10.9 Hz	2.3 Nmm/deg
2	12.5 Hz	3.0 Nmm/deg
3	17.2 Hz	5.7 Nmm/deg

The holding current setting 3 was used when testing the unit in real life. Hence, the value $k = 5.7 \text{ Nmm/deg}$ was set for the torsion spring in the model. Damping was initially set to $c = 1.0 \cdot 10^{-5}$.

Driving ramps

The motor is accelerated to the desired velocity with an *acceleration ramp*. The acceleration is almost always smoothed and rarely linear. A smooth S-shaped ramp resembling a *Step-function* in ADAMS is used to drive the dispensing unit in real life. It is known as an *S-ramp*. A comparison of these three ramps is shown in Fig 41. It is noticeable that the S-ramp has gentler acceleration at the beginning of the motion and at the end. In the middle the S-ramp has higher acceleration for a longer duration than the Step-function. The motivation for this is to have lower acceleration to lower the vibrations when the movement stops.

4.2.3 Constructing the Model

The topology of a full dispensing unit is shown in Fig 42. First, the frame is connected to the analyzer base plate by four M4 bolts. Then, y motor, ϕ motor and bearing assembly are connected to the frame by four M3 bolts each. The motors

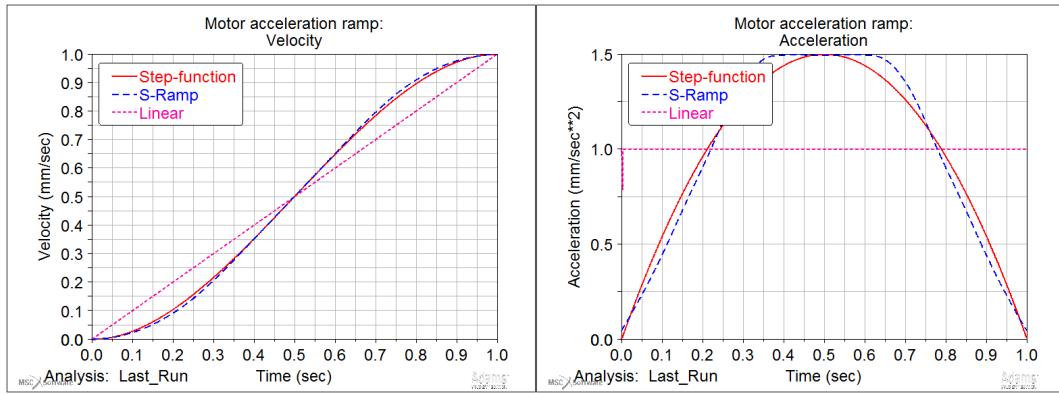


Figure 41: Different motor acceleration ramps: Step-function, S-ramp and linear

are further connected by toothed belts to the spline shaft. The ϕ motor limits the shaft's degrees of freedom by one, rotation around y axis. The y motor limits the movement along the y axis. The bearing unit limits the rest, that is, displacements along and rotation around the x and z axes. Next, the arm is fixed to the shaft with three M3 bolts, and the needle holder to the arm with one M4 bolt. The needle itself is mounted to the holder by a conical contact which a spring holds in place. The point of interest is at the tip of the needle.

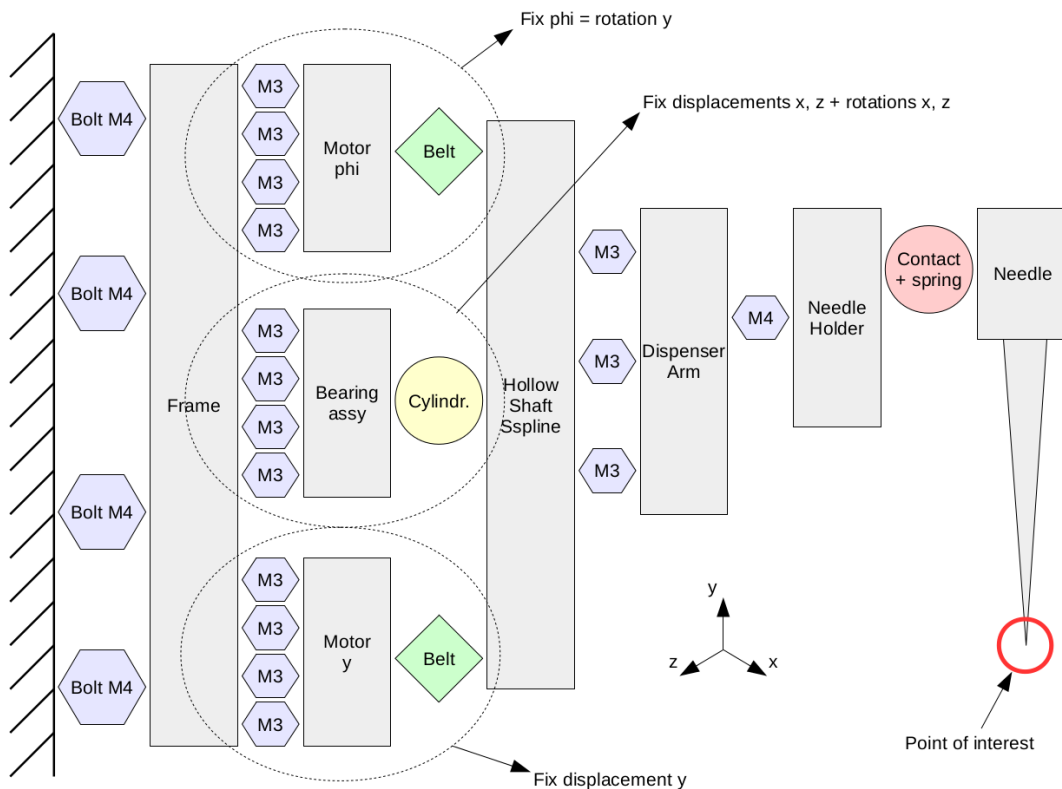


Figure 42: Topology of a dispensing robot

To simplify the multibody model, some elements were left out. For example, the bolts were assumed to be rigid enough to be modelled as fixed joints. The maximum torque the stepper produce is about 0.34 Nm, that is, there may not be torques above this limit in the unit. Therefore the bolts' deformations at 0.34 Nm were deemed to be insignificant and were replaced by fixed joints.

Additionally, a special arm was constructed to replace the complicated arm, needle holder and the conical spring loaded contact to needle itself. In the new arm, the needle attached to it directly with screws. The arm was modelled as a flexible beam consisting of 8 sections using ADAMS' beam tool. Likewise, the needle was also modelled as a flexible beam comprising of six segments. More of this later on.

The final ADAMS model topology can be seen in Fig 43. Overall, there are four flexible parts as described in Table 7. The frame is fixed to the ground. The two motors' pulleys and the upper motor's shaft are attached to the frame via revolution joints. The upper motor's shaft's and the lower motor's joints also have motions in them to move the unit in ϕ and y directions, respectively. The torsion spring between the upper motor and the upper pulley takes into account the flexibility of the stepper motor as described in Section 4.2.2. The lower motor is kept still during the analysis so it is not necessary to model it flexibility.

Table 7: Flexible parts in the model

No	Part	Type
1	Stepper	Torsion spring
2	Upper belt	Torsion spring
3	Arm	Discrete beam
4	Needle	Discrete beam

The lower motor is coupled to the spline shaft transforming the motor's rotation to the shaft's translative movemen via a coupler. The upper motor is connected to the bearing assembly also with a coupler to model the two pulleys gear ratio of $z = 24/60 = 0.4$. The bearing assembly attaches to the spline shaft with a translational joint. Finally, the 8 segment arm beam is fixed to the shaft as well as the 6 segment needle is fixed to the arm.

Belts

The belt flexibility will be taken into account by regarding it as a torsion spring attached in series to the spring modelling the stepper's flexibility. In real life the belt would be positioned between the upper pulley and bearing assembly but the coupler used in ADAMS does not allow this. Hence, the belt must be combined with the only torsion spring in the system. By combining these two spring some accuracy is lost, especially due to the belt not being in the correct position to take into account the upper pulley and its inertia. These small modelling inaccuracies were assumed to be negligible, though.

The belt is type T2.5 steel reinforced toothed timing belt. The manufacturer of the belt lists it to have a specific spring rate of $c_{spec} = 2.45 \text{ N}/10\text{mm} \cdot 10^4$. Note

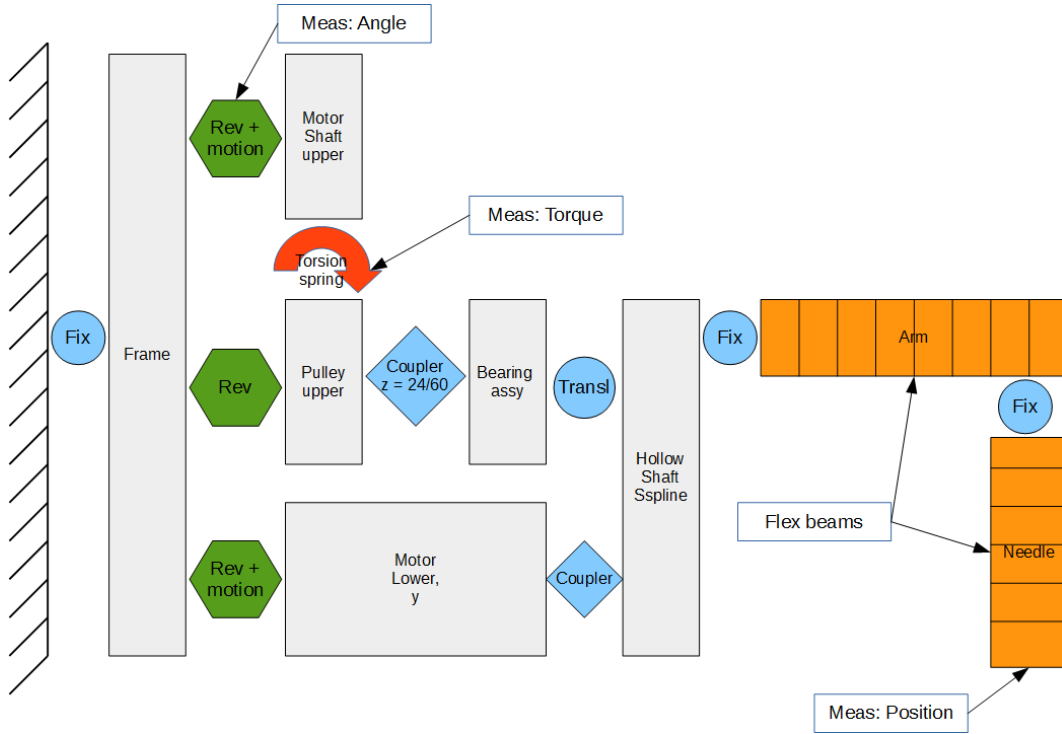


Figure 43: ADAMS topology of the dispensing robot

that this value is given at a belt width of 10 mm [35]. The spring rate of the whole belt can be calculated with the following formula [7]

$$c_{min} = \frac{4 \cdot c_{spec}}{l} \quad (4.11)$$

where l is the length of the belt. In this particular case the belt is 210 mm long and 6 mm wide, we get

$$c_{min} = \frac{6 \text{ mm}}{10 \text{ mm}} \cdot \frac{4 \cdot 2.45 \text{ N} \cdot 10^4}{210 \text{ mm}} = 280 \frac{\text{N}}{\text{mm}} \quad (4.12)$$

Linear stiffness can be converted to torsional with equations

$$F = kx \quad (4.13)$$

$$x = \theta R \quad (4.14)$$

$$M = FR = k\theta R^2 \quad (4.15)$$

$$= \theta c \quad (4.16)$$

$$\Rightarrow c = \frac{k\theta R^2}{\theta} = k\theta R^2 \quad (4.17)$$

And taking in to account the radius of the pulley and converting the stiffness to

degrees

$$c = 280 \frac{N}{mm} \cdot (23.5 \text{ mm})^2. \quad (4.18)$$

$$= 154630 \frac{Nmm}{rad} \cdot \frac{\pi}{180 \text{ deg}} \quad (4.19)$$

$$= 2697.434 \frac{Nmm}{deg} \quad (4.20)$$

$$\approx 2697 \frac{Nmm}{deg} \quad (4.21)$$

Damping ratio was set to $1.0 \cdot 10^{-5}$. There are limitations to this belt model, however. It does not take in to account the stiffness of individual teeth and, therefore, it probably characterizes the belt stiffer than it actually is.

Combining the motor flexibility with belt stiffness yields

$$\frac{1}{k} = \frac{1}{k_1} + \frac{1}{k_2} \quad (4.22)$$

$$= \frac{1}{5.7 \text{ Nmm/deg}} + \frac{1}{2697 \text{ Nmm/deg}} \quad (4.23)$$

$$\Rightarrow k = 5.688 \text{ Nmm/deg} \quad (4.24)$$

It would appear the effect of the belt is negligible compared to the stepper's stiffness.

Excluding the arm and the needle the *Gruebler count* is

$$m = 6 \cdot (n_b - 1) - n_c \quad (4.25)$$

$$= 6 \cdot (5 - 1) - 6 - 5 - 5 - 1 - 1 - 4 - 1 \quad (4.26)$$

$$= 1 \quad (4.27)$$

That is, there is only one DOF free and that is the belt connecting the upper motor and bearing assembly together.

Arm beam

The arm and the needle were constructed using ADAMS' discrete flexible link tool. The arm has eight segments and the needle six. Dimensions of the arm are shown in Fig 44. It is essentially a beam with a rectangular cross section. For the sake of simplicity, fixing features such as bolt holes were left out from this picture. Material is Polyoxymethylene (POM) with elastic modulus of $E = 2600 \text{ MPa}$, Poisson's ratio $\mu = 0.3859$ and density of $\rho = 1390 \text{ kg/m}^3$. The arm is 200 mm long resulting in 8 segments of 25 mm each. The segments are listed in Table 8. The area moment of inertia is calculated as described in Section 3.4.5:

$$I_x = \frac{h^3 b}{12} = \frac{(13\text{mm})^3 \cdot 28\text{mm}}{12} = 5126.333 \text{ mm}^4 \quad (4.28)$$

$$I_y = \frac{hb^3}{12} = \frac{13\text{mm} \cdot (28\text{mm})^3}{12} = 23781.333 \text{ mm}^4 \quad (4.29)$$

One section of the arm weights

$$m = 25 \text{ mm} \cdot 28 \text{ mm} \cdot 13 \text{ mm} \cdot 1390 \frac{\text{kg}}{\text{m}^3} \quad (4.30)$$

$$= 12.649 \text{ g} \quad (4.31)$$

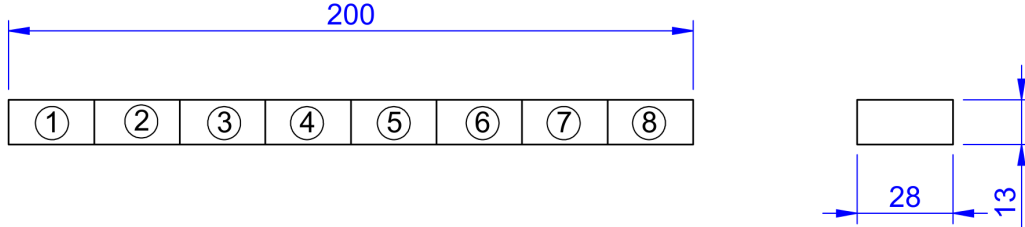


Figure 44: Drawing of the arm

Table 8: Properties of the arm according to Fig 44.

Segment	Length	Width	Height	Weight	I_x	I_y
1	25 mm	28 mm	13 mm	12.65 g	5126 mm ⁴	23781 mm ⁴
2	25 mm	28 mm	13 mm	12.65 g	5126 mm ⁴	23781 mm ⁴
3	25 mm	28 mm	13 mm	12.65 g	5126 mm ⁴	23781 mm ⁴
4	25 mm	28 mm	13 mm	12.65 g	5126 mm ⁴	23781 mm ⁴
5	25 mm	28 mm	13 mm	12.65 g	5126 mm ⁴	23781 mm ⁴
6	25 mm	28 mm	13 mm	12.65 g	5126 mm ⁴	23781 mm ⁴
7	25 mm	28 mm	13 mm	12.65 g	5126 mm ⁴	23781 mm ⁴
8	25 mm	28 mm	13 mm	12.65 g	5126 mm ⁴	23781 mm ⁴

Needle beam

A corresponding drawing of the needle is shown in Fig 45. In the figure is a simplified cross section of the needle. The dimensions are given as D_o/D_i , that is, outer diameter and then inner diameter. The needle is 138 mm long and consists of six 23 mm segments. Material is stainless steel AISI316 with the following properties: elastic modulus $E = 190 \text{ GPa}$, Poisson's ratio $\mu = 0.28$ and density $\rho = 7800 \text{ kg/m}^3$. Due to the changing dimensions, the properties of the segments are listed in Table 9. The area moments of inertia were calculated as above with the arm:

$$I_x = \frac{\pi}{4}(R_o^4 - R_i^4) = \frac{\pi}{4}\left(\left(\frac{2 \text{ mm}}{2}\right)^4 - \left(\frac{1.5 \text{ mm}}{2}\right)^4\right) = 0.537 \text{ mm}^4 \quad (4.32)$$

$$= I_y \quad (4.33)$$

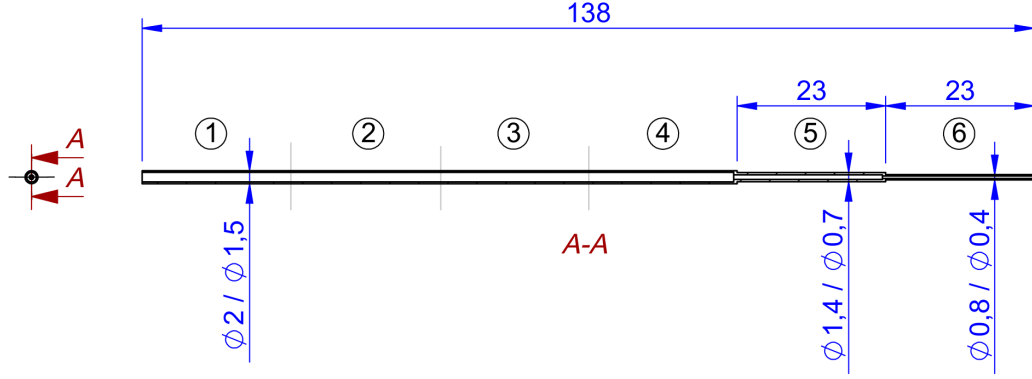


Figure 45: Drawing of the needle

Table 9: Properties of the needle according to Fig 45

Segment	Length	Weight	Outer D	Inner D	A	$I_x = I_y$
1	23 mm	0.245 g	2 mm	1.5 mm	1.374 mm ²	0.537 mm ⁴
2	23 mm	0.245 g	2 mm	1.5 mm	1.374 mm ²	0.537 mm ⁴
3	23 mm	0.245 g	2 mm	1.5 mm	1.374 mm ²	0.537 mm ⁴
4	23 mm	0.245 g	2 mm	1.5 mm	1.374 mm ²	0.537 mm ⁴
5	23 mm	0.215 g	1.4 mm	0.7 mm	1.154 mm ²	0.177 mm ⁴
6	23 mm	0.073 g	0.8 mm	0.4 mm	0.377 mm ²	0.019 mm ⁴

The damping ratios of the flexible components were left to default $1.0 \cdot 10^{-5}$.

Beam natural frequencies

The natural frequency of a beam can be calculated with equation from Section 3.4.4 [27]

$$\omega_n = A \sqrt{\frac{EI}{\mu L^4}} \quad (4.34)$$

where μ is the mass per unit length of beam. For the arm beam we get

$$\omega_{n,x} = 13 \text{ mm} \cdot 28 \text{ mm} \sqrt{\frac{2600 \text{ MPa} \cdot 5126 \text{ mm}^4}{12.65 \text{ g/25 mm} \cdot (200 \text{ mm})^4}} \quad (4.35)$$

$$= 46.70 \text{ Hz} \quad (4.36)$$

$$\omega_{n,y} = 13 \text{ mm} \cdot 28 \text{ mm} \sqrt{\frac{2600 \text{ MPa} \cdot 23781 \text{ mm}^4}{12.65 \text{ g/25 mm} \cdot (200 \text{ mm})^4}} \quad (4.37)$$

$$= 100.60 \text{ Hz} \quad (4.38)$$

These values seem reasonable. The frequency in the x-direction is lower since the

moment of inertia is lower compared to y-direction. For the needle

$$\omega_{n,x} = 1.374 \text{ mm}^2 \sqrt{\frac{190000 \text{ MPa} \cdot 0.537 \text{ mm}^4}{0.245 \text{ g/23 mm} \cdot (138 \text{ mm})^4}} \quad (4.39)$$

$$= 0.22 \text{ Hz} \quad (4.40)$$

$$= \omega_{n,y} \quad (4.41)$$

The needle frequency is simplified as if the whole needle has same geometry. Therefore, the value above is not exact. Additionally, the frequency of 0.22 Hz is not entirely convincing. From empirical tests it is known that the needle vibrates more than hundred times per second, that is, at a frequency higher than 100 Hz. It might be necessary to refine the needle beam model if more tests and analyses are conducted including it.

Degrees of freedom

The arm introduces eight new bodies to the system and the needle six. Each of these bodies have six degrees of freedom. The two fixed joints reduce six DOFs each bringing the Gruebler count of the model to

$$m = 6 \cdot (n_b - 1) - n_c \quad (4.42)$$

$$= 6 \cdot (5 + 8 + 6 - 1) - 6 - 5 - 5 - 1 - 1 - 4 - 1 - 6 - 6 \quad (4.43)$$

$$= 6 \cdot 18 - 35 \quad (4.44)$$

$$= 73 \quad (4.45)$$

The transformation from the full model to the simulation model is shown in Fig 46. The 3D parts were imported to ADAMS as parasolids.

Measures

Eight measures were needed and they are described in Table 10. The needle's tip's deflection is calculated utilizing the rotational angle of the tip as it rotates around the shaft, this is done in measure *Meas_Ndl_angle*. The angle measure consists of three points/markers: the first one is at the tip of the needle, the second one at the rotational axis of the shaft and the third one fixed to ground between the two previous points. As a result, the angle is zero at rest and increases as the unit starts to rotate. Additionally, the rotational angle of the upper motor/pulley is used. It works as a reference as to where the tip of the needle should be located. Finally, the unit's arm is 200 mm long.

The function to calculate the deflection at the tip of the needle is thus (angles are in radians)

$$Func_Ndl_defl = (Meas_Ndl_angle - Meas_MotorUp_angle) \cdot 200 \quad (4.46)$$

The function *Func_Ndl_dist* is defined as

$$Func_Ndl_dist = Meas_Ndl_angle \cdot 200 \quad (4.47)$$

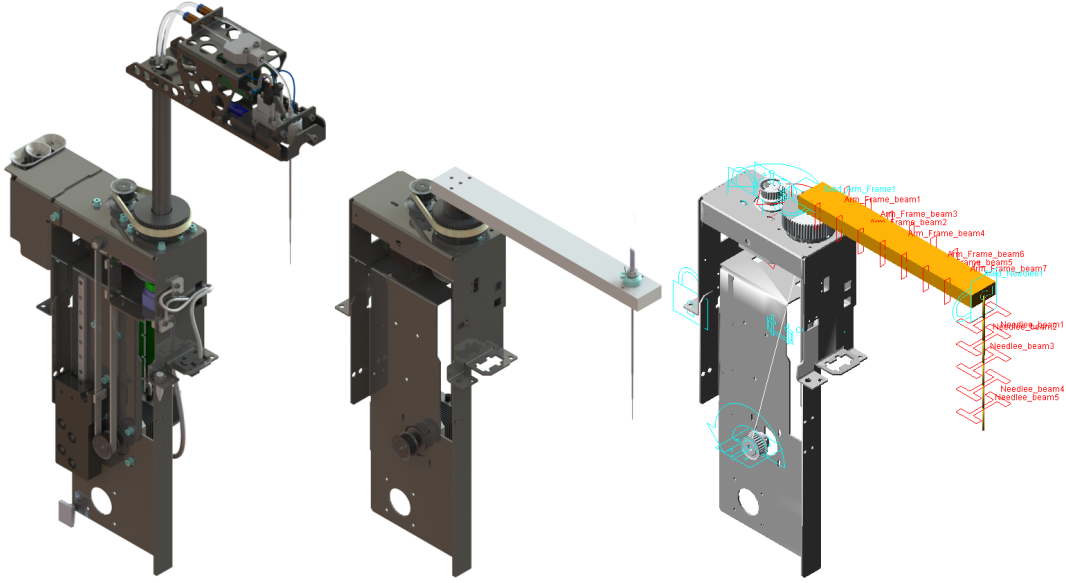


Figure 46: Evolution of the model. Left: full model, middle: simplified model, right: ADAMS model

Table 10: A list of measures in the model

Type	Name	Description
Angle	Meas_MotorUp_angle	Rotational angle of the upper Motor shaft
Torque	Meas_Belt_torque	Torque of the torsional spring-damper, equals to torque generated by the upper Motor/Pulley
Angle	Meas_Ndl_angle	Rotational angle of the tip of the needle
Angle	Meas_PullBr_angle	Rotational angle of the bearing assembly
Distance	Func_Ndl_defl	Deflection of the tip of the needle, see separate description
Distance	Func_Ndl_dist	Distance the tip of the needle has travelled
Distance	Func_PullBr_dist	Imaginary distance of bearing assembly moved scaled to match the tip of the needle
Distance	Func_MotorUp_dist	Imaginary distance of motor driven scaled to match the tip of the needle

The function $Func_PullBr_dist$ is defined as

$$Func_PullBr_dist = Meas_PullBr_angle \cdot 200 \quad (4.48)$$

The function $Func_MotorUp_dist$ is defined as

$$Func_MotorUp_dist = Meas_MotorUp_angle \cdot 200 \cdot \frac{24}{60} \quad (4.49)$$

4.2.4 Analysis

Two different scenarios were analysed based on real life situations:

1. Controlled stop from high speed
2. Sudden stop from low speed

The first item above is relevant in normal operation. The unit is moved as quickly as possible to obtain high productivity. Empirical studies have shown the needle to tend to vibrate somewhat even though the movement is relatively smooth. In the analysis the unit is rotated for two seconds at maximum speed of $0.4 \text{ rev/s} = 144 \text{ deg/s}$ and then stopped in 130 milliseconds. The two second rotation is required for vibrations induced during the acceleration to vanish and thus not interfere the actual measurements during deceleration. In two seconds time the unit rotates approximately one full rotation.

In the second scenario the unit is rotated at a low speed of $0.05 \text{ rev/s} = 18 \text{ deg/s}$ and then suddenly stopped in 5 milliseconds. This imitates the procedure to find the unit's home position which is done with optical sensors. A value of 5 ms was chosen since it is short enough to seem like a sudden stop but long enough for ADAMS' integrator not get confused. A truly instant velocity change with an IF condition could not be used because the integrator demands few calculation points in the deceleration phase. When the unit is rotating and the optical sensor changes its state, the motor controller stop the unit instantaneously. The unit is rotated for 2 seconds before the stop, this equals to approx 45 degrees.

Simulation time was set to 5 seconds with a 0.001 second step size. The 5 seconds include one second of acceleration, two seconds of rotation and two seconds of measuring time after the movement has stopped. Simulation type is dynamic due to multiple degrees of freedom. All simulations were started at equilibrium.

Input motion to the system is the upper motor/pulley which handles the rotational movement of the unit. The lower motor/pulley, that is, the y direction, is kept at zero. The upper motor/pulley angle, velocity and acceleration graphs are shown in Figs 47 and 48 for controlled stop and sudden stop tests, respectively. Since *Step*-functions were used in ADAMS, the movements are smooth as can be seen in the acceleration graph of Fig 47.

The simulated deflections and vibrations at the tip of the needle are displayed in Figs 49 and 50. Included is a FFT transformation of the graph. The FFT transformation was done with *Hamming* type window and 4096 points.

During controlled stop the needle has first a high peak with a maximum deflection of 7.4 mm. After this the vibrations start to dampen. From the FFT transformation it can be seen that the primary vibration happens at a frequency of approximately 5.9 Hz.

In the sudden stop a much lower amplitude is observed. Immediately after the stop amplitude is almost 0.9 mm. Like in the controlled stop, the amplitude decreases as time progresses. In the FFT graph a strong peak can be seen at the same frequency of 5.9 Hz as in controlled stop.

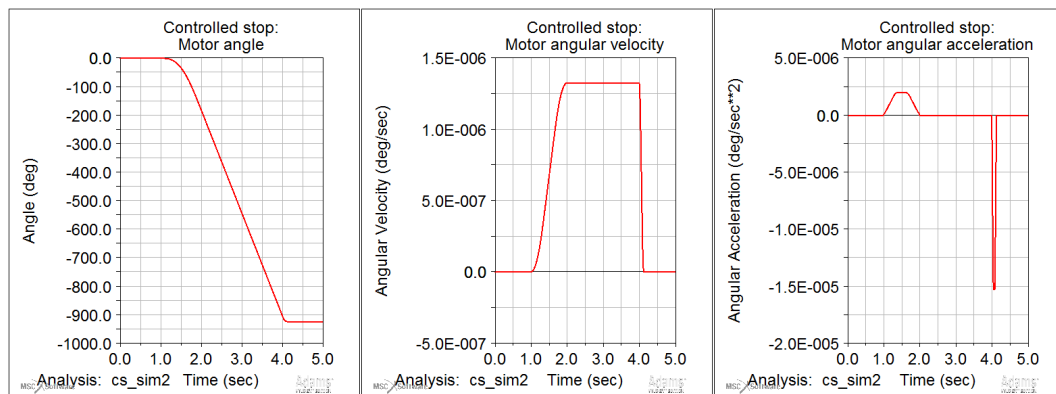


Figure 47: Upper motor/pulley movement in controlled stop test

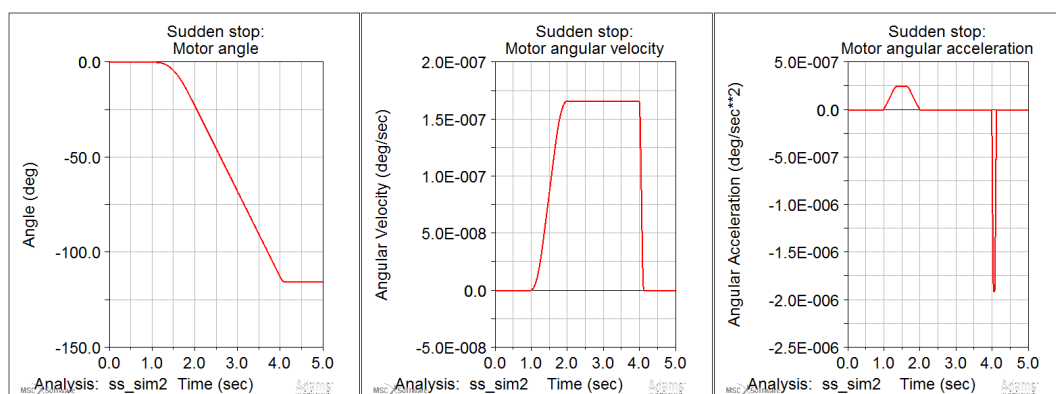


Figure 48: Upper motor/pulley movement in sudden stop test

In Fig 51 stepper motor torque magnitudes for ϕ direction can be seen. During the controlled stop maximum torque is approximately 30 Nmm and in sudden stop 3.7 Nmm. The rated holding torque for this motor model is 34 Nmm according to manufacturer.

The controlled stop torque is very close to this value but the sudden stop torque is quite significantly lower. It must be remembered though, that in the model the sudden stop is fixed to 5 ms while in reality the phenomenon is immediate. Perhaps with even quicker sudden stop the motor holding torque would be higher.

4.2.5 Results Verification

The results were checked with a real system. The unit described in Fig 37 was mounted on a test platform, shown in Fig 52. The platform is manufactured of 3 mm aluminum and its dimensions are approximately 430 x 430 x 530 mm (W x D x H). The platform is used to test the robot in various situations and tasks.

The real world test was executed with the same parameters as the simulations. A high speed video camera was used to capture the stops at 400 fps. The video was

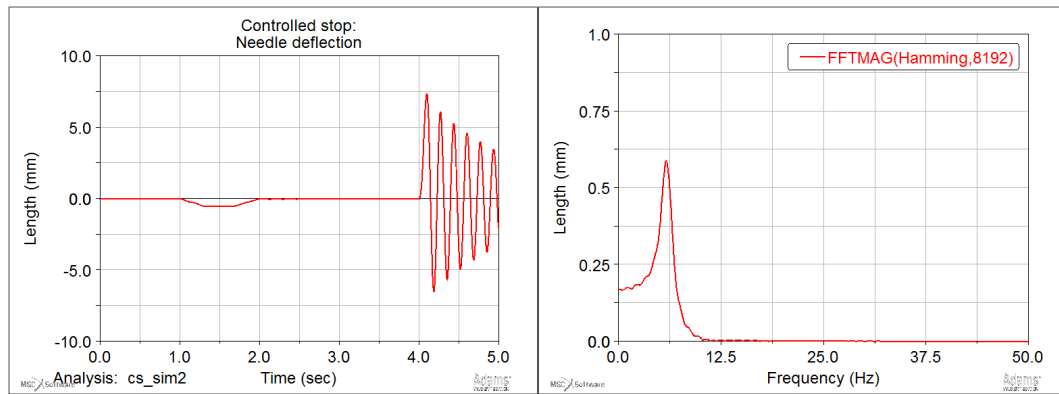


Figure 49: Needle tip deflection and FFT during controlled stop

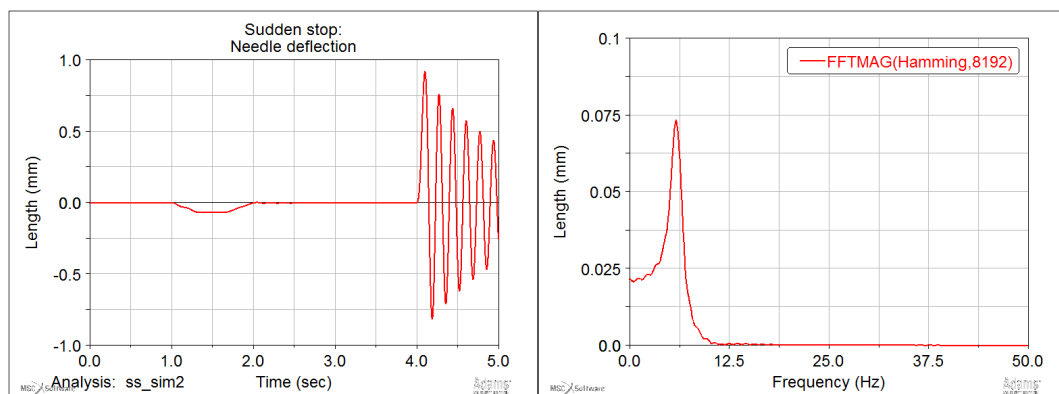


Figure 50: Needle tip deflection and FFT during sudden stop

then analysed with tracking software. In Fig 53 is a scene from the video showing the needle and a ruler.

The test data from Tracker was then exported to a table, fine-tuned with a spread sheet program and finally imported as a measurement to ADAMS. The data could then be used in ADAMS as any other measurement. The test data was then presented in the same graph as the simulated data. Controlled stop results can be seen in Fig 54 and sudden stop in Fig 55.

The results of simulated vs. measured needle deflection as seen in Figs 54 and 55. In the controlled stop, measured amplitude is 4.9 mm and simulated 7.4 mm. The same can be seen in sudden stop test where measured is 1.1 mm and simulated 0.9 mm. The frequencies are more corresponding: in controlled stop measured 6.3 Hz and simulated 5.9 Hz, in sudden stop measured 7.0 Hz and simulated 5.9 Hz. Notable from the FFT graphs is, that there is a second resonance at approximately 40 Hz in both Figs 54 and 55. This secondary peak is completely absent in the simulated results but is close to the natural frequency of the arm, 46.7 Hz, as calculated in Section 4.2.3.

In Fig 56 there are graphs of needle velocity in both stopping scenarios. Also in

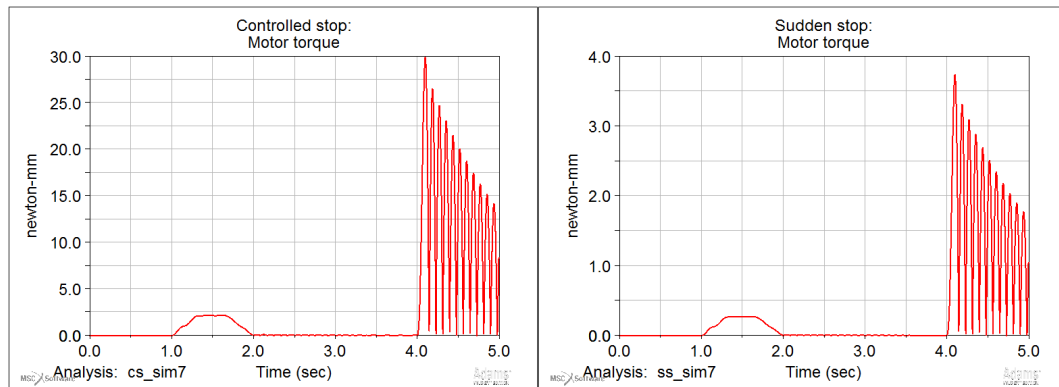


Figure 51: Stepper motor torques

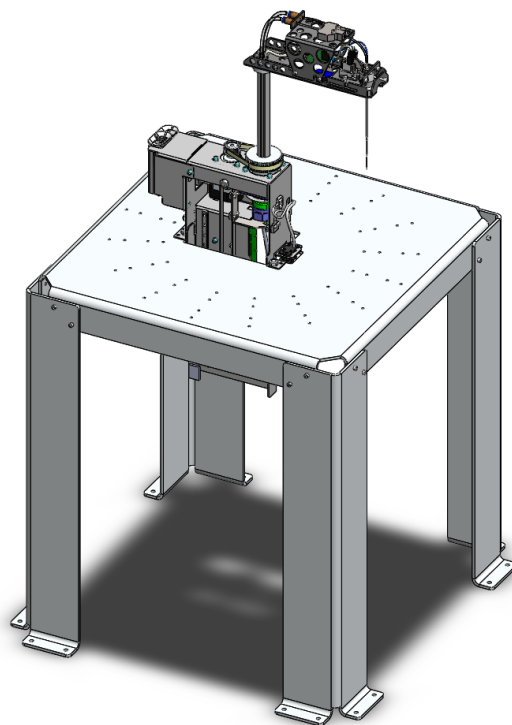


Figure 52: Dispensing robot mounted on a test platform

the figure graphs of needle and motor distance are seen but they are only for the purpose of calculating their derivatives to determine their speeds. From the graph it can be observed that after the motor has begun decelerating the needle maintains a velocity for a short time. Then the needle's speed decreases dramatically to the point where the motor has stopped but the needle has too much kinetic energy to also stop. Only after some time after the motor stop the needle's amplitude reaches its maximum and needle's velocity changes direction. The needle then begins to oscillate for some time.

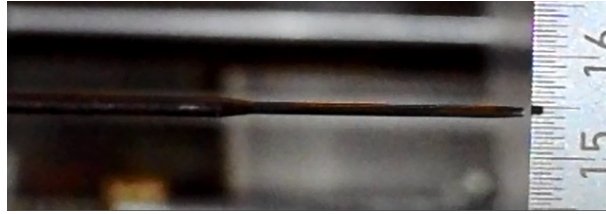


Figure 53: A scene from a high speed video showing the needle and a ruler

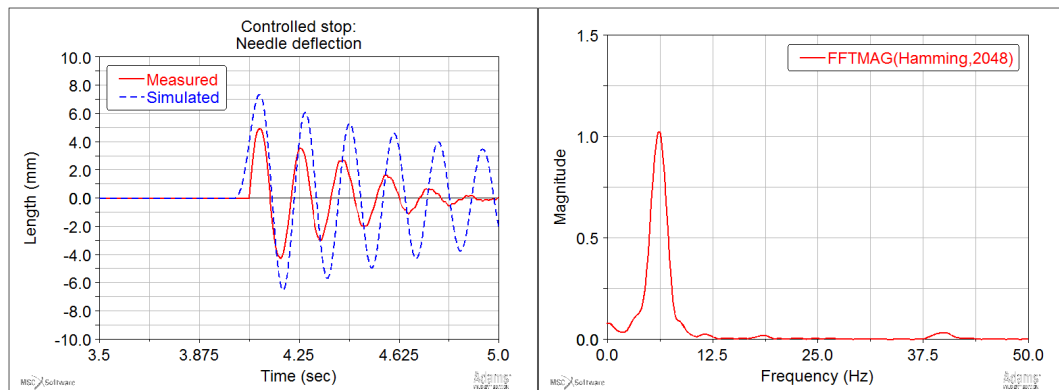


Figure 54: Measured and simulated needle deflection in controlled stop

Additionally, the impact of the different flexible components in Table 7 were analysed. First, the whole system as a reference, then, the system without the flexibility of arm and needle, and thirdly, with the flexibility of the belt, arm and needle. An analysis with just the arm and needle could not be ran successfully as ADAMS encountered problems with the integrator and aborted the simulation.

The results are described in Fig 57 and Table 11. The amplitude of the vibrations at the end of the needle was the result. From the Table it can be observed that the

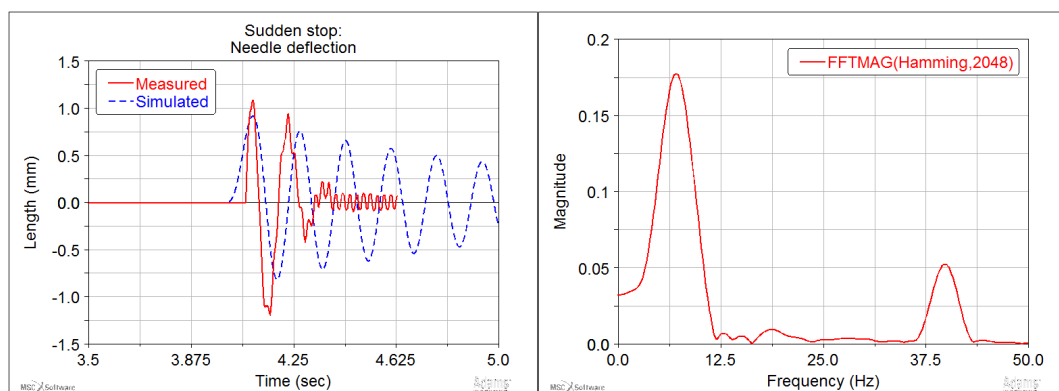


Figure 55: Measured and simulated needle deflection in sudden stop

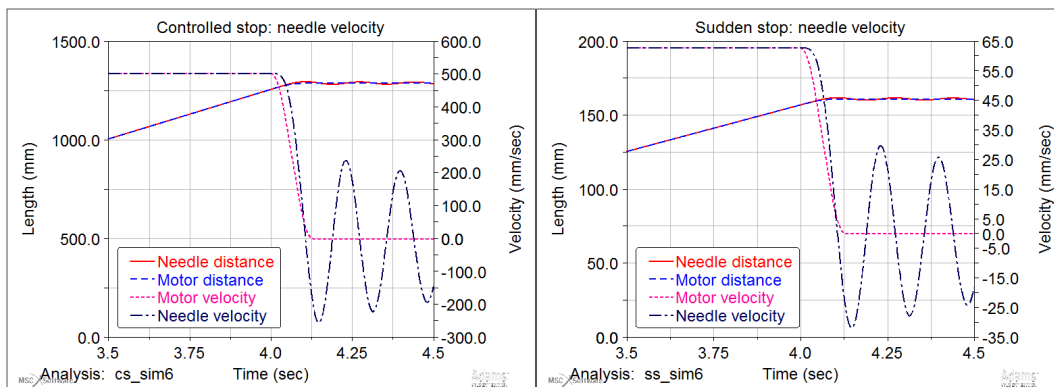


Figure 56: Needle velocity compared to motor velocity in controlled and sudden stops

stepper motor accounts over 99 % of the amplitude. The belt, arm and needle account for few percents at most. The sum of percentages does not add up to 100 % due to separate analyses.

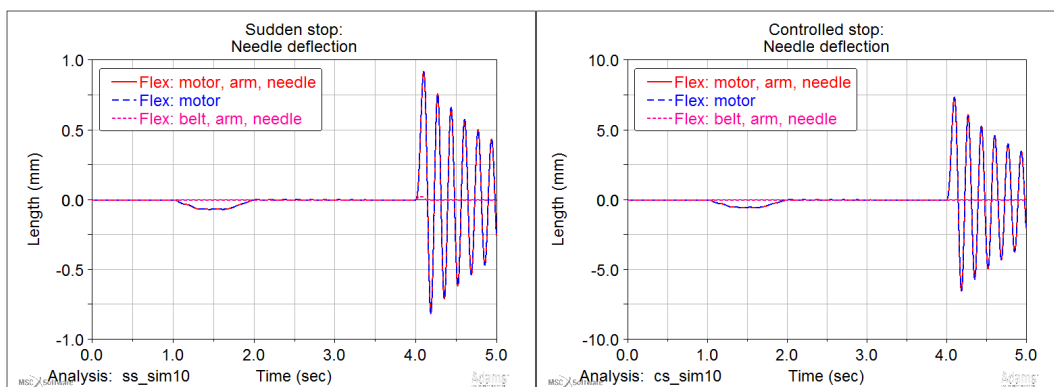


Figure 57: Needle deflection compared to motor velocity in controlled and sudden stops

Table 11: A list of analyses with different flexible components

Flex parts	A, ctrl	%, ctrl	A, sudden	%, sudden
Motor, belt, arm, needle	7.357 mm	100 %	0.919 mm	100 %
Motor, belt	7.331 mm	99.65 %	0.916 mm	99.67 %
Belt, arm, needle	0.003 mm	0.04 %	0.023 mm	2.50 %
Arm, needle	n/a	n/a	n/a	n/a

4.2.6 Conclusions

In this Section, a vibrational problem was introduced involving a polar coordinate robot and a needle. The problem were vibrations induced in the unit when rotating and then stopping. These vibrations are occasionally somewhat a problem in real life, especially when doing precise movement with tight tolerances. A MBS model of the unit was constructed in ADAMS to examine the problem.

The original model of the robot unit consists of 28 components most of which are bolts. The simulation model was constructed of 6 imported parts and from eight and six segment discrete beams. The two beams were ADAMS' discrete flexible links. Each segment had six DOFs bring the system's free degrees to 73. Thus, the simulation were ran as dynamic. The beams' properties were replicated according to real world parts. There were also two input motions to the system but one of these was not utilized in simulations. Excluding the beams the system had one degree of freedom.

The stepper motor presents an additional flexible component to the system. This flexibility was modelled as a torsional spring combined with the stiffness of the toothed timing belt being driven by the motor. The rotational stiffness of the motor was determined with a frequency analysis done with a known inertia on the motor's axis and a high speed camera. The belt's stiffness was obtained from the manufacturer. Both these stiffnesses were combined and reduced to a single torsion spring located on the motor's axis.

Two different scenarios were analysed with the simulation model: one with a controlled stop and another with a sudden stop. Both of these scenarios were replicated based on real world observations and driving parameters. In the controlled stop the unit is accelerated to rated rotational speed and then with a smooth deceleration brought to stop within 130 milliseconds. The sudden stop, on the other hand, has a more aggressive stop of just 5 milliseconds representing a very abrupt stopping of the unit. The controlled stop occurs often in real normal operation whereas the sudden stop less frequently but still perhaps once a day.

The construction of the model was straight forward and first results were quickly obtained. Comparing the sudden stop results to actual measurements the was clearly coherence. The amplitude and frequency of the vibrations were quite close in both scenarios. From this we can deduce that the model was successfully built with sensible parameters.

In the controlled stop though, the simulation model turned out to be too stiff compared to real life measurements. Possible explanations for this behaviour were the stepper motor or the needle beam. The stepper motor's stiffness might not be linear as presumed but varies depending on the position of the axis and angle from the stopped or equilibrium position. Another explanation is the needle which was modelled as a beam. As described in Section 4.2.3, the calculated natural frequency of the needle is far from empirical measurements. This might result as incorrect deflections for the needle.

The root cause for the vibrations turned out to be stepper motor, being responsible of a very significant percentage of over 95 % of maximum amplitude in both

scenarios. The result of the analysis is that if the vibrations are to be reduced, something must be done to the stepper motor. One solution might be a mass-spring damper mounted on the stepper motor's axis and especially designed to match its natural frequency, as described in Section 3.6.1. This kind of damper could be easily implemented for a variety of different steppers and applications.

In terms of generally aiding research and development, the root cause of the vibrations is a significant piece of information. Additionally, the effects of other components can be ruled out. With a fully functioning simulation model different usage scenarios could easily be analysed and the vibrations estimated. This helps in determining suitable driving sequences and parameters while minimizing possible vibrations. This saves a considerable amount of empirical testing time and result analysing as the simulation model's driving sequence can be modified in seconds.

5 Conclusions

As described in Section 3, multibody systems are systems comprised of multiple parts and components attached together with joints. Often there are motions or external forces applied to the parts' changing their dynamics and behaviour. Multibody system dynamics is a field of applied mechanics that studies these systems. Commonly, the aim for using MBD tools is to determine the behaviour or movement of parts in a multibody system under the influence of forces and motions. Due to heavy mathematics involved, a computer is generally utilized. Multibody simulation programs are computer applications that are designed to solve and illustrate multibody system problems.

Multibody simulation is a growing trend in product development. While some large corporations have been using MBD simulation software for many years, smaller businesses are often wondering about the benefits, savings and potential they could produce. There are numerous applications of MBD software to different types of problems. For instance, dynamical, vibrational and flexibility questions can be solved. One such program is MSC Software's ADAMS which is widely used in various industries and offers many features.

The purpose of this thesis is to present companies and businesses a showcase of MBD's capabilities and benefits. The work was written in collaboration with Thermo Fisher Scientific Oy located in Vantaankoski, Vantaa, the corporation's headquarters in Finland. At TFS Finland a MBD software has not been utilized in product development before. In this work, a quick introduction to multibody simulations and dynamics is done followed by methods used to analyse these systems. Finally, two problems are solved with MSC ADAMS. These problems were picked from the product development of clinical analyzers. This thesis aims to help visualize the capabilities of the software to the designers in product development with familiar problems to them.

In the first problem a lifting aid for a main cover is dimensioned and chosen. The aid is a gas spring chosen from the catalogues of various manufacturers. The position and force of the spring were determined with MSC ADAMS. The positioning included resolving the attaching points of the gas spring and the spring's dimensions. The required force generated by the spring was also obtained. There were multiple design goals for the lifting aid which all were met. For example, only a small force was to be required to be able to open the main cover. A prototype was constructed to verify the results and it proved to function very well and as intended. The cover was easy to open and stayed open at any angle. Additionally, the gas spring could be effortlessly installed. The problem was considered to be successfully solved and the designer of the main cover was pleased with the results.

The second problem was a vibrational one. In product development it has been observed that the needle of a dispensing robot vibrates after the robot's movement stops. This has somewhat been an issue and its root cause has been unresolved. A simulation model was built with a purpose to find out the principle behind of these vibrations and to test how accurately the phenomenon can be simulated. A model was built with ADAMS and 3D parts imported. In the model, four parts which were

suspected to be the origin of the vibrations were made flexible. Flexible modelling elements used were discrete beams and torsion springs. For comparison purposes, real life measurements were recorded with a high speed camera and processed with an image tracking software. The simulation model turned out to be reasonably close to real situation. More importantly, the main cause behind the vibrations was found: the stepper motor accounted for more than 95 % of the vibration amplitude.

The benefits of multibody simulations is shown clearly with these two problems. In the first one where a lifting aid is devised, the effort of designing a well performing system is drastically reduced. Usually the design takes few prototypes spanning multiple weeks or even months to mature to an acceptable level, costing valuable time and money. With a MBD software, the task takes only a few days and without the need for an actual prototype. In the second problem where needle vibrations were investigated, the added value comes from knowing the real cause of the phenomenon. Once it is found out the issue can be negated by designing and adding a vibrational damper to the system. By eliminating the vibrations altogether the dispensing robots' as well as the whole instrument's performance can be increased possibly leading to growth reliability and in sales.

References

- [1] About thermo fisher scientific, 2014. URL <http://www.thermofisher.com/en/about-us.html>. Retrieved 25.12.2014.
- [2] Robert A. Adams and Christopher Essex. *Calculus: A complete course*. Pearson Education, 7th edition, 2009. ISBN 978-0321549280.
- [3] Haim Baruh. Another look at the describing equations of dynamics. *Journal of the Chinese Society of Mechanical Engineers, Transactions of the Chinese Institute of Engineers, Series C/Chung-Kuo Chi Hsueh Kung Ch'eng Hsuebo Pao*, 21(1):15–23, 2000.
- [4] O. A. Bauchau. *Flexible Multibody Dynamics*, volume 176 of *Solid Mechanics and Its Applications*. Springer Netherlands, 2011. ISBN 978-94-007-0335-3. doi: 10.1007/978-94-007-0335-3.
- [5] Mike Blundell and Damian Harty. *Multibody Systems Approach to Vehicle Dynamics*. SAE International, 2004. ISBN 0768014964.
- [6] Monica Bordegoni and Caterina Rizzi. *Innovation in Product Design: From CAD to Virtual Prototyping*. Springer London, 2011. ISBN 978-0-85729-774-7. doi: 10.1007/978-0-85729-775-4.
- [7] L.L.C. BRECOflex Co. Polyurethane timing belts. Brochure.
- [8] Douglas Brown. Tracker: Video analysis and modeling tool, 2014. URL <https://www.cabrillo.edu/~dbrown/tracker/>. Retrieved 18.11.2014.
- [9] MSC Software Corporation. Adams help. Electronic, 2013.2.
- [10] J. D'Alembert. *Traité de Dynamique*. David, Paris, 1743.
- [11] Peter Eberhard and Werner Schiehlen. Computational dynamics of multibody systems: History, formalisms, and applications. *Journal of Computational and Nonlinear Dynamics*, 1(1):3–12, January 2006. doi: 10.1115/1.1961875.
- [12] L. Euler. Nova methodus motum corporum rigidorum determinandi. *Novi Commentarii academiae scientiarum imperialis Petropolitanae*, 20, 1776.
- [13] Giancarlo Genta, editor. *Vibration Dynamics and Control*. Springer US, 2009. ISBN 978-0-387-79580-5. doi: 10.1007/978-0-387-79580-5.
- [14] Patrick Guillaume. Modal analysis. Department of Mechanical Engineering, Vrije Universiteit Brussel.
- [15] Douglas W. Jones. An907: Stepper motor fundamentals. University of Iowa, 2004.

- [16] Thomas R. Kane and David A. Levinson. *Dynamics, Theory and Applications*. McGraw Hill, 1985. ISBN 0070378460.
- [17] J.-L. Lagrange. *Mécanique Analytique*. L'Académie Royal des Sciences, Paris, 1788.
- [18] Lesjöfors. Stock spring catalogue no 13. Electronic, 2014.
- [19] Simon J. A. Malham. *An Introduction to Lagrangian and Hamiltonian Mechanics*. Maxwell Institute for Mathematical Sciences, 2014.
- [20] MSC.Software.Corporation. Adm704-705 section 8: Optimization, 2005.
- [21] MSC.Software.Corporation. Adams: Real dynamics for functional virtual prototyping, 2013.
- [22] MSC.Software.Corporation. Adams: Multibody dynamics for functional virtual prototyping, 2014.
- [23] MSC.Software.Corporation. Multibody dynamics, 2014. URL <http://www.mscsoftware.com/application/multibody-dynamics>. Retrieved 15.7.2014.
- [24] Isaac Newton. *Philosophiae Naturalis Principia Mathematica*. Royal Society, London, 1687.
- [25] Schote Norbert. Wheel suspension for a front axle of an automobile, Nov 2000. URL <http://www.google.com/patents/WO2000066377A2>. Patent WO2000066377 A2.
- [26] Robert L. Norton. *Design of Machinery: An Introduction to the Synthesis and Analysis of Mechanisms and Machines*. Mcgraw-Hill, 5th edition, 2012. ISBN 978-0-07-352935-6.
- [27] Allan Piersol and Thomas Paez. *Harris' Shock and Vibration Handbook*. McGraw Hill, 2009. ISBN 9780071508193.
- [28] A. Purushotham and J. Anjeneyulu. Kane's method for robotic arm dynamics: a novel approach. *IOSR Journal of Mechanical and Civil Engineering*, 6(4): 7–13, 2013.
- [29] Kari Santaoja. *Lujuusoppi I. Sasata*, 2010. ISBN 9789526718910.
- [30] Kari Santaoja. Kul-49.3400 lecture notes. Electronic, 2012.
- [31] W. Schiehlen. Multibody system dynamics: Roots and perspectives. *Multibody System Dynamics*, 1(2):149–188, 1997.
- [32] W. Schiehlen. Research trends in multibody system dynamics. *Multibody System Dynamics*, 18(1):3–13, 2007. doi: 10.1007/s11044-007-9064-4.

- [33] M. Selic, A. Reekers, and A. Steinacker. Robot arm, July 2001. URL <http://www.google.com/patents/USD444488>. US Patent D444488 S1.
- [34] Ahmed A. Shabana. *Dynamics of Multibody Systems*. Cambridge University Press, 4th edition edition, 2013. ISBN 978-1-107-04265-0.
- [35] Siegling. Timing belts - calculation methods. Brochure.
- [36] WolframMathWorld. Circular sector, 2014. URL <http://mathworld.wolfram.com/CircularSector.html>. Retrieved 3.9.2014.

A First appendix: HAHN G 10-28

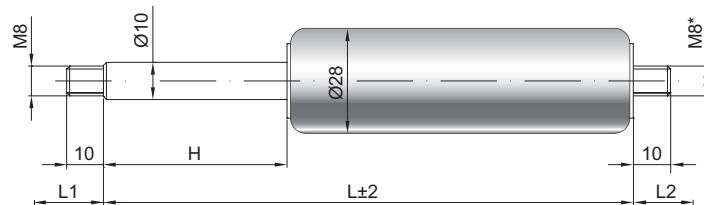
Hahn G 10-28 brochure

Gas spring G 10-28

Extension force F1	100 - 1200 Newton
Progression	20 %
Piston rod	Ø 10 mm, stainless steel
Cylinder	Ø 28 mm, stainless steel, black spray coated
Threads	Steel zinc plated

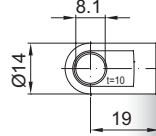
GZ 10

GZ 10

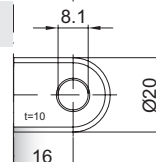


Details

AU 19



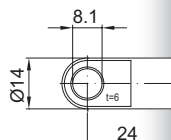
Standard strokes H (mm)
 40, 50, 60, 80, 85, 100, 120, 140, 150,
 160, 180, 200, 220, 250, 270, 300, 325,
 350, 400, 450, 500, 600, 700



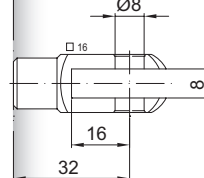
AB 16

only possible on cylinder end

AS 24

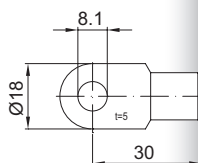


Standard length GL
 $L = 2 \times H + 70$
 $GL = L + L1 + L2$

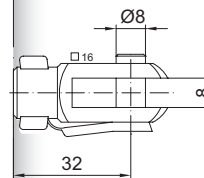


GA 32

AS 30

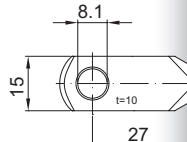


Extras
 1 - Wiper ring
 2 - Radial valve
 4 - Grease chamber
 5 - Valve
 6 - Protection tube
 7 - Floating piston
 8 - Locking device
 B3 - Sealing system
 NT - Low-temperature design
 HT - High-temperature design
 RK - Friction element

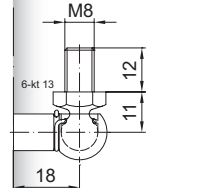


GG 32

AR 27



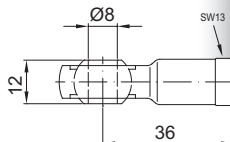
*Threaded end optional M8 or M10



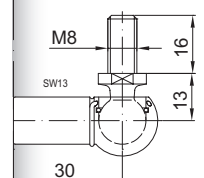
WG 18

until max. 800 N

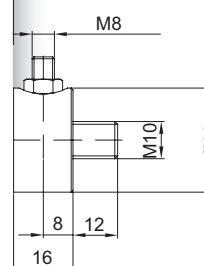
AG 36



More **threads** can be found here [▶](#)
 Matching **brackets** can be found here [▶](#)



WG 30



QV GZ