Saimaa University of Applied Sciences
Faculty of Technology, Lappeenranta
Degree Programme in Mechanical Engineering and Production Technology

Anna Nikitina

# Flexible body tutorial for Mevea real-time simulation software

Thesis 2017

# Abstract

Anna Nikitina
Flexible body tutorial for Mevea real-time simulation software, 66 pages, 3 appendices
Saimaa University of Applied Sciences
Faculty of Technology, Lappeenranta
Degree programme in Mechanical Engineering and Production Technology
Thesis 2017
Instructors: Lecturer Simo Sinkko, Saimaa University of Applied Sciences
Director, Customer Projects, Vesa Ahola, Mevea Ltd.

The main goal of this study was to develop a tutorial about creation of flexible bodies in Mevea real-time simulation software. The work was commissioned by Mevea Ltd.

Various sources were used to collect the data for the study. Background information about multibody system dynamics was received during Simulation of Mechatronic Machine course by Aki Mikkola, Lappeenranta University of Technology, as well as from books found using resources of Lappeenranta Academic Library. Previous tutorials for Mevea software were used as a training material, combined with guidance from Mevea personnel.

The final result of this thesis was the tutorial about flexible body creation in Mevea Modeller, and ways of customising Mevea Solver window. The initial concept of the tutorial was changed to adopt the needs of the commissioner better. The tutorial demonstration and analysis of future opportunities showed possible ways of developing Mevea tutorial base: creating video tutorials, separating the tutorials into smaller parts, including more topics that interest students.

Keywords: Mevea, real-time simulation, multibody system dynamics, flexible body, tutorial

## Table of contents

List of Abbreviations ........ 5
1 Introduction ........ 6
  1.1 Background and motivation ........ 7
  1.2 Objectives ........ 9
  1.3 Structure of the thesis ........ 9
2 Theoretical overview ........ 10
  2.1 Real-time simulation ........ 11
  2.2 Multibody system dynamics ........ 12
    2.2.1 Multibody systems ........ 12
    2.2.2 Solution of MBS using method of Lagrange multipliers ........ 14
    2.2.3 Recursive solution method in MBS ........ 15
    2.2.4 Rigid body kinematics in MBS ........ 17
    2.2.5 Degrees of freedom and joints ........ 19
  2.3 Modelling of hydraulics in real-time environment ........ 21
    2.3.1 Viscosity ........ 22
    2.3.2 Bulk modulus ........ 23
    2.3.3 Flow continuity equation ........ 25
    2.3.4 Description of selected hydraulic components ........ 25
  2.4 Modelling of flexible bodies in MBS ........ 27
    2.4.1 Solution of connector forces ........ 27
    2.4.2 Floating frame of reference ........ 29
    2.4.3 Solution of the shape matrix ........ 30
  2.5 Creation of software tutorials ........ 33
3 Tutorial creation ........ 33
  3.1 Creation of the initial tutorial model ........ 34
    3.1.1 Solution method choice ........ 35
    3.1.2 Creation of graphics ........ 35
    3.1.3 Bodies and respective coordinate systems ........ 36
    3.1.4 Constraints ........ 39
    3.1.5 Forces ........ 40
    3.1.6 Dummies ........ 43
    3.1.7 Collisions ........ 44
    3.1.8 Hydraulics ........ 45
    3.1.9 Inputs ........ 51
  3.2 Adding flexible body to simulation model ........ 52
  3.3 Tutorial production ........ 53
    3.3.1 Changing Basic Parameters ........ 54
    3.3.2 Recording input ........ 55
    3.3.3 Creating plots ........ 55
    3.3.4 Saving and Loading Workspace ........ 56
4 Discussion ........ 57
  4.1 Tutorial demonstration with Saimaa UAS students ........ 58
  4.2 Possibilities for improvement and future topics for tutorials ........ 59
5 Conclusion ........ 61
References ........ 63
List of figures ........ 65
List of tables ........ 66
Appendix 1. Mevea tutorial ........ 67

Appendix 2. Feedback form for tutorial demonstration with Saimaa UAS students 76
Appendix 3. Summary of feedback form responses ............................................. 78

# List of Abbreviations

CS – Coordinate system

DAE – Differential algebraic equation

FE – Finite element

MBS – Multibody systems

ODE – Ordinary differential equation

# 1 Introduction

This thesis focuses on development of a tutorial for creation of a flexible body model in Mevea real-time simulation software. As it was identified during discussions with Mevea management, the current tutorial system was created a long time ago and consists of only four tutorials, which are not sufficient to cover all the major topics in Mevea software. The first area that Mevea decided to cover with new tutorials is modelling of a flexible body with the simple jib boom as an example.

In most of the cases in real-time simulation rigid bodies are enough to describe the behaviour of the system. However, to depict the behaviour of the system more accurately, flexible bodies are introduced into the system. One of the reasons why flexible bodies are not that widely used for modelling, is because the simulation of a flexible body is complicated regardless the method used and requires a lot of computational power. Nevertheless, with modern day computers flexible bodies can be used for real-time simulation without major problems, and Mevea software allows to use flexible bodies without problems.

Mevea is a Finnish company that specialises in real-time simulation and simulator technology. Its history started in 2005 as a spin-off company of Lappeenranta University of Technology. Currently, the company has more than 50 customers around the globe, most of them still in Finland, however more and more customers are coming from abroad. Some of the most notable customers are Liebherr Group Ltd., Ammann Group Holding AG, Sandvik Mining and Construction Ltd., Normet Group Ltd., VTT – Technical Research Centre of Finland Ltd., Mantsinen Group Ltd. Mevea also maintains close ties with universities, such as Lappeenranta University of Technology, and research institutes in order to be on the verge of the real-time simulation technology (Mevea Ltd., 2016)

Mevea simulation software is a totally in-house built, meaning that Mevea is able to fulfil the customer’s needs in the best way possible. Mevea software package for real-time simulation consists of three parts: Mevea Modeller, Mevea Solver and Mevea I/O Toolbox. The first program is used to develop the models and their environment; the second is used to run the real-time simulation and record selected parameters to be used for research later; the last one is used to connect external systems to the simulation, for instance Simulink simulation models, machine control and automation systems, motion platform and other hardware. Mevea simulation models are used a lot in trainings, therefore a separate package for training management exists. It consists of two parts: Training Manager and Task Editor. The first one focuses on managing students, courses and reporting, while the second one is used to edit existing and add new training scenarios to the simulator (Mevea Ltd., 2016). During the work on this thesis only Mevea Modeller and Mevea Solver were used.

## 1.1 Background and motivation

Computer simulation has become a hot topic in product development and training over the last decade. The theoretical background started to form in the late 1980s, when the computational power of the computers was not sufficient enough to handle complicated differential equations needed for real-time simulation. Nowadays with increasing computational power, computer real-time simulation becomes more wide-spread and efficient.

Unlike traditional modelling and prototyping, which is based on analytical solutions or empirical testing, computer simulation offers quicker response to changing environment and design parameters. The user of real-time simulation model is able to change the parameters of the internal components of the machine by simply changing several parameters in the model instead of replacing the whole component of the machine.

The other way of using real-time simulation is for marketing purposes, especially for user training and product showcasing. With real-time simulator the company is able to conduct the staff trainings with decreased costs, as one simulator station can be adapted to various machines, and the idle time that is previously used for training is decreased, meaning successive increase in productivity.

In the modern quickly evolving world the speed is the key to competitivity, and real-time simulation is the approach that allows the companies to save their resources to produce and deliver better and more competitive products and services.

Skinnarila campus in Lappeenranta is a great place to get make the first steps in real-time simulation. Both higher education institutions here are using real-time simulation in their studies: Saimaa University of Applied Sciences focuses on practical implementation and teaching students how to build workable models using existing software, while Lappeenranta University of Technology provides the students with theoretical knowledge about real-time simulation and supports deeper theoretical research that is needed to comprise more accurate models.

In order to encourage students to take courses about real-time simulation, the universities use simulators by Mevea Ltd., which are available for students to try out. Figure 1 shows two four-wheel loader simulators that are installed in Lappeenranta University of Technology corridor, open for public use.

Figure 1. Simulation studio in Lappeenranta University of Technology

These simulators always attract a lot of people, as they allow them to try out the machine that they would be otherwise unable or not allowed to use. It is understandable that there is always something behind the model and the fact that it is publicly available, makes students interested in what is behind that model. Luckily, they are able to study real-time simulation and Mevea software in the university, however not all of the Mevea customers are able to dedicate time for a specific training. This is the moment when the tutorials on various topics come into use, but currently Mevea tutorial library is limited and requires more topics to be covered.

### 1.2 Objectives

The objectives of this thesis are relatively simple. Firstly, it should provide enough background knowledge on the topics that are used during the creation process of the tutorial. Secondly, the thesis should cover the creation process for the multibody model and flexible boom. Finally, the practical outcome of the thesis should be a tutorial that can benefit Mevea Ltd. and its customers in the future.

### 1.3 Structure of the thesis

The first part of the thesis is dedicated to theoretical overview of real-time simulation and multibody system dynamics. At the same time, the approach to hydraulics and flexible bodies simulation is introduced. This part is concluded by a short overview on best practices of tutorials creation.

The second major part of this thesis is the practical implementation, namely creation of a multibody model with flexible body in Mevea simulation software. This part covers all the practical steps required to create the flexible body model from creation of the bodies to final visualisation adjustments.

The final part includes the self-reflection about the created tutorial, combined with analysis of feedback provided by students of Saimaa University of Applied Sciences that completed this tutorial, as well as ideas for possible future development of the tutorials for Mevea software. The conclusion summarises the main achievements of this thesis.

# 2 Theoretical overview

This thesis covers the topic of flexible body model creation with a real-time simulation software, therefore it is important to understand from the beginning, what the model and real-time simulation in general are.

Model is a representation of a real-world system. The system, in turn, is a collection of items that operate and are related to each other to achieve a certain goal. This means that using the subset of objects for one study can create one system, and in case of another study, the same subset can be used to create a different system. (Averill M. Law, 2000, p. 3)

Different ways to study the system might be employed, as shown in Figure 2 below. Firstly, one can decide whether to experiment with the actual system or experiment with a model of the system. It is obvious that experimenting with the real system will be more beneficial than experimenting with a model, but this is generally not feasible, especially in a modern fast-developing reality. The costs of the complete system that failed because of the insufficient analysis behind it is much higher than the costs for a model development that can be used for a preliminary testing. At the same time, even if the slight change in design parameters is required, the system would possibly have to be completely rebuilt. Therefore, the experiment with a model is preferred.

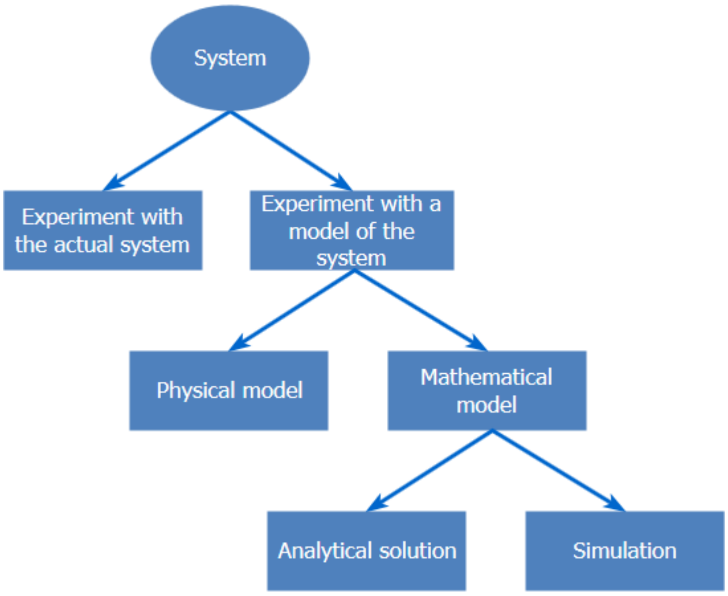


Figure 2. Ways to study a system (Averill M. Law, 2000, p. 4)

After choosing to experiment over a model rather than a real system, one still has a choice of doing a physical model or a mathematical model. Physical modelling includes, for instance, a 3D-printed model of a vehicle, a disconnected airplane cockpit used for pilot training, or miniature submarines in a swimming pool. Physical model is still a viable choice in a lot of cases, however in modern fast-developing world even the speed with which physical prototypes are produced is not enough to keep competing with companies that use mathematical modelling before creating a physical model.

The mathematical modelling, in turn, is divided into analytical modelling and simulation. In case the model is simple enough, it can be described with equations that produce an exact, analytical solution. These equations generally utilise methods of algebra, calculus and probability theory. However, sometimes the systems are highly complex to be analysed using an analytical model, therefore the means of simulation should be used instead. Simulation utilises numerical methods instead of exact solutions, thus enabling the opportunity to estimate the behaviour of complex systems that are impossible to analyse using analytical methods. (Averill M. Law, 2000, pp. 4-5)

## 2.1 Real-time simulation

One of the approaches in simulation is a real-time simulation. In the real-time simulation, as shown in Figure 3, the user provides the input to the control system that changes the parameters of the dynamics model. This model provides both the input for simulated visual, motion, sound and control forces cues, and control feedback that is returned to control system. This process works in a loop, as the user evaluates visual, motion, sound and control forces cues to change his or her actions and provide new input for the control system. (Haug & Deyo, 1991, p. 5)

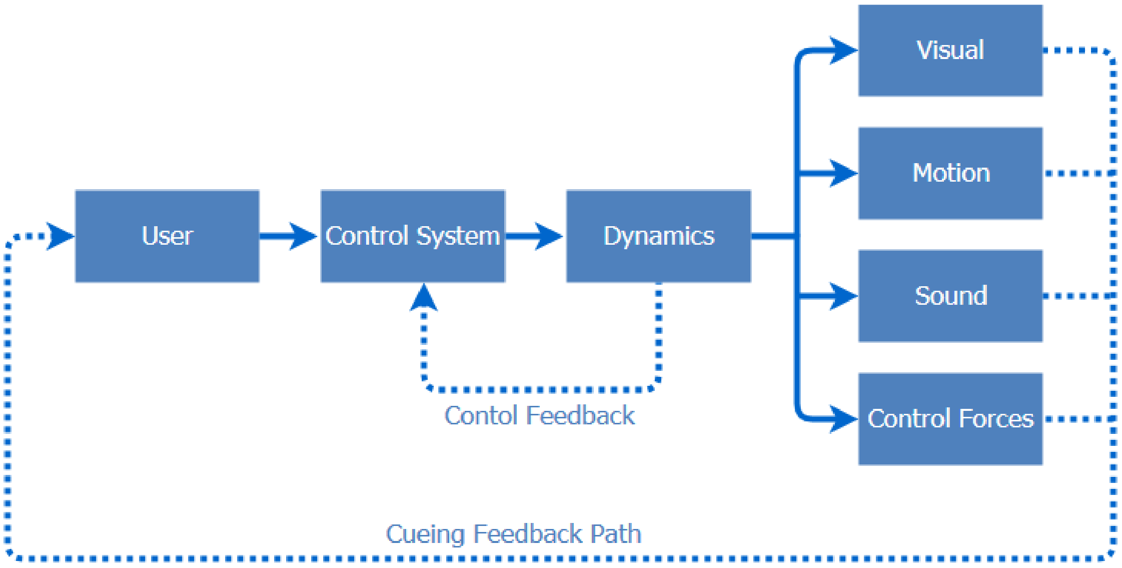


Figure 3. Simulation functional diagram (Haug & Deyo, 1991, p. 5)

The main requirements for the method used in real-time simulation are ability to formulate the mathematical model precisely and efficiently to take as little computational power as possible. To model a complete real-time system representing, for instance, a mobile vehicle, several modelling tasks are executed: contact modelling, multibody modelling, hydraulic modelling and friction modelling (Baharudin, 2016, p. 23). This thesis focuses on multibody and hydraulic modelling; therefore, it does not include theoretical overview of contact and friction modelling.

## 2.2 Multibody system dynamics

Generally, the analysis of dynamic systems is based on the Newton's laws and the derived equations of motions for particles. The particle mass, in turn, is used to represent a body as an idealised concept, that is used in simplest dynamics modelling. However, this approach is not suitable for the modelling of complex system of interconnected bodies. That is why a new approach called multibody system dynamics was developed.

### 2.2.1 Multibody systems

As the title implies, multibody system is the system that consists of multiple rigid or deformable components. The multibody system is a set of subsystems (bodies, components, or substructures), which motion is kinematically constrained due to the different types of joints between them, and each of the related subsystems may undergo large displacements and rotations (Shabana, 1998, p. 1)

However, not every system that consists of several bodies is considered a multibody system. Figure 4 presents a multibody system (a) and a “non-multibody” system (b). As it is clearly seen in the figure, the first system has mechanical joints between the bodies involved, however, the second system has only springs that connect the bodies. Both systems can be modelled using multibody approach, although FE-analysis might be more suitable in the second case (Mikkola, 2016, pp. 3-4).

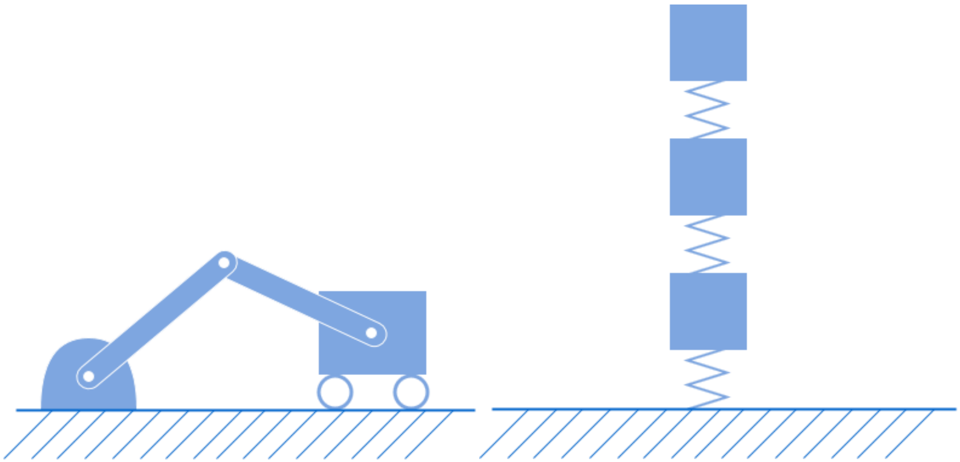

a) b)

Figure 4. a) Cam-shaft mechanism b) spring-mass system (Mikkola, 2016, p. 4)

The kinematics and dynamics of a multibody system are solved using a number of approaches. All of these approaches are based on solving the equations of motion, that are formulated based on the following methods: Lagrange multipliers method, penalty formulation, augmented Lagrangian method and semi-recursive method. At the same time, rigid and flexible bodies require different treatment regarding reference system used in these equations of motions. Rigid bodies can be described using fixed frame of reference, because the relative deformation inside those bodies is zero, while flexible bodies have to be described using floating frame of reference (Baharudin, 2016, p. 23).

All the above-mentioned solution methods have their advantages and disadvantages, that is why the user should be able to choose the appropriate method based on the accuracy that is needed and what is the purpose of modelling the concrete system. In Mevea Modeller, however, Lagrange, augmented formulation, penalty formulation, and recursive methods are available (Mevea Ltd., 2017, p. 10).

### 2.2.2 Solution of MBS using method of Lagrange multipliers

The Lagrange formulation characterizes the performance of a dynamic system in connection with work and energy accumulated in the system. The general Lagrange equation is presented in the form:

$$\boldsymbol{\tau} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\boldsymbol{q}}}\right)^T - \left(\frac{\partial L}{\partial \boldsymbol{q}}\right)^T \; (1)$$

where $\boldsymbol{\tau}$ is the vector of generalised forces applied to the body; $\boldsymbol{q}$ is the vector of generalised coordinates; $\dot{\boldsymbol{q}}$ is the vector of generalised velocities, and $L$ is a Lagragian function as in Equation 2 below. $E$ in this case is the kinetic energy of the system and $U$ is the potential one.

$$L = E - U \; (2)$$

For a rigid body, the Equation 1 yields to the form:

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}}) \; (3)$$

where $\boldsymbol{M}(\boldsymbol{q})$ is the generalised inertia matrix and $\boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is the vector responsible for Coriolis, centrifugal and gravity effects (Briot & Khalil, 2015, pp. 61-62).

For a constrained system, however, the equations of motion turn to be:

$$\begin{cases} \boldsymbol{M}\ddot{\boldsymbol{q}} - \boldsymbol{Q}_e - \boldsymbol{Q}_v + \boldsymbol{\Phi}_q^T\boldsymbol{\lambda} = \boldsymbol{0} \\ \boldsymbol{\Phi}(\boldsymbol{q},\boldsymbol{t}) = \boldsymbol{0} \end{cases} \; (4)$$

where $\ddot{\boldsymbol{q}}$ is the accelerations vector, $\boldsymbol{Q}_e$ is the vector of generalised forces, $\boldsymbol{Q}_v$ is the quadratic velocity vector, $\boldsymbol{\Phi}_q$ is the Jacobian matrix of the constraints, and $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. This is the DAE (differential algebraic equation), which is converted to ODE (ordinary differential equation) in order to be solved. This is done by differentiating the first part twice with respect to time resulting into the following:

$$\boldsymbol{\Phi}_q\ddot{\boldsymbol{q}} = -\left(\boldsymbol{\Phi}_q\dot{\boldsymbol{q}}\right)_q\dot{\boldsymbol{q}} - 2\boldsymbol{\Phi}_{qt}\dot{\boldsymbol{q}} - \boldsymbol{\Phi}_{tt} \; (5)$$

where $\mathbf{\Phi}_{qt}$ is the first derivative of Jacobian matrix of constraints, $\mathbf{\Phi}_{tt}$ is the second derivative of the vector of constraints. The term $-\left(\mathbf{\Phi}_{q}\dot{\boldsymbol{q}}\right)_{\boldsymbol{q}}\dot{\boldsymbol{q}} - 2\mathbf{\Phi}_{qt}\dot{\boldsymbol{q}}$ can be replaced with $\boldsymbol{Q}_C$ and Equation 5 is rewritten in the matrix format:

$$\begin{bmatrix} \boldsymbol{M} & \mathbf{\Phi}_q^T \\ \mathbf{\Phi}_q & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{Q}_e + \boldsymbol{Q}_v \\ \boldsymbol{Q}_C \end{bmatrix} \quad (6)$$

The drawback of this method is that because the equation is differentiated twice with respect to time, some constant components of the equations disappear. That is why penalty terms are introduced, and the first part of the Equation 4 becomes the following:

$$\boldsymbol{M}\ddot{\boldsymbol{q}} = \boldsymbol{Q}_e + \boldsymbol{Q}_v - \alpha\mathbf{\Phi}_q^T(\ddot{\mathbf{\Phi}} + \mathbf{2}\boldsymbol{\xi}\boldsymbol{\omega}_{\boldsymbol{n}}\dot{\mathbf{\Phi}} + \boldsymbol{\omega}_{\boldsymbol{n}}^2\mathbf{\Phi}) \quad (7)$$

where $\alpha$ is penalty factor, $\boldsymbol{\xi}$ is damping ratio, and $\boldsymbol{\omega}_n$ is natural frequency, all of them being diagonal matrices (Baharudin, 2016, pp. 28-29).

### 2.2.3 Recursive solution method in MBS

Penalty formulation of Lagrange multipliers method is rather computationally heavy, therefore with some assumptions made, recursive and semi-recursive methods were developed. With these methods, the kinematic properties are acquired based on the relative coordinates of the neighbouring bodies connected through a joint. The equations of motion are written in terms of the system's degrees of freedom, which results in lower dimensionality compared to Lagrangian formulation.

Let us assume two contiguous bodies in multibody system: bodies $B_{j-1}$ and $B_j$ connected with a joint, as shown in Figure 5. In this case, the position of the vector $\boldsymbol{r}_j$ can be described as:

$$\boldsymbol{r}_{\boldsymbol{j}} = \boldsymbol{R}_{\boldsymbol{j-1}}^{\boldsymbol{cm}} + \boldsymbol{A}_{\boldsymbol{j-1}}\bar{\boldsymbol{u}}_{\boldsymbol{j-1}} + \boldsymbol{d}_{\boldsymbol{j-1,j}} \quad (8)$$

where $\boldsymbol{R}_{j-1}^{cm}$ is the position vector for the centre of mass of body $B_{j-1}$, $\boldsymbol{A}_{j-1}$ is the rotation matrix of the body $B_{j-1}$, $\bar{\boldsymbol{u}}_{j-1}$ is the position vector within body CS and joint of the body $B_{j-1}$, and $\boldsymbol{d}_{j-1,j}$ is the relative displacement vector between the bodies (Baharudin, 2016, p. 31).

The Equation 8 is then differentiated twice with respect to time to obtain velocity and acceleration of the vector $\boldsymbol{r}_j$:

$$\dot{\boldsymbol{r}}_j = \dot{\boldsymbol{R}}^{cm}_{j-1} + \tilde{\boldsymbol{\omega}}_{j-1}\boldsymbol{u}_{j-1} + \dot{\boldsymbol{d}}_{j-1,j} \quad (9)$$

$$\ddot{\boldsymbol{r}}_j = \ddot{\boldsymbol{R}}^{cm}_{j-1} + \dot{\tilde{\boldsymbol{\omega}}}_{j-1}\boldsymbol{u}_{j-1} + \tilde{\boldsymbol{\omega}}_{j-1}\tilde{\boldsymbol{\omega}}_{j-1}\boldsymbol{u}_{j-1} + \ddot{\boldsymbol{d}}_{j-1,j} \quad (10)$$

where $\dot{\boldsymbol{R}}^{cm}_{j-1}$ is the velocity of the centre of mass of the body $B_{j-1}$, $\tilde{\boldsymbol{\omega}}_{j-1}$ is the skew-symmetric matrix of angular velocity of the body $B_{j-1}$, $\boldsymbol{u}_{j-1} = \boldsymbol{A}_{j-1}\boldsymbol{A}_{j-1,j}$, and $\dot{\boldsymbol{d}}_{j-1,j}$ is the relative velocity between bodies; $\ddot{\boldsymbol{R}}^{cm}_{j-1}$ is, accordingly, the acceleration of the centre of mass of the body $B_{j-1}$, $\dot{\tilde{\boldsymbol{\omega}}}_{j-1}$ is the first time derivative of the relative angular velocity skew symmetric matrix, and $\ddot{\boldsymbol{d}}_{j-1,j}$ is the relative angular acceleration between two bodies.

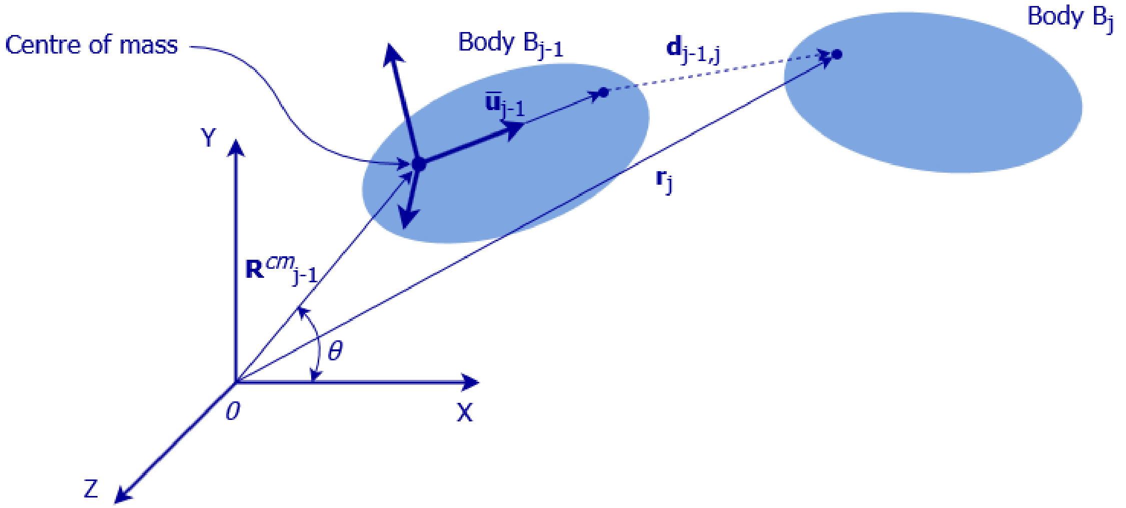


Figure 5. Recursive body position description of multibody system (Baharudin, 2016, p. 31)

At this point, Newton-Euler equations are used to formulate the equation of motion:

$$\begin{bmatrix} m_j\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_j\boldsymbol{J}_j\boldsymbol{A}_j \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{R}}^{cm}_j \\ \dot{\boldsymbol{\omega}}_j \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \tilde{\boldsymbol{\omega}}_j\boldsymbol{J}_j\boldsymbol{\omega}_{j-1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_j \\ \boldsymbol{T}_j \end{bmatrix} \quad (11)$$

or

$$\boldsymbol{M}_j\ddot{\boldsymbol{q}}_j + \boldsymbol{Q}_{vj} = \boldsymbol{Q}_{ej} \quad (12)$$

where $\boldsymbol{J}_j$ is the inertia tensor, $\dot{\boldsymbol{\omega}}_j = \dot{\boldsymbol{\omega}}_{j-1} + \dot{\boldsymbol{\omega}}_{j-1,j}$ is the angular velocity of body *B*$_j$, $\boldsymbol{F}_j$ is the vector of external forces, $\boldsymbol{T}_j$ is the vector of torsional forces, $\boldsymbol{M}_j$ is the mass matrix, $\ddot{\boldsymbol{q}}_j$ is the vector of generalised accelerations of body *B*$_j$, $\boldsymbol{Q}_{vj}$ is the quadratic velocity vector, and $\boldsymbol{Q}_{ej}$ is the vector of generalised forces.

If the complete set of generalised coordinates and constraint equations is used for formulating the Equation 12, a lot of computational power is used, especially if the system consists of a lot of interconnected bodies. Because of that, the efficiency of the method is not higher than penalty Lagrangian, but if the equations of motion are written in terms of relative (joint) coordinates, then the number of differential equations is reduced (Baharudin, 2016, pp. 31-32).

For the model developed in this thesis, recursive method was chosen. Recursive formulation is computationally more efficient, as was mentioned before; however, it restricts the available joint types to floating, spherical, universal, hinge, translational, cylindrical, fixed and inline (Mevea Ltd., 2017, p. 10) Nevertheless, these joint types are sufficient for successful modelling, and computational efficiency is more important for demo and tutorial models than ability to use more joint types.

### 2.2.4 Rigid body kinematics in MBS

An important part of multibody system modelling is defining a correct frame of reference or coordinate system. The frame of reference is represented by three orthogonal axes rigidly connected to a point called the origin of the mentioned reference frame. Multibody system modelling requires two coordinate systems: global, or inertial coordinate system and body reference coordinate system. The global coordinate system is stationary in time and defines the unique standard for all the other bodies in the system. Body reference coordinate system, in turn, is assigned to each component in the system. It rotates and translates together with the body in time, thus its location and orientation with respect to global coordinate system change in time. Figure 6 shows a global coordinate system with axes $\mathbf{X}_1$, $\mathbf{X}_2$ and $\mathbf{X}_3$, and a body *i* with respective body reference coordinate system. (Shabana, 1998, pp. 4-5)

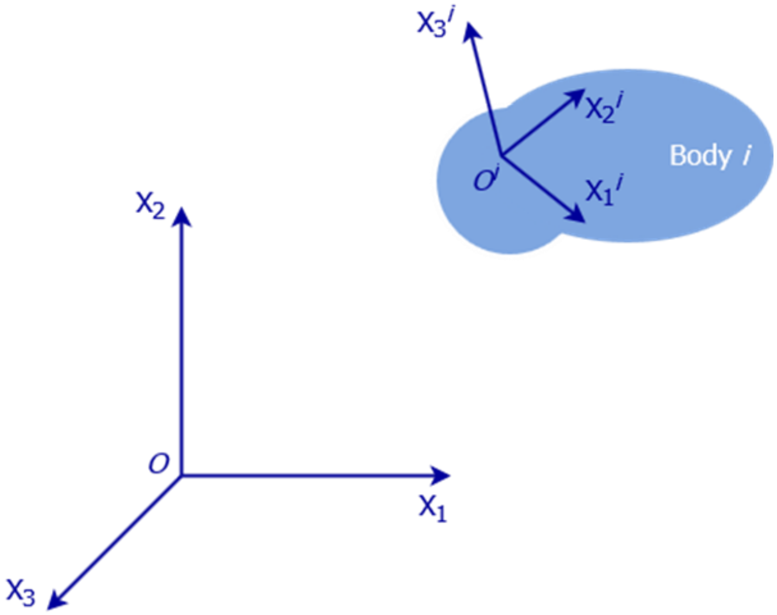


Figure 6. Global and body reference coordinate systems (Shabana, 1998, p. 5)

The mechanics for rigid bodies in multibody dynamics are done with respect to body reference coordinate system. The configuration of any arbitrary rigid body can be described using six coordinates: three coordinates for the translation and three coordinates for orientation. Considering the body as shown in Figure 7, the position of point $P_i$ within this body can be defined by the Equation 13:

$$\boldsymbol{r^i} = \boldsymbol{R^i} + \boldsymbol{u^i} \qquad (13)$$

where $\boldsymbol{r}^i = [r_1^i \quad r_2^i \quad r_3^i]^T$ is the global position of the point $P^i$, $\boldsymbol{R}^i = [R_1^i \quad R_2^i \quad R_3^i]^T$ is the global position of the body reference system origin $O^i$, and $\boldsymbol{u}^i = [u_1^i \quad u_2^i \quad u_3^i]^T$ is the position vector of point $P^i$ with respect to body reference system origin $O^i$ (Shabana, 1998, p. 11).

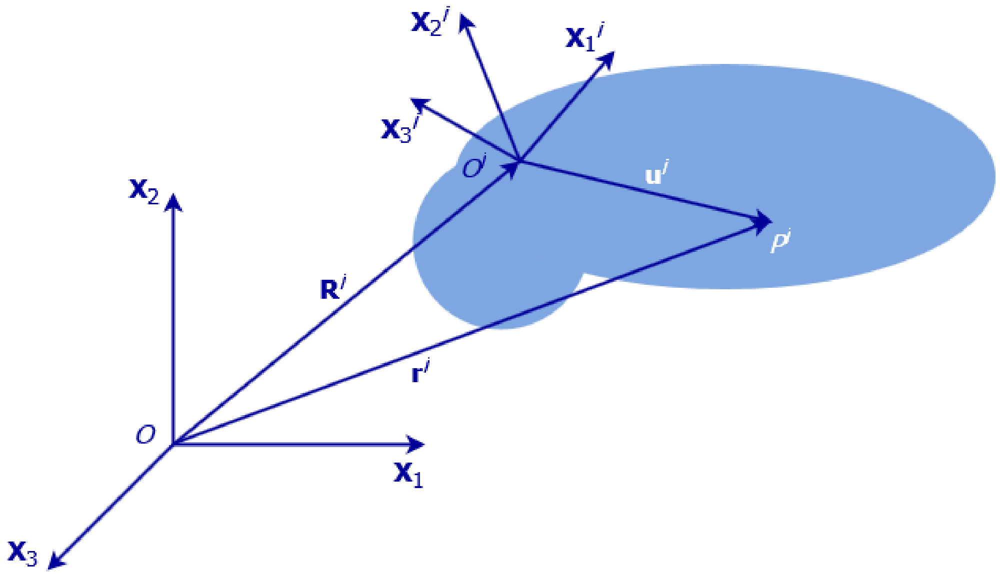


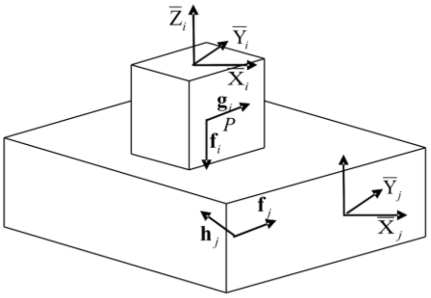
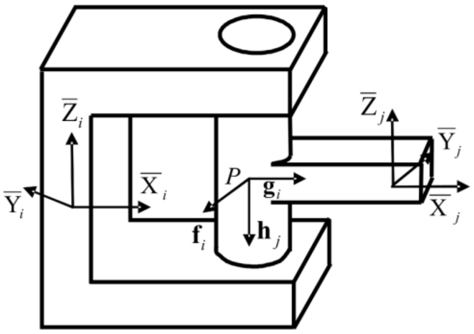
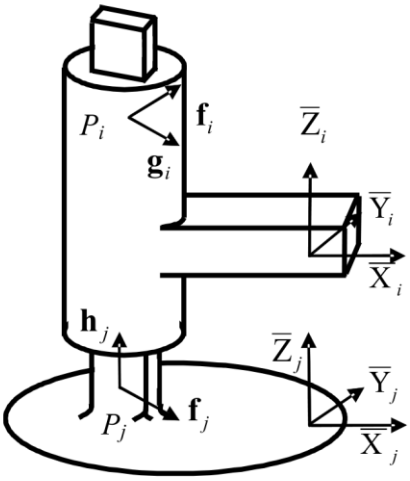
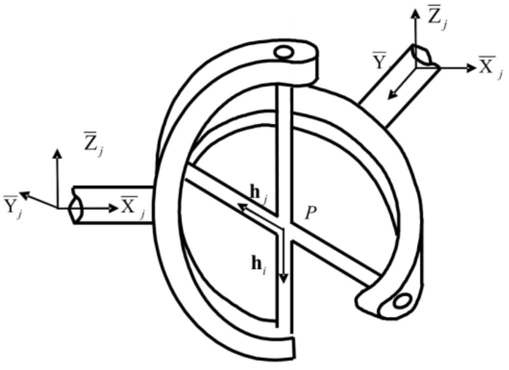
Figure 7. Rigid body mechanics (Shabana, 1998, p. 11)

### 2.2.5 Degrees of freedom and joints

The motion of the bodies in multibody systems is restricted to a certain point by mechanical joints (e.g. translational, revolute, cylindrical) between the bodies. Because six coordinates are required to describe the configuration of a rigid body in space, $6 \times n_b$ coordinates are required to describe the motion of $n_b$ unconstrained bodies in space. Mechanical joints, however, reduce the possible mobility of the bodies in the system, as they make the motion of the system not independent. Mechanical joints can be described mathematically, using a set of non-linear algebraic constraint equations. Knowing the amount of system coordinates and the constraint equations, one can define the number of degrees of freedom of the system according to Kutzbach criterion: the amount of system coordinates minus the amount of constraint equations, as shown in Equation 14 (Shabana, 1998, p. 19):

$$DOF = 6 \times n_b - n_c \ (14)$$

Mevea Modeller uses the following joint types: floating, spherical, universal, translational, cylindrical, fixed, hinge, inplane, inline, perpendicular, spherical-spherical, revolute-spherical and revolute-revolute, however only the first six are presented in Table 1 as the most relevant ones. The table shows the pictures of the joints, number of degrees of freedom and the translational or rotational movement not constrained by this joint.

| **Name** | **Figure** | **DOF** | **Movement permitted** |
|---|---|---|---|
| Floating | - | 6 | Translation and rotation around all axis |
| Fixed |  | 0 | Translation and rotation around all axes not permitted |
| Revolute |  | 1 | Rotation around axis $\boldsymbol{h}_j$ |
| Translational |  | 1 | Translation around axis $\boldsymbol{h}_j$ |
| Universal |  | 2 | Rotation around axes $\boldsymbol{h}_j$ and $\boldsymbol{h}_i$ |

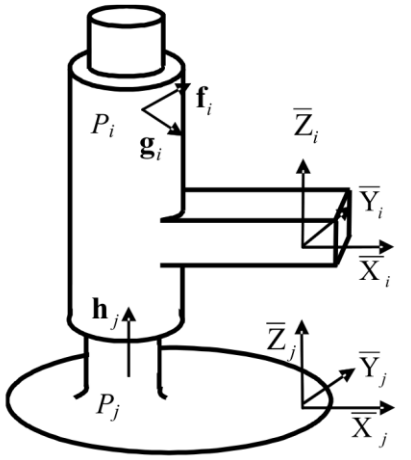
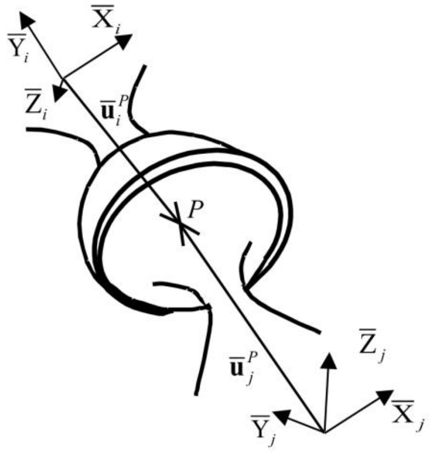
| | | | |
|---|---|---|---|
| Cylindri-cal |  | 2 | Translation and rotation around axis $\boldsymbol{h_j}$ |
| Spheri-cal |  | 3 | Rotation around all three axes |

Table 1. Summary of joint types (Mevea Ltd., 2017, pp. 20-27)

### 2.3 Modelling of hydraulics in real-time environment

Hydraulic systems play important role in mechanical engineering currently, as they allow generation of large forces with rather compact equipment and acceptable, for most cases, control frequencies. However, modelling of hydraulics is still quite computationally heavy task, as the behaviour of all components have to be accounted, and especially fluid compressibility in them (Pfeiffer, 2008, p. 187).

Due to complexity of hydraulic components, the separate parts of each single component are not modelled, but represented as just one single component, “black box”. These “black boxes” or hydraulic volumes are assumed to have the constant pressure inside them. This approach is called lumped fluid theory: it assumes that the all system oscillations are significantly less than the ones caused by wave propagation. In lumped fluid theory, the interconnecting volume then takes into account fluid compressibility, as well as fluid mass effects in order to acquire the required dynamic terms. The simple hydraulic system in general terms and in terms of lumped fluid theory is presented in Figure 8.

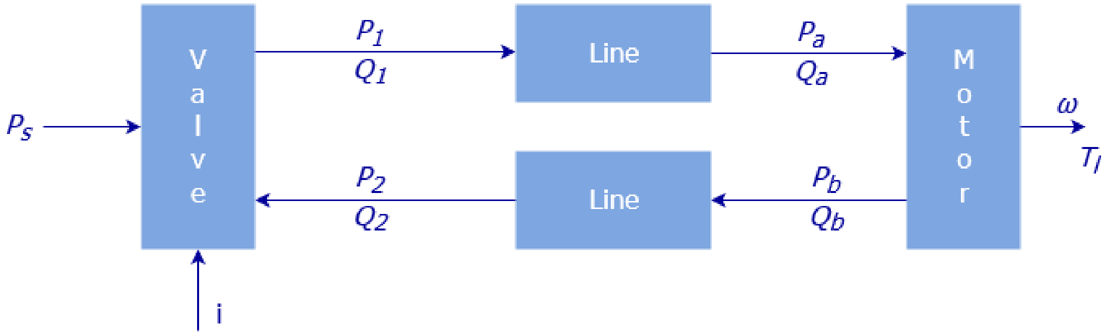


Figure 8. System according to lumped fluid theory (Watton, 1989, pp. 96-97)

Therefore, the model of hydraulics, written in simple terms, for example for motor in Figure 8, looks like:

$$[Q_a - Q_b] = \text{steady-state flow losses} + \text{"a time-dependent dynamic term"}\ (15)$$

$$[P_a - P_b] = \text{steady-state pressure losses} + \text{"a time-dependent dynamic term"}\ (16)$$

Hydraulic systems use hydraulic fluids as mediums, which have their own parameters to be taken care of. Two most important parameters for modelling are viscosity and bulk modulus (Watton, 1989, p. 96)

### 2.3.1 Viscosity

Viscosity is a measure that describes a fluid’s resistance to flow; the internal friction of the moving fluid. Larger viscosity occurs in the fluids which give a lot of internal friction due to their molecular makeup (Princeton University, 1998).

As viscosity is associated with friction, it can be described using Newtonian shear stress equation:

$$\tau = \eta \frac{dv}{dy}\ (17)$$

where $\tau$ is the shear stress, $\eta$ is kinematic viscosity, and terms $dv$ and $dy$ describe the change of speed and location of the fluid. However, in practical modelling, kinematic viscosity is replaced with dynamic viscosity:

$$\nu = \frac{\eta}{\rho} \ (18)$$

where $\rho$ is the density of the fluid. Kinematic viscosity is expressed in centistokes ($1cS = 10^{-6} m^2/s$) and is experimental data provided by manufacturer of hydraulic fluid. It highly depends on the temperature and operating pressure, as seen in Figure 9 (Watton, 1989, pp. 26-27).

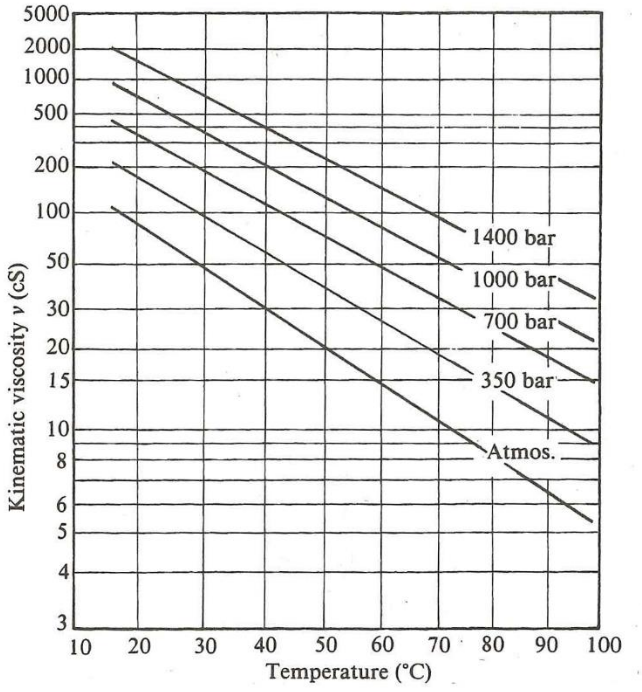


Figure 9. Kinematic viscosity dependence on pressure and temperature (Watton, 1989, p. 27)

### 2.3.2 Bulk modulus

Bulk modulus describes the compressibility of the fluid and is required to take into account the undamped frequencies of the hydraulic system. Different parameters affect bulk modulus; therefore, its calculation is rather difficult, but crucial for developing a reliable model (Watton, 1989, p. 28).

Let us assume a rigid container with two volumes $V_1$ and $V_2$ connected to one another as in Figure 10. The fluid in both containers is the same and has the bulk modulus of $B_0$.

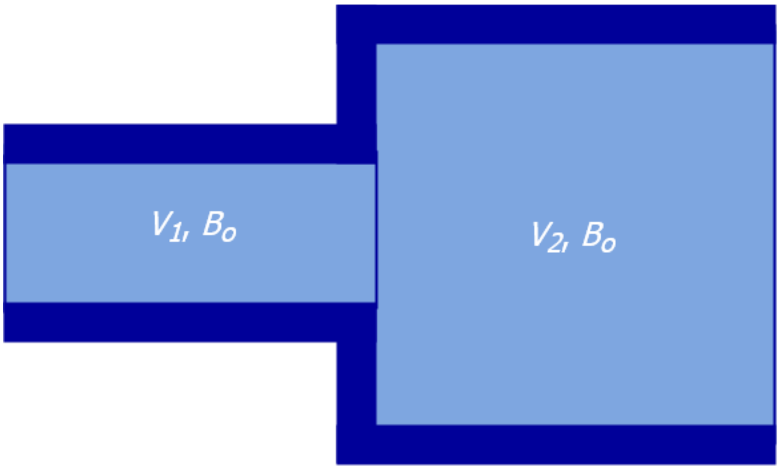


Figure 10. Rigid container with two volumes

Then the total volume of fluid in the container is calculated by the following equation:

$$V_o = V_1 + V_2 = V_{c1} + V_{c2} \ (19)$$

where $V_{c1}$ and $V_{c2}$ are volume of the first and the second container respectively. The volume change is therefore expressed as:

$$dV_t = -dV_1 - dV_2 + dV_{c1} + dV_{c2} \ (20)$$

Plugging in general equation for bulk modulus ($B = -V\frac{dP}{dV}$) into Equation 20, the following equation is obtained:

$$\frac{V_t dP}{B_e} = \frac{V_1 dP}{B_o} + \frac{V_2 dP}{B_o} + \frac{V_{c1} dP}{B_{c1}} + \frac{V_{c2} dP}{B_{c2}} \ (21)$$

where $B_e$ is the effective bulk modulus, $B_{c1}$ and $B_{c2}$ are the bulk moduli of the first and the second container. Dividing this equation by $V_t dP$, it yields to:

$$\frac{1}{B_e} = \frac{V_1}{V_t B_o} + \frac{V_2}{V_t B_o} + \frac{V_{c1}}{V_t B_{c1}} + \frac{V_{c2}}{V_t B_{c2}} \ (22)$$

One can notice that $V_t = V_1 + V_2$, $V_1 \cong V_{c1}$ and $V_2 \cong V_{c2}$, therefore the final form of effective bulk modulus equation is:

$$\frac{1}{B_e} = \frac{1}{B_o} + \frac{V_1}{V_t B_{c1}} + \frac{V_2}{V_t B_{c2}} \ (23)$$

Bulk modulus of metallic container is calculated with the following formula:

$$B_c \cong \frac{tE}{d} \quad (24)$$

where $t$ is the wall thickness, $E$ is the elastic modulus of the material and $d$ is the diameter of the volume in question.

Equation 23 might be expanded to have terms for additional volumes and effect of dissolved air, however, as some researchers suggest, realistic values of dissolved and entertained air do not dramatically reduce the value of effective bulk modulus (Watton, 1989, pp. 29-34).

### 2.3.3 Flow continuity equation

One of the most important equation in lumped fluid theory is flow continuity equation, that is based on mass flow rate continuity. Assuming a fluid volume with input mass flow rate $m_i$ and output mass flow rate $m_o$, the following equation is derived:

$$\rho_i Q_i - \rho_O Q_O = \frac{\mathrm{d}}{\mathrm{d}t}(\rho V) \quad (25)$$

where $\rho_i$ is the density of fluid at input and $\rho_O$ – at output; $Q_i$ is the input flow rate and $Q_O$ – output; $\rho$ and $V$ are density and volume of fluid inside a hydraulic volume. Replacing the input and output densities with mean density and assuming the simplified bulk modulus formulation, the following equation is obtained:

$$Q_i - Q_O = \frac{\mathrm{d}V}{\mathrm{d}t} + \frac{V}{B_e}\frac{\mathrm{d}P}{\mathrm{d}t} \quad (26)$$

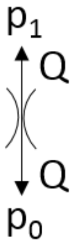
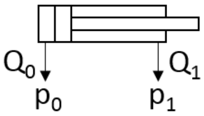
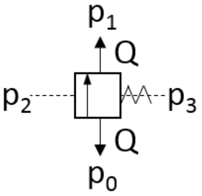
where $P$ is the pressure. This is the equation of flow continuity that is later applied to all components analysed using lumped fluid theory (Watton, 1989, pp. 97-98).

From Equation 26, the differential equation for hydraulic pressure in any volume can be derived (Baharudin, 2016, p. 39):

$$\dot{p} = \frac{B_e}{V}\left(Q_i - Q_o - \dot{V}\right) \quad (27)$$

### 2.3.4 Description of selected hydraulic components

Hydraulic components in MBS are described using differential and algebraical equations, that are partially based on experimental data. The equations for most common hydraulic components were developed, and Table 2 includes summary of them: name, hydraulic symbol, related equations and related values (as used in Mevea Modeller).

| **Component name** | **Schematic symbol** | **Related equations** | **Related values** |
|---|---|---|---|
| Throttle |  | $Q =$<br>$UC_v\sqrt{\lvert p_0 - p_1\rvert}$<br>$STEP(p_0 - p_1$<br>$- 1e3, 1e3, -1{,}1$<br>$C_v = \frac{Q_{nom}/60000}{\sqrt{dp_{nom} \times 1e5}}$ | $Q$ – throttle flow rate<br>$U$ – relative throttle opening<br>$d$ – throttle diameter<br>$C_v$ – flow rate constant of the throttle |
| Hose/pipe |  | $V_k = \frac{\pi D^2 L}{4}$<br>$B_k$<br>$= MIN\left(500e6,\right.$<br>$\left.MAX\left(Be_{hk}\sqrt{p_i}, 70e6\right)\right)$ | $V_k$ – constant volume of a hose/pipe<br>$D$ – diameter of a pipe<br>$L$ – length of a hose<br>$B_k$ – bulk modulus for a selected hose<br>$Be_{hk}$ – bulk modulus value typical for a hose<br>$p_i$ – pressure in the volume $i$, to which the hose is connected to |
| Double-acting cylinder |  | $Q_{jA} = -\dot{x}A_A \; Q_{jB} =$<br>$\dot{x}A_B$<br>$F = p_0A_A - p_1A_B - F_\mu$<br>$F_\mu = (p_0A_A - p_1A_B)$<br>$(1 - \eta)f(\dot{x})$ | $Q_{jA}$ and $Q_{jB}$ – flow rate produced for the hydraulic volume $i$<br>$F$ – force produced by hydraulic cylinder<br>$F_\mu$ – total friction of the cylinder<br>$\eta$ – cylinder friction coefficient |
| Pressure valves |  | $Q = K\sqrt{(\lvert p_0 - p_1\rvert)}$<br>$STEP(p_0 - p_1, -1e3,$<br>$1e3, -1, 1)$ | $K$ – relative spool position |

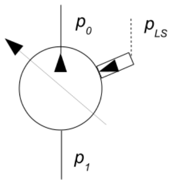
| | | | |
|---|---|---|---|
| Pump |  | $M_{load} = \frac{V_{rot}\lvert\alpha\rvert(p_0 - p_1)i}{\nu_m \nu_v}$<br>$Q = V_{rot}\omega i \alpha \nu_v$ | $M_{load}$ – torque passed to the motor<br>$V_{rot}$ – radial displacement of the pump<br>$\alpha$ – pump relative volume<br>$i$ – ratio between motor and pump<br>$\nu_m$ – mechanical coefficient of the pump<br>$\nu_v$ – volumetric coefficient of the pump<br>$\omega$ – angular velocity of the motor |

Table 2. Selected hydraulic components and associated parameters (Mevea Ltd., 2017, pp. 57-91)

## 2.4 Modelling of flexible bodies in MBS

In rigid body kinematics, no distinction between the kinematics of the body and the kinematics of its reference frame is made, therefore all the movements of the body are described using just one reference frame. This is possible due to the fact that in rigid body the distance between two points inside the body does not change with time. However, in flexible bodies two arbitrary points move relatively to each other, therefore one coordinate system is not enough to describe the behaviour of that body; theoretically an infinite number of coordinate systems is required to properly describe the motion of such a body (Shabana, 1998, p. 15).

### 2.4.1 Solution of connector forces

One of the methods that is used in modelling of flexible bodies in MBS is finite segments method. In this case, the flexible body is divided into discrete mass points that are connected with a spring-damper components. These mass points are treated as rigid bodies, which allows the usage of rigid bodies equations of motion (Sopanen, 2004, p. 19).

Connector forces are the ones that describe the spring-damper elements between two mass points. Assuming two different segments *i* and *j* in a flexible body, the local forces $\bar{\boldsymbol{F}}^{ij}$ and local torques $\bar{\boldsymbol{T}}^{ij}$ can be described using the following equation:

$$\begin{bmatrix}\bar{\boldsymbol{F}}^{ij}\\ \bar{\boldsymbol{T}}^{ij}\end{bmatrix} = -\boldsymbol{K_b}\begin{bmatrix}\boldsymbol{d_b}\\ \boldsymbol{\theta_b}\end{bmatrix} - \boldsymbol{D_b}\begin{bmatrix}\dot{\boldsymbol{d}}_{\boldsymbol{b}}\\ \dot{\boldsymbol{\theta}}_{\boldsymbol{b}}\end{bmatrix} \; (28)$$

where $\boldsymbol{d}_b = [d_x - l \quad d_y \quad d_z]^T$ and $\boldsymbol{\theta}_b = [\theta_x \quad \theta_y \quad \theta_z]^T$ are relative translation and rotation of segment *i* about body *j*, $\boldsymbol{K}_b$ is the equivalent stiffness matrix and $\boldsymbol{D}_b$ is the damping matrix that can be evaluated by multiplying equivalent stiffness matrix by the damping ratio. The stiffness matrix is defined as below:

$$\boldsymbol{K_b} = \begin{bmatrix} \frac{EA}{l} & 0 & 0 & 0 & 0 & 0 \\ & \frac{12EI_{zz}}{l^3(1+\phi_y)} & 0 & 0 & 0 & \frac{6EI_{zz}}{l^2(1+\phi_y)} \\ & & \frac{12EI_{yy}}{l^3(1+\phi_z)} & 0 & \frac{6EI_{yy}}{l^2(1+\phi_z)} & 0 \\ & & & \frac{GJ}{l} & 0 & 0 \\ & symm. & & & \frac{(4+\phi_z)EI_{yy}}{l(1+\phi_z)} & 0 \\ & & & & & \frac{(4+\phi_y)EI_{zz}}{l(1+\phi_y)} \end{bmatrix} \; (29)$$

where $E$ is the material's modulus of elasticity, $A$ is the cross-section area of the beam, $l$ is the undeformed length of the beam, $I_{yy}$ and $I_{zz}$ are the second and third moments of inertia, $\phi_y = \frac{12E\,I_{yy}}{k_{sy}GAl^2}$, $\phi_z = \frac{12EI_{zz}}{k_{sz}GAl^2}$, $G$ is the material's modulus of rigidity, $J$ is the torsion constant, $k_{sy}$ and $k_{sz}$ are the shear correction factors (Sopanen, 2004, pp. 20, 92).

After that, the equations for forces that act on a body *j* due to interaction with body *i* can be written as follows:

$$\boldsymbol{F}^{ji} = -\boldsymbol{F}^{ij} \; (30)$$

$$\boldsymbol{T}^{ji} = -\boldsymbol{T}^{ij} - \boldsymbol{I} \times \boldsymbol{F}^{ij} \; (31)$$

where $\boldsymbol{I}$ is the vector from body *j* to body *i*. Based on these two equations, the vectors of generalised forces can be formulated:

$$(\boldsymbol{Q}_{\boldsymbol{e}}^{\boldsymbol{i}})_{\boldsymbol{R}} = \boldsymbol{A}^{\boldsymbol{i}}\boldsymbol{F}^{\boldsymbol{ij}} \; (32)$$

$$(\boldsymbol{Q}_e^i)_\theta = (\boldsymbol{A}^i\boldsymbol{G}^i)^T\left(\boldsymbol{A}^i\tilde{\boldsymbol{u}}_P{}^i\boldsymbol{A}^i\boldsymbol{F}^{ij} + \boldsymbol{A}^i\boldsymbol{T}^{ij}\right) \quad (33)$$

(Sopanen, 2004, pp. 20-21)

### 2.4.2 Floating frame of reference

Floating frame of reference consists of two sets of coordinates: reference and elastic. First part shows the location and orientation of a chosen body reference. The second part, on the other hand, shows the deformation of the body with respect to the same body reference. The global position of an arbitrary point on a flexible body is then described using a coupled set of reference and elastic coordinates. This way of formulation increases the computational burden, therefore different approximation methods such as Rayleigh-Ritz, FE and finite segments are employed (Shabana, 1998, p. 191).

The definition of floating frame of reference in an arbitrary flexible body *i* is represented in Figure 11. As in rigid body formulation, the body reference system $\boldsymbol{X}^i{}_1\boldsymbol{X}^i{}_2\boldsymbol{X}^i{}_3$, position vector $\boldsymbol{R}^i$ of point $O^i$, position vector $\boldsymbol{u}^i$ of point $\boldsymbol{p}^i$ and vector $\boldsymbol{r}^i{}_P$ are defined to describe the global position of point $P^i$.

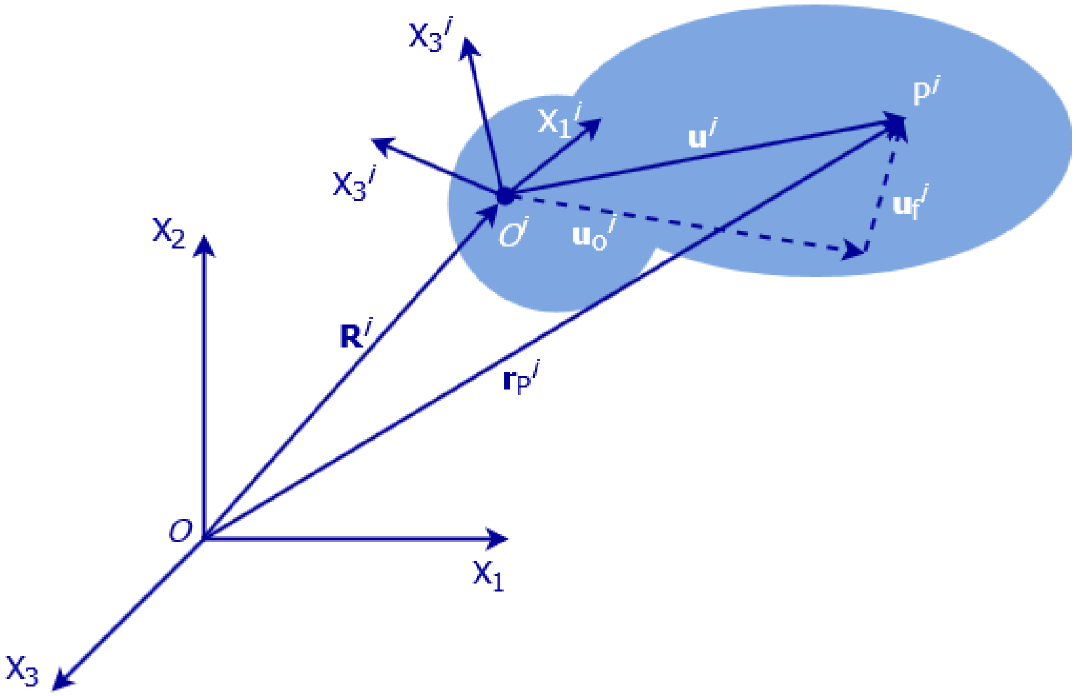


Figure 11. Deformable body coordinates (Shabana, 1998, p. 16)

Taking into account notations used in the Figure 11, the position of an arbitrary point on deformable body can be described using the following equation:

$$\boldsymbol{r^i} = \boldsymbol{R^i} + \boldsymbol{u_O^i} + \boldsymbol{u_f^i} \quad (34)$$

where $\boldsymbol{r}^i = [r_1^i \quad r_2^i \quad r_3^i]^T$ is the global position of the point $P^i$, $\boldsymbol{R}^i = [R_1^i \quad R_2^i \quad R_3^i]^T$ is the global position of the body reference system origin $O^i$, $\boldsymbol{u_O^i}$ is the undeformed local position of point $P^i$ and is constant in time, $\boldsymbol{u_f^i}$ is deformation vector, which is time- and space-dependent (Shabana, 1998, pp. 15-16).

Equation 34 is differentiated once with respect to time to obtain the velocities vector $\dot{\boldsymbol{r}}^i$:

$$\dot{\boldsymbol{r}}^{\boldsymbol{i}} = \dot{\boldsymbol{R}}^{\boldsymbol{i}} + \dot{\boldsymbol{A}}^{\boldsymbol{\iota}}\boldsymbol{u^i} + \boldsymbol{A^i}\dot{\boldsymbol{u}}^{\boldsymbol{i}} \quad (35)$$

where $\dot{R}^i$ is the velocity of the global position of the body reference system, $\boldsymbol{A}^i$ is the rotation matrix and $\dot{\boldsymbol{A}}^i$ is its derivative, $\boldsymbol{u^i} = \boldsymbol{u}_o^i + \boldsymbol{u}_f^i$ and $\dot{u}^{\boldsymbol{i}}$ is derivative of the respective function. Applying the formula $\dot{\boldsymbol{A}}^{\iota}\boldsymbol{u^i} = -\dot{\boldsymbol{A}}^{\iota}\widetilde{\boldsymbol{u}}^{\boldsymbol{i}}\boldsymbol{G}^i\dot{\boldsymbol{\theta}}^i$ and $\dot{\boldsymbol{u}}^{\boldsymbol{i}} = \boldsymbol{\Phi}\dot{\boldsymbol{p}}$, the following equation for the velocity in partitioned formed can be written:

$$\dot{\boldsymbol{r}}^{\boldsymbol{i}} = [\boldsymbol{I} \quad -\boldsymbol{A}^i\widetilde{\boldsymbol{u}}^{\boldsymbol{i}}\boldsymbol{G}^i \quad \boldsymbol{A}^i\boldsymbol{\Phi}^{\boldsymbol{i}}]\begin{bmatrix}\dot{\boldsymbol{R}}^{\boldsymbol{i}} \\ \dot{\boldsymbol{\theta}}^i \\ \dot{\boldsymbol{p}}^i\end{bmatrix} \quad (36)$$

where $\boldsymbol{I}$ is the 3 x 3 identity matrix, $\widetilde{\boldsymbol{u}}^{\boldsymbol{i}}$ is the skew-symmetric identity matrix of vector $\boldsymbol{u^i}$, $\boldsymbol{G}^i$ is the matrix that defines the relationship between the angular velocities in the local body frame and the time derivatives of the orientation coordinates, $\boldsymbol{\Phi}^i$ is the shape matrix, $\dot{\boldsymbol{\theta}}^i$ is the velocity of generalised orientation coordinates and $\dot{\boldsymbol{p}}^i$ is the derivative of the vector of elastic coordinates (Sopanen, 2004, pp. 23-24).

### 2.4.3 Solution of the shape matrix

The shape matrix $\boldsymbol{\Phi}^i$ that was mentioned in previous section is solved based on FE shape functions. However, the usage of this method may result in very complex models containing a big number of nodal variables. Therefore, the analysis based on the whole structure is not performed, but instead methods like component synthesis are used. In this method, the vibration modes are solved from the FE model. The vibration modes then are defined as shown:

$$(\boldsymbol{K_s} - \omega_k^2 \boldsymbol{M_s})\boldsymbol{\varphi_k} = \boldsymbol{0} \; (37)$$

where $\boldsymbol{K}_s$ is the stiffness matrix, $\omega_k^2$ is the $k$th eigenvalue, $\boldsymbol{M}_s$ is mass matrix and $\boldsymbol{\varphi}_k$ is the $k$th eigenvector. In this case, the dimensionality of the matrix is reduced, because only $n_m$ modes remain in the model compared to $n_f$ amount of total mode shapes. $n_m$ is always smaller than $n_f$. The vibrations are generally solved using an unconstrained system, which can cause problems with description of local deformations, that are caught using additional static correction nodes.

A Craig-Bampton method is used to substructure the stiffness and mass matrices, in which the system is separated into interior and exterior degrees of freedom. Stiffness and mass matrices in partitioned form, are thus written as follows:

$$\boldsymbol{K_s} = \begin{bmatrix} \boldsymbol{K}^{BB} & \boldsymbol{K}^{BI} \\ \boldsymbol{K}^{IB} & \boldsymbol{K}^{II} \end{bmatrix} \; (38)$$

$$\boldsymbol{M_s} = \begin{bmatrix} \boldsymbol{M}^{BB} & 0 \\ 0 & \boldsymbol{M}^{II} \end{bmatrix} \; (39)$$

where *I* and *B* superscripts refer to interior and boundary.

The constraint modes, in this case static correction modes mentioned above, are the mode shapes of interior degrees of freedom because of the successive unit displacement of the boundary degrees of freedom. The static force equilibrium, when all the forces at the interior degrees of freedom ($\boldsymbol{F}^I$) are set to zero, is used to obtain the constraint modes, as in the equations below:

$$\begin{bmatrix} \boldsymbol{F^B} \\ \boldsymbol{F^I} \end{bmatrix} = \begin{bmatrix} \boldsymbol{K^{BB}} & \boldsymbol{K^{BI}} \\ \boldsymbol{K^{IB}} & \boldsymbol{K^{II}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta^B} \\ \boldsymbol{\delta^I} \end{bmatrix} \; (40)$$

$$\boldsymbol{\delta^I} = -(\boldsymbol{K^{II}})^{-1} \boldsymbol{K^{IB}} \boldsymbol{\delta^B} = \boldsymbol{\Phi}^{\mathrm{C}} \boldsymbol{\delta^B} \; (41)$$

where $\boldsymbol{\delta}^B$ and $\boldsymbol{\delta}^I$ are the physical displacements of the boundary and interior nodes; $\boldsymbol{\Phi}^{\mathrm{C}}$ is the matrix of the constraint nodes.

The relation between the initial, physical, coordinates and the final, modal coordinates $\hat{\boldsymbol{p}}$ is then described as:

$$\boldsymbol{u_f} = \begin{bmatrix} \boldsymbol{\delta^B} \\ \boldsymbol{\delta^I} \end{bmatrix} \cong \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{\Phi}^{\mathrm{C}} & \boldsymbol{\Phi}^{\mathrm{N}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\hat{p}^C} \\ \boldsymbol{\hat{p}^N} \end{bmatrix} = \boldsymbol{\hat{\Phi}\hat{p}} \ (42)$$

where $\boldsymbol{\Phi}^{\mathrm{N}} = [\varphi_1^N \quad \dots \quad \varphi_{n_m}^N]$. Plugging $\boldsymbol{\hat{\Phi}}$ – generalised mode shape matrix – into stiffness and mass matrices, the generalised stiffness and mass matrices are obtained:

$$\boldsymbol{\hat{K}} = \boldsymbol{\hat{\Phi}^T K_s \hat{\Phi}} \ (43)$$

$$\boldsymbol{\hat{M}} = \boldsymbol{\hat{\Phi}^T M_s \hat{\Phi}} \ (44)$$

These matrices are the Craig-Bampton representation of the original structure, however Craig-Bampton modes cannot be used directly with the floating frame of reference formulation. This happens due to the fact that these modes contain rigid body modes, while the vibration of the constraints can be described using different reference frames. This problem is solved by orthogonalization: evaluating the eigenvalues from the Craig-Bampton representation as shown below:

$$\left[\boldsymbol{\hat{K}} - \hat{\omega}_k^2 \boldsymbol{\hat{M}}\right]\boldsymbol{n_k} = \boldsymbol{0} \ (45)$$

The final mode shapes and physical degrees of freedom are found as below:

$$\boldsymbol{\Phi} = \boldsymbol{\hat{\Phi} N} \ (46)$$

$$\boldsymbol{u_f} \cong \boldsymbol{\Phi}\mathbf{p} \ (47)$$

where $\boldsymbol{N} = [n_1 \quad \cdots \quad n_{n_{CB}}]$ and $n_{n_{CB}}$ is the number of Craig-Bampton nodes. In the end, the stiffness and mass matrices become diagonal as shown:

$$\boldsymbol{K_p} = \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\hat{K}\Phi} = \mathrm{diag}\left(\hat{\omega}_1^2, \dots, \hat{\omega}_{n_{CB}}^2\right) \ (48)$$

$$\boldsymbol{M_p} = \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\hat{M}\Phi} = \mathbf{I} \ (49)$$

When this procedure is performed, all of the modes have a frequency associated with the order of their frequency contribution to the system’s dynamics. However, at the same time, the physical meaning of the correction modes is lost (Sopanen, 2004, pp. 24-27).

All the equations derived in Section 2.4 are contributing to the final form of the system’s equations of motion, that is in general form the same as in rigid body case – as was pointed in the beginning of the section. However, the formulations behind the terms in these equations of motions are different and quite cumbersome, as shown in Sopanen, 2004, pp. 27-31.

### 2.5 Creation of software tutorials

Everybody encounters tutorials every once in a while, and there are certain similarities in them that make the tutorials easy to understand, follow, and replicate the result later when needed. Different sources advise different approaches on tutorial writing, but the most important steps can be identified as follows:

1. Choose a good topic, avoid too wide definitions.
2. Research the relevance of the chosen topic.
3. Create a clear outline and structure.
4. Write the steps around the created outline.
5. Illustrate the steps with screenshots and/or pictures.
6. Proofread and refine the language and structure if needed.
7. Test the tutorial with another person who have not seen it before.

(Quach, 2008), (Southpaw, 2014), (Meylah, 2009).

It is also important to take the following factors into account: a tutorial should have a narrative base that includes motivation and structure. At every step of the tutorial the student should understand why the step is needed and what will be achieved in the end. Repetitions and internal references should be included in the tutorial text: this way the student maintains the interest in the tutorial and does not get bored if the tutorial is too easy or too difficult. Concrete examples are needed for illustrating the steps, as human brain works better by coming from concrete to abstract, but not the other way around. Language of a tutorial should be clear and understandable, but it should not affect the flow of narrative (Hartl, 2015).

## 3 Tutorial creation

The final goal of this thesis was to create a tutorial that would benefit Mevea as a company. The tutorial, therefore, covers only the following topics: adding flexibility to the existing body in the model and customization of Mevea Solver window. However, the major part of the tutorial creation process was devoted to model creation that is why the process is described below.

### 3.1 Creation of the initial tutorial model

The tutorial starts with the initial completely functioning model that contains only rigid bodies. The student can try this model out and see how it behaves, so that it is easier to compare the changes in behaviour after adding flexibility. The initial model is depicted in Figure 12.

Figure 12. Initial model in tutorial

This model was created in the following steps:

1. Solution method choice
2. Creation of graphics
3. Bodies
4. Constraints
5. Forces
6. Dummies
7. Collisions
8. Hydraulics
9. Inputs

These steps are described in detail below.

### 3.1.1 Solution method choice

As was already mentioned in Section 2.2, there are four different methods used in solution of multibody dynamics. Mevea software offers Lagrange, augmented formulation, penalty formulation and recursive methods. For this model, recursive method was chosen, because it offers more computational efficiency, even though it does not support some joint types that are not used in this model.

### 3.1.2 Creation of graphics

As Mevea Modeller focuses more on real-time simulation rather than 3D modelling, it allows the user to import graphics from several different 3D/CAD modelling softwares. Mevea Modeller supports the following file formats: .3ds, .fbx, .obj, .stl, .dae, .osg, .osgt and .osgb. All of these file formats can be used to add the graphics to the bodies, however only .3ds, .stl and .dae are suitable for collision graphics (Mevea Ltd., 2017, p. 33).

Blender – free open source 3D computer graphics software – was used to create the models needed for this thesis. The main advantages of Blender as the software for graphics creation is that it is free to use for any purpose and supports import to numerous file formats, including .obj, .stl and .3ds that were used in the process of model creation for this thesis (Blender Foundation, 2017).

The models were exported as .obj files for usage in OpenSceneGraph, the open source middle-ware used by application developers in various simulation applications (OpenSceneGraph, 2017). In OpenSceneGraph, files were exported as .osgt. The models exported in .stl (stereolithography) format were used in Solidworks to create rough models for mass properties evaluation. Finally, .3ds files were used for collision graphics definition.

After all the files were created, the modelling in Mevea started. The graphics is added into respective part of the *Model* tree in Mevea Modeller. The graphics follow the same naming convention as bodies, but with the *_Graphics* after the name of the body. Collision graphics has *COLL_* in front of the body name to clearly indicate the purpose of that graphic. Body graphics of Truck_Body and collision graphics for the same body as seen in body preview window of Mevea Modeller are presented in Figure 13.

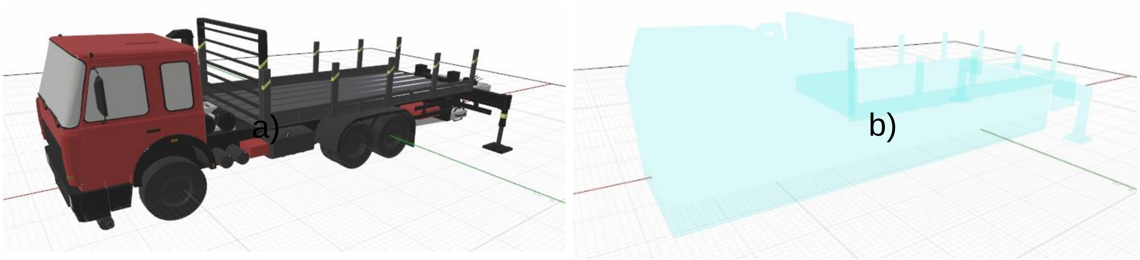


Figure 13. a) Body graphics for Truck_Body b) Collision graphics for Truck_Body

### 3.1.3 Bodies and respective coordinate systems

After the graphics is imported, it is easier to create the bodies, as the visualisation graphics helps to place the bodies in the world, as well as to ensure the correct location of coordinate systems.

The bodies are created in the order that allows to clearly see the connections between the bodies and identify their location in the system. The following bodies were created:

- Ground
- Truck_Pillar
- Truck_LiftBoom
- Truck_JibBoomAttachment
- Truck_JobBoom
- Truck_Hook
- Bag

The location of the bodies within the model is specified in Figure 14. The bodies that are pointed in the model but not listed above are dummies and their creation will be covered in the respective section of this thesis.

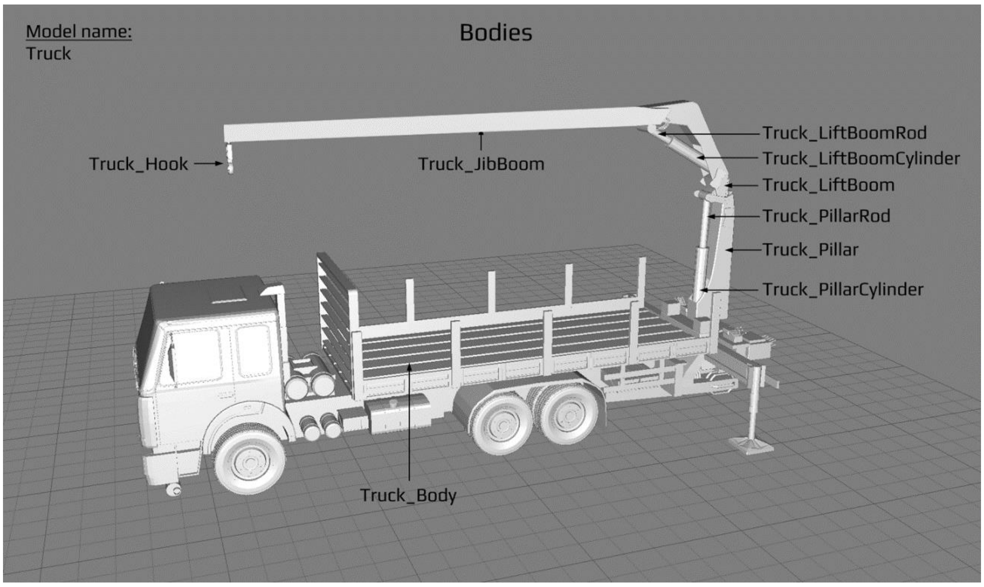


Figure 14. Bodies of Truck model

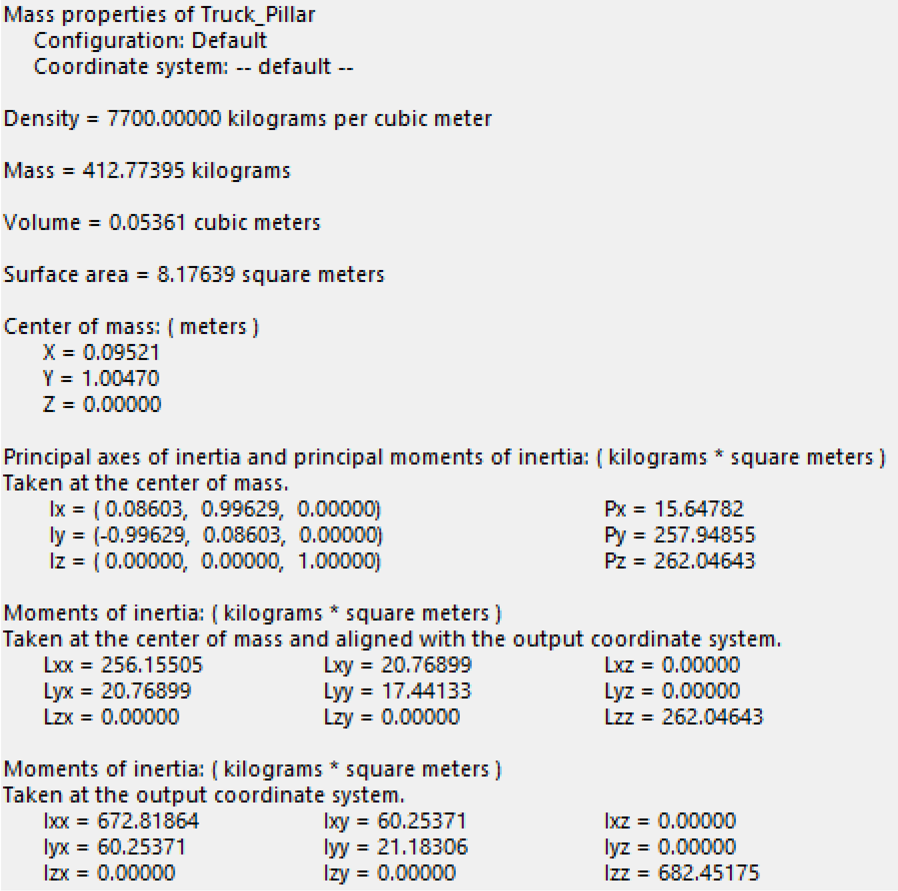
Each body requires specifying the following parameters, as presented in Table 3 with example of body Truck_Pillar.

| | |
|---|---|
| **Body Type** | Rigid |
| **Relative to body** | Ground |
| **Visualisation Graphics** | Truck_Pillar_Graphics |
| **Position [m]** | x = 3.21922<br>y = 0.8830184<br>z = 0.00036178 |
| **Orientation [rad]** | Phi = 90°<br>Theta = 37.5°<br>Psi = -90° |
| **Mass [kg]** | 412.77395 |
| **Moments and products of inertia** | Ixx = 672.81864 Iyx = 60.25371 Izx = 0<br>Ixy = 60.25371 Iyy = 21.18306 Iyz = 0<br>Ixz = 0 Izy = 0 Izz = 682.45175 |
| **Center of Mass [m]** | x = 0.09521<br>y = 1.0047<br>z = 0 |
| **Use Inertia Frame** | Inertia defined using Inertia Definition Frame |
| **Inertia Definition Frame [m]** | x = 0<br>y = 0<br>z = 0 |

Table 3. Parameters of the body Truck_Pillar

The position of the body is always described with relation to a certain body. The values in Position and Orientation parameters define the local coordinate system of the body. Orientation is defined using Euler angels and successive rotations.

Inertia properties of this body (mass, moments and products of inertia, and centre of mass) are defined using Solidworks and then imported to Mevea Modeller. For this, the 3D model is converted into .stl file that is possible to open with Solidworks. After that, a rough Solidworks model is created around the imported model, as .stl file is imported as a shell without possibility to add mass properties. Finally, the material is specified for the model and mass properties analysis is executed. Solidworks output for Truck_Pillar mass properties is presented in Figure 15 below.



```
Mass properties of Truck_Pillar
    Configuration: Default
    Coordinate system: -- default --

Density = 7700.00000 kilograms per cubic meter

Mass = 412.77395 kilograms

Volume = 0.05361 cubic meters

Surface area = 8.17639 square meters

Center of mass: ( meters )
     X = 0.09521
     Y = 1.00470
     Z = 0.00000

Principal axes of inertia and principal moments of inertia: ( kilograms * square meters )
Taken at the center of mass.
      Ix = ( 0.08603,  0.99629,  0.00000)                  Px = 15.64782
      Iy = (-0.99629,  0.08603,  0.00000)                  Py = 257.94855
      Iz = ( 0.00000,  0.00000,  1.00000)                  Pz = 262.04643

Moments of inertia: ( kilograms * square meters )
Taken at the center of mass and aligned with the output coordinate system.
     Lxx = 256.15505          Lxy = 20.76899          Lxz = 0.00000
     Lyx = 20.76899           Lyy = 17.44133          Lyz = 0.00000
     Lzx = 0.00000            Lzy = 0.00000           Lzz = 262.04643

Moments of inertia: ( kilograms * square meters )
Taken at the output coordinate system.
     Ixx = 672.81864          Ixy = 60.25371          Ixz = 0.00000
     Iyx = 60.25371           Iyy = 21.18306          Iyz = 0.00000
     Izx = 0.00000            Izy = 0.00000           Izz = 682.45175
```

Figure 15. Mass properties of body Truck_Pillar

The parameters for other bodies listed above are created in the same manner. The parameter *Relative to body* is always set to call the previous body in the model (e.g. body Truck_LiftBoom is relative to body Truck_Pillar). The position and orientation of the body is therefore defined relatively to the previous body.

### 3.1.4 Constraints

Constraints are created in the location of joints. Mevea Modeller supports 13 different joint types (floating, spherical, universal, hinge, translational, cylindrical, inplane, fixed, inline, perpendicular, spherical-spherical, revolute-spherical and revolute-revolute). The joint defines the number of degrees of freedom and which directions of translational and rotational movements are constrained.

For simple constraints used in this thesis model, definition contains at least the following parameters: name, joint type, body 1 and joint coordinate system in that body, body 2 and the respective coordinate system. Therefore, one additional task in this step is creating joint coordinate systems.

If the bodies are positioned correctly in the previous step, then the joint coordinate systems follow the rule: in Body 1 it has the same coordinates as position of Body 2; in Body 2 it has coordinates (0; 0; 0). Sometimes, though, the orientation of created coordinate systems has to be adjusted to ensure the fact that the correct axis are constrained. Table 4 below presents the coordinate systems associated with joint between Ground and Truck_Pillar.

| **Name** | Ground.Pillar | Truck_Pillar.Ground |
|---|---|---|
| **Position** | x = 3.21922<br>y = 0.8830184<br>z = 0.00036178 | x = 0<br>y = 0<br>z = 0 |
| **Orientation** | x = 90°<br>y = 0<br>z = 0 | x = 90°<br>y = 0<br>z = 0 |

Table 4. Coordinate Systems for joint between Ground and Truck_Pillar

After that, the joint itself is created. Conventionally, the constraints are named as the *FirstBody_SecondBody*. This way the parameters for bodies and respective coordinate systems are quite self-explanatory, therefore Table 5 below presents only the names of the constraints and types of the joints used.

| **Name** | **Joint type** |
|---|---|
| Body_Pillar | Revolute |
| Pillar_LiftBoom | Revolute |
| LiftBoom_JibBoomAttachment | Revolute |
| JibBoomAttachment_JibBoom | Fixed |
| JibBoom_Hook | Fixed |
| Hook_Bag | Spherical |

Table 5. Names and joint types in Truck model

If the locations of the coordinate systems as specified in Table 4 are correct and are located according to the joint types, then the constraint definition is rather a simple task, as shown in Table 6.

| | |
|---|---|
| **Name** | Body_Pillar |
| **Joint Type** | Revolute |
| **Active** | Yes |
| **Body1** | Ground |
| **Body1 CS** | Ground.Pillar |
| **Body2** | Truck_Pillar |
| **Body2 CS** | Truck_Pillar.Ground |

Table 6. Parameters of constraint Body_Pillar

### 3.1.5 Forces

Mevea Modeller offers nine different types of forces, but in the model developed only translational and rotational forces are needed. Two translational and one rotational force is needed to represent the hydraulics of the system, and three more rotational forces act as friction forces that restrict the movement of the sand bag.

Translational forces represent the movement of two hydraulic cylinders associated with bodies Pillar and LiftBoom. The parameters shown in Table 7 below describe the force ForceT_Lift that represents the cylinder between parts Truck_Pillar and Truck_LiftBoom. The coordinates specified in *Position of force in body* describe the ends of the cylinder, that has to be represented using more components: dummy to ensure that mass and inertia properties are taken into account and the force is placed correctly, and the hydraulic actuators in Hydraulic components.

In Mevea Modeller the hydraulic actuators have to be connected to a force or a torque component, that is why the following forces are created. However, these forces are in the end overridden by the properties of hydraulic actuators, that is why all spring force components are set to zero.

| | |
|---|---|
| **Name** | ForceT_Lift |
| **Body 1** | Truck_Pillar |
| **Position of force in body1 [m]** | x = - 0.221712 |

| | |
|---|---|
| | y = 0.22223<br>z = - 0.00376529 |
| **Body 2** | Truck_LiftBoom |
| **Position of force in body2 [m]** | x = - 0.221712<br>y = 0.165<br>z = -0.00376529 |
| **Spring constant [N/m]** | 0 |
| **Damping constant [Ns/m]** | 0 |
| **End damper spring constant [N/m]** | 0 |
| **End damper damping constant [Ns/m]** | 0 |
| **Spring initial length [m]** | 0 |
| **Spring minimum length [m]** | 0 |
| **Spring maximum length [m]** | 0 |

Table 7. Parameters of ForceT_Lift

The same logic applies to parameters for ForceT_Jib that represents the cylinder between Truck_LiftBoom and Truck_JibBoomAttachment.

One rotational force represents the slewing movement of Truck_Pillar. The parameters for this force are represented in Table 8. In this case, the logic behind spring parameters is pretty much the same as for the translational forces, although this time the spring is not linear anymore, but rotational. *Direction vector f* represents the vector perpendicular to torque axis in local coordinate system of Body1. *Direction vector g*, on the other hand, is perpendicular to both torque axis and *Direction vector f*. These vectors are unit vectors, and the direction that is not set to one shows around which axis the force will act.

| **Name** | ForceR_Pillar |
|---|---|
| **Body1** | Ground |
| **Direction vector f, body 1** | x = 0<br>y = 0<br>z = 1 |
| **Direction vector g, body 1** | x = 1<br>y = 0<br>z = 0 |
| **Body2** | Truck_Pillar |
| **Direction vector f, body 2** | x = 0<br>y = 0<br>z = 1 |
| **Spring constant [Nm/rad]** | 0 |
| **Damping constant [Nms/rad]** | 0 |
| **End damper spring constant [Nm/rad]** | 0 |
| **End damper damping constant [Nms/rad]** | 0 |
| **Spring minimum angle [rad]** | 0 |
| **Spring maximum angle [rad]** | 0 |

| **Spring initial length [m]** | 0 |
|---|---|

Table 8. Parameters of ForceR_Pillar

Finally, three forces that represent the friction in joint between Truck_Hook and Bag are added. The idea behind these forces is to create the counter torque that would resist the excess rotation of the Bag. As a force can define the rotation only around one axis at a time, three separate forces for each axis are defined. The summary of parameters for these forces is presented in Table 9.

| **Name** | ForceR_FrictionX | ForceR_FrictionY | ForceR_FrictionX |
|---|---|---|---|
| **Body1** | Truck_Hook | Truck_Hook | Truck_Hook |
| **Direction vector f, body 1** | x = 0<br>y = 0<br>z = 1 | x = 0<br>y = 0<br>z = 1 | x = 1<br>y = 0<br>z = 0 |
| **Direction vector g, body 1** | x = 0<br>y = 1<br>z = 0 | x = 1<br>y = 0<br>z = 0 | x = 0<br>y = 1<br>z = 0 |
| **Body2** | Bag | Bag | Bag |
| **Direction vector f, body 2** | x = 0<br>y = 0<br>z = 1 | x = 0<br>y = 0<br>z = 1 | x = 1<br>y = 0<br>z = 0 |
| **Spring constant [Nm/rad]** | 0 | 0 | 0 |
| **Damping constant [Nms/rad]** | 5000 | 5000 | 5000 |
| **End damper spring constant [Nm/rad]** | 0 | 0 | 0 |
| **End damper damping constant [Nms/rad]** | 0 | 0 | 0 |
| **Spring minimum angle [rad]** | 0 | 0 | 0 |
| **Spring maximum angle [rad]** | 0 | 0 | 0 |
| **Spring initial length [m]** | 0 | 0 | 0 |

Table 9. Parameters of rotational forces ForceR

The forces are the same with only difference in the axis that they affect. All spring parameters expect of damping constant are set to zero to create needed effect of counter torque.

### 3.1.6 Dummies

Dummies are the bodies that are not connected to kinematic chain itself, but affect mass and inertia properties of the body that they are attached to. Mevea Modeller has five types of dummies: static, body-to-body force, tyre, part and constraint. In this model only static and body-to-body force dummies were used. The position of static dummy does not change during the simulation, while body-to-body force is attached to the respective primitive and affects the position of the body involved during the simulation.

The parameters of the dummies depend on their types. Examples of parameters for Dummy_Pillar_Cylinder are shown in Table 10. It can be easily noted that the parameters of the dummy in Table 10 are quite similar as of the bodies defined previously, however some changes occur.

The dummies that represent hydraulic cylinders are created in two parts: cylinder and piston. Both of them are created as B2BF type and are connected to the respective translational force. The *Dummy index* shows to which body of force the dummy is related to. Normally the cylinder has index 1 and the piston – 2. As seen in Table 10, other parameters are again the same as in bodies definition, with the only difference that the position and orientation cannot be edited, as those parameters are defined by the location of the force connected to the dummy. For B2BF dummies in this model, mass and inertia parameters were evaluated in Solidworks, in the similar manner as for the previously described bodies.

| | | | |
|---|---|---|---|
| **Name** | Dummy_PillarCylinder | | |
| **Dummy type** | B2BF | | |
| **Relative to** | ForceT_Lift | | |
| **Dummy index** | 1 | | |
| **Visualisation Graphics** | Truck_PillarCylinder_Graphics | | |
| **Mass [kg]** | 31.63103 | | |
| **Moments and products of inertia** | Ixx = 0.31882<br>Ixy = 0<br>Ixz = 0 | Iyx = 0<br>Iyy = 14.16431<br>Izy = 0 | Izx = 0<br>Iyz = 0<br>Izz = 14.16431 |
| **Center of Mass [m]** | x = -0.56644<br>y = 0<br>z = 0 | | |

| | |
|---|---|
| **Use Inertia Frame** | Inertia defined using Inertia Definition Frame |
| **Inertia Definition Frame [m]** | x = 0<br>y = 0<br>z = 0 |

Table 10. Parameters of Dummy_PillarCylinder

After creation of dummies, the static model starts to look almost like a finished model, however it is not functional yet, as collisions, hydraulics and the control system for the system is absent.

### 3.1.7 Collisions

Collisions in simulation models allow achieving realistic behaviour of the simulator. Different bodies and dummies can collide within the model. In Mevea Modeller, the collisions are presented as spring-damper systems with friction between colliding parts. To define the collisions correctly, firstly, pairs of colliding components have to be defined. Generally, the number of collision pairs has to be reduced to save computational power and provide smoother real-time simulation, but the realism of the model should not suffer because of that.

In the model created, six collision pairs were defined: Ground and Truck_Body, Ground and Bag, Truck_Body and Bag, Truck_Body and Hook, Truck_Body and JibBoom, Bag and JibBoom. In Mevea Modeller, the collision definition requires the parameters as presented in Table 11 for collision between Ground and Bag.

| | |
|---|---|
| **Name** | Ground_Bag |
| **Body A** | Ground |
| **Body B** | Bag |
| **Graphics A** | COLL_Truck_Ground_Graphics |
| **Graphics B** | COLL_Bag_Graphics |
| **Spring constant** | 5e7 |
| **Restitution coefficient** | 0.1 |
| **Use friction** | Yes |
| **Friction** | Collision_Friction |

Table 11. Parameters of collision Ground_Bag

The order in which the bodies are specified in this window does not matter, however it is important to follow the convention of Body A and Body B when selecting the collision graphics. Spring constant here defines the “stiffness” of the collision between the bodies: low spring constant might result into bodies penetrating each other, while higher spring constant will mean more sharply defined contact between two bodies.

Friction in this case is responsible for the stick-slip phenomenon and has to be defined separately. Stick-slip phenomenon is quite complicated and is out of the scope of this thesis, therefore only the definition of parameters is presented in Table 12.

| **Name** | Collision_Friction |
|---|---|
| **Constant pressing force [N]** | 0 |
| **Stiffness of contact surface [N/m]** | 4000 |
| **Damping of contact surface [Ns/m]** | 500 |
| **Stribeck velocity [m/s]** | 0.01 |
| **Viscosity coefficient [Ns/m]** | 0.1 |
| **Static friction coefficient [0..1]** | 0.3 |
| **Dynamic friction coefficient [0..1]** | 0.1 |
| **Exponent** | 2 |

Table 12. Parameters of Collision_Friction

### 3.1.8 Hydraulics

Mevea Modeller allows hydraulic system to be created directly inside the software, so no external programmes are needed. The components are added into *Body* tree under *Hydraulics*. Mevea Modeller supports a lot of different kinds of hydraulic components that are enough to model mobile machines.

It is useful to create a topological scheme in the beginning, as shown in Figure 16. Most of the components that are used in the model were already described in Section 2.3.4, expect of L90LS pressure compensated directional valve (L90Slew, L90Lift and L90Jib in Figure 16). This is a mobile directional control valve that is often used in real mobile machines, and it combines 4/3 directional valve, pressure reducing valve with lock spool, pressure relief valve and load sensing valve.

After the topology of the system was created, the components are added into Mevea Modeller. Firstly, volumes are added. They represent the volume of oil in components they are connected to. The only variable specified in Mevea Modeller for hydraulic volume is initial pressure, as shown in Table 13.

Now the hoses can be created, as they are related to just created volumes. The parameters in case of hoses are estimations of possible real values for a mobile crane, and an example of parameters for hose connecting volume Vol_SlewCyl_A and slewing cylinder SlewCyl_CraneSlew is shown in Table 14.

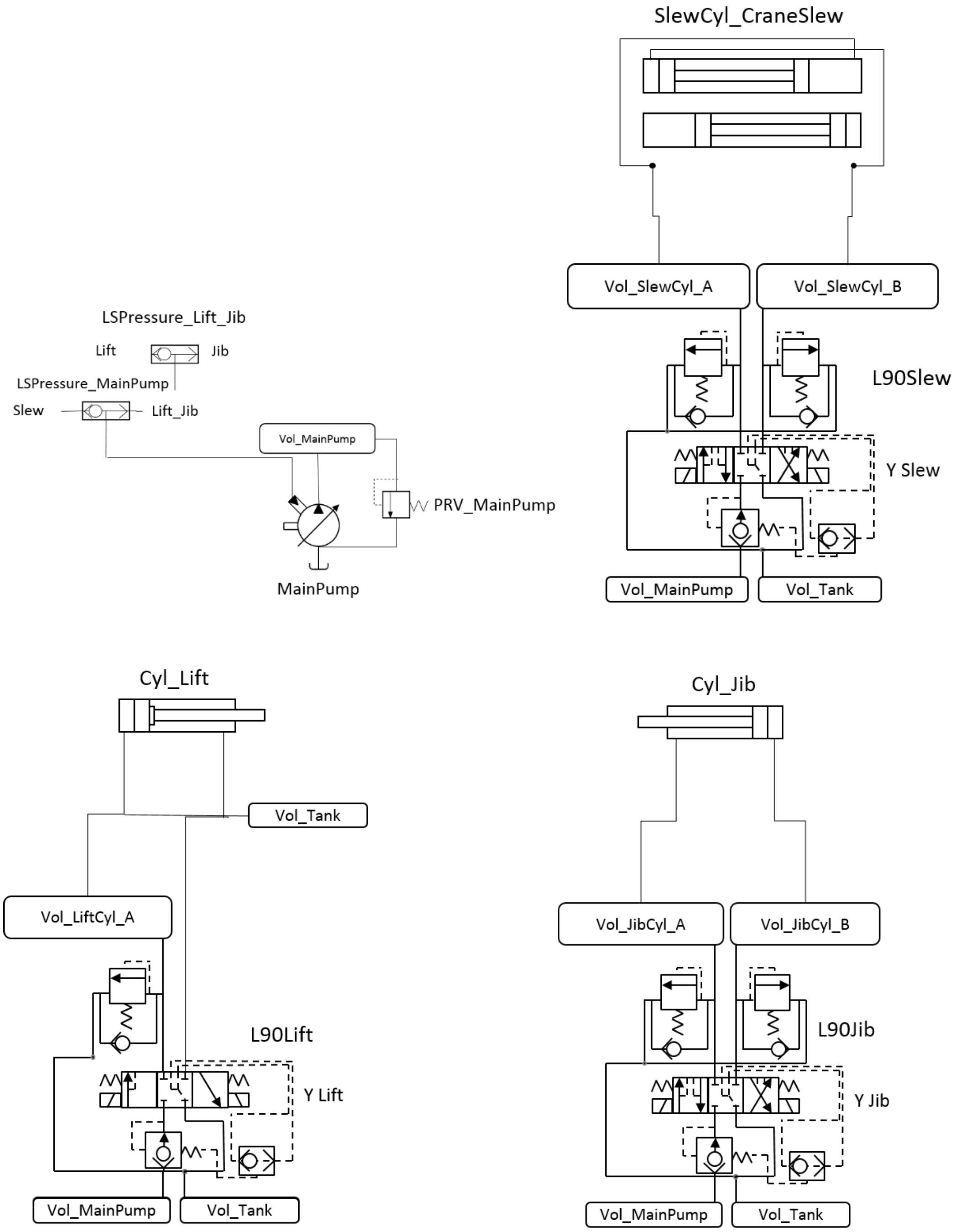


Figure 16. Topology of hydraulic circuit for Truck model (courtesy of Timo Julkunen, Etteplan)

| | |
|---|---|
| **Name** | Vol_MainPump |
| **Type** | Volume |

| | |
|---|---|
| **Initial pressure [Pa]** | 1e5 |

Table 13. Parameters of Vol_MainPump

| | |
|---|---|
| **Name** | Hose_Slew_A |
| **Type** | Hose |
| **Connected volume** | Vol_SlewCyl_A |
| **Length [m]** | 1.5 |
| **Hose diameter [inch]** | 0.75 |
| **Bulk modulus of hose/pipe [Pa]** | 7e9 |

Table 14. Parameters of Hose_Slew_A

Next component to be created is the L90LS pressure compensated directional valve. This is the most difficult component to give the parameters for, as it combines a lot different components inside it. However, with this component, the system in the end becomes easier for modelling and simulation. As the model created is a more of a demonstration model rather than a real one, that is why estimations and values of general knowledge were used, as shown in Table 15.

| | |
|---|---|
| **Name** | L90Slew |
| **Spool section type** | TOT |
| **Main spool type** | Closed centre position, D |
| **Pressure control on A – position** | No |
| **Force feedback on main spool** | No |
| **A port anticavitaion and shock valve** | Anticavitaion + Shock |
| **B port anticavitaion and shock valve** | Anticavitaion + Shock |
| **Port P** | Vol_MainPump |
| **Port A** | Vol_SlewCyl_A |
| **Port B** | Vol_SlewCyl_B |
| **Port T** | Vol_Tank |
| **Positive overlap [%]** | 32 |
| **Maximum control signal [V, mA]** | 500 |
| **Input signal on nominal flow rate [V, mA]** | 500 |
| **Compensator type** | Compensator with lock-hold |
| **Nominal flow rate P -> A, A -> T, P -> B, B -> T [l/min]** | 70 |
| **Pressure drop on nominal flow rate P -> A, A -> T, P -> B, B -> T [bar]** | 35 |
| **Leak flow rate [l/min]** | 0.1 |
| **Pressure difference on leak flow [bar]** | 100 |
| **Pressure raise required for full spool opening [bar]** | 1 |
| **Step response [s]** | 0.1 |
| **Pressure compensator setting pressure [bar]** | 8 |
| **Pressure compensator flow rate [l/min]** | 150 |

| | |
|---|---|
| **Pressure compensator pressure drop on nominal flow [bar]** | 8 |
| **A and B port shock valve setting pressure [bar]** | 250 |
| **Maximum LS-pressure A&B port [bar]** | 180 |

Table 15. Parameters of L90Slew

L90Slew valve represents the control system for slewing cylinder SlewCyl_CraneCyl, connected to this cylinder via respective hose and volume. The parameters used for SlewCyl_CraneCyl are presented in Table 16. Cylinders are always connected to a force or torque primitive, in order to adjust the magnitude of the force acting between the bodies. In this case it is *ForceR_Pillar* that was previously defined in Table 8. Cylinder modelling always involves friction modelling, in this case represented by a friction spline, that is specified separately. The parameters of friction spline are shown in Table 17. $x$ values show the piston velocity in m/s and $y$ values are the friction coefficient. The values for cylinder piston diameter and length were estimated based on the system performance. This is one of the biggest advantage of real-time simulation: it is possible to easily adjust parameters on the go without having to replace real component. The rest of the values are general most common values used in mobile machines.

| **Name** | SlewCyl_CraneSlew |
|---|---|
| **Primitive name** | ForceR_Pillar |
| **Friction spline** | Spline_CylinderFriction |
| **Port 1** | Vol_SlewCyl_A |
| **Port 2** | Vol_SlewCyl_B |
| **Cylinder piston diameter [mm]** | 100 |
| **Cylinder piston length [mm]** | 300 |
| **Cylinder chamber length [mm]** | 1300 |
| **Minimum stroke [mm]** | 20 |
| **Radius of slew cylinder pinion [m]** | 0.144 |
| **Cylinder bulk modulus [Pa]** | 1e10 |
| **Cylinder coefficient** | 0.9 |
| **Cylinder leak flow [l/min]** | 0.01 |
| **Cylinder leak flow pressure difference [bar]** | 100 |
| **End damping spring coefficient [N/m]** | 5e7 |
| **End damping damping coefficient [Ns/m]** | 5e5 |

Table 16. Parameters of SlewCyl_CraneSlew

| **Name** | Spline_CylinderFriction | |
|---|---|---|
| **Number of points** | 12 | |
| **x1 / y1** | 0 | 0 |
| **x2 / y2** | 0.0001 | 1 |

| | | |
|---|---|---|
| **x3 / y3** | 0.025 | 0.9 |
| **x4 / y4** | 0.05 | 0.4 |
| **x5 / y5** | 0.075 | 0.225 |
| **x6 / y6** | 0.1 | 0.2 |
| **x7 / y7** | 0.125 | 0.225 |
| **x8 / y8** | 0.2 | 0.275 |
| **x9 / y9** | 0.3 | 0.35 |
| **x10 / y10** | 0.4 | 0.415 |
| **x11 / y11** | 1 | 0.8 |
| **x12 / y12** | 3 | 0.85 |

Table 17. Parameters of Spline_CylinderFriction

Parameters for double-acting cylinders Cyl_Lift and Cyl_Jib follow the same logic as just specified SlewCyl_CraneSlew, with minor adjustments due to the different type of cylinder used. Table 18 shows the summary of most important parameters of Cyl_Lift that are different from SlewCyl_CraneSlew. The values of piston and piston rod diameter are estimations based on the forces that act on the cylinder. Parameters related to cylinder and piston rod length were estimated based on the tryout of the system without hydraulics and recording the length of the force responsible for the respective component; some parameters were calculated based on those estimations.

| **Name** | Cyl_Lift |
|---|---|
| **Cylinder type** | Double acting |
| **Primitive name** | ForceT_Lift |
| **Friction spline** | Spline_CylinderFriction |
| **Port 1** | Vol_LiftCyl_A |
| **Port 2** | Vol_LiftCyl_B |
| **Cylinder piston diameter [mm]** | 200 |
| **Cylinder piston rod diameter [mm]** | 105 |
| **Piston rod length [mm]** | 1137 |
| **Cylinder chamber length [mm]** | 1120 |
| **Piston length [mm]** | 50 |
| **Cylinder attachment length [mm]** | 100 |
| **Minimum stroke (end damper) [mm]** | 700 |
| **Maximum stroke (end damper) [mm]** | 1050 |

Table 18. Parameters of Cyl_Lift

At this point, the circuits for slewing, lifting and jibbing movements are ready, however there is no pump associated, so there is no power to activate the movement of these cylinders. The first step for creating pump circuit is adding load sensing pressures, LSPressure_Lift_Jib and LSPressure_MainPump as shown in **Table 19**.

| **Name** | LSPressure_Lift_Jib | LSPressure_MainPump |
|---|---|---|
| **Port 1** | L90Lift | L90Slew |
| **Port 2** | L90Jib | LSPressure_Lift_Jib |

Table 19. Parameters of load sensing pressures

At this point, the only two components left are pressure relief valve and the pump itself. Parameters for pressure relief valve connected to the pump are specified in Table 20. Pressure relief valve protects the system from receiving extra pressure, that is why *Setting pressure* value should be chosen carefully to prevent the system from blowing. In this case, it was chosen as one of the standard values in mobile machines.

| **Name** | PRV_MainPump |
|---|---|
| **Port 1** | Vol_MainPump |
| **Port 2** | Vol_Tank |
| **Port 3** | Vol_MainPump |
| **Spool functionality** | Closed without piloting, Pressure release valve |
| **Nominal flow rate [l/min]** | 180 |
| **Pressure drop on nominal flow rate [bar]** | 10 |
| **Setting pressure [bar]** | 270 |
| **Pressure rise required for full valve opening [bar]** | 7 |
| **Step response time [s]** | 0.1 |

Table 20. Parameters of PRV_MainPump

The last component of the hydraulic system for the Truck is a pump that supplies the pressure to the whole hydraulic system. The parameters for the pump are shown in Table 21. These are parameters that are generally used in hydraulic pumps for mobile machines.

| **Name** | MainPump |
|---|---|
| **Type** | Variable displacement pump |
| **Port A** | Vol_MainPump |
| **Port B** | Vol_Tank |
| **LS / pilot pressure port** | LS_Pressure_MainPump |
| **Constant rotational velocity [rpm]** | 1500 |
| **Geometric displacement [cc]** | 140 |
| **Mechanical coefficient** | 0.94 |
| **Volumetric coefficient** | 0.93 |
| **Controller gain** | 3.5e-5 |
| **Time constant [s]** | 0.085 |
| **Set pressure [bar]** | 25 |
| **Stand-by pressure [bar]** | 25 |
| **Maximum LS-pressure [bar]** | 230 |

| | |
|---|---|
| **Pump leak flow [l/min]** | 0.1 |
| **Pump leak flow pressure difference [bar]** | 100 |

Table 21. Parameters of MainPump

After the pump is created, the hydraulic circuit is complete, and the model behaves in a way that it should. However, the model still lacks the ability to interpret the user's input, which is probably one of the most important parts of real-time simulation.

### 3.1.9 Inputs

Finally, the model needs the control signals that would translate the human input to simulated model. Mevea Modeller allows the user to easily specify the inputs directly inside the program. The default controller is a joystick; however, it can be replaced with a keyboard-emulated joystick inside Mevea Solver. Both digital and analog input types are available.

The created model requires only three inputs, that match the forces created in Section 3.1.4 and hydraulics from Section 3.1.8: Slew, Lift and Jib. These movements are controlled by pressure compensated proportional directional valves L90LS. The configuration of the input representing slewing movement is described in Table 22.

| | |
|---|---|
| **Name** | Input_Slew |
| **Input type** | Analog |
| **Primitive type** | L90LS |
| **Primitive name** | L90Slew |
| **Primitive sub** | Control Voltage |
| **I/O block** | 0 |
| **I/O channel** | 0 |
| **Scale** | -500 |
| **Deadzone** | 0.2 |

Table 22. Parameters of Input_Slew

The analog input is chosen as the most logical solution that represents the behaviour of the real system. Primitive type and name depend entirely on the component that is chosen, and primitive sub describes the type of the control signal fed, that in turn depends on the primitive type. I/O block in this case describes the number of joysticks used, and I/O channel stands for joystick axis/button index. Scale parameter shows scaling co-efficient of the input, it is used to reduce or increase the responsiveness of the component to the signal fed. (Mevea Ltd., 2017, p. 149).

### 3.2 Adding flexible body to simulation model

After the initial model was created, the flexibility is added to the body JibBoom, as it is the body that is most affected in the process of simulation.

The body was divided into 19 separate similar elements, as was described in Section 2.4. This results into 20 nodes needed for definition of this body. To accommodate this graphically, the 3D model created in Blender was divided in 19 elements as well, but the end user of the simulation model is not able to see this directly.

The process of changing the body type is very simple, as shown in Table 23. In the body parameters, *Body Type* is changed from rigid (as was specified originally) to flexible. After that mass and inertia parameter fields become inactive, as they have to be replaced with so-called Flexible Files in .dat format, a prefix of which is specified in the respective field. Flexible files are placed in the root folder of the model and follow the naming convention *ModelName_FlexFilePrefix_MatrixName.dat*. Flex modes tell the number of nodes that are used in vibration modes analysis. Generally only two first are taken, as they produce the lowest vibration frequencies that can be used in real-time simulation. High vibration frequencies are not used in real-time simulation as the time step of the simulation is rather small. The last parameter describes the relative damping of the modes, and it is taken as a fraction of the mass matrix.

| | |
|---|---|
| **Body type** | Flexible |
| **Flex File Prefix** | beam_hard |
| **Flex Modes** | 1;2 |
| **Relative damping of modes** | 0.005;0.005 |

Table 23. Parameters of JibBoom responsible for flexibility of the body

Flexible files contain the matrices that are calculated based on SciLab code of D.Sc. Jussi Sopanen, that he developed in connection with his Doctoral thesis *Studies of rotor dynamics using a multibody simulation approach*. The code uses the theoretical approaches that were described in Section 2.4 of the present thesis. The main parameters that the code requires to run are only the original location of the nodes and the material parameters. After specifying those parameters, the data files that are ready to be used in Mevea Modeller can be produced.

Changing material parameters affects the output quite dramatically, as can be seen by simply changing the material's modulus of elasticity. The general value for steel is about $2 \times 10^{11} Pa$, but if this value is decreased to $2 \times 10^{10} Pa$, then the behaviour of the structure changes quite dramatically, as shown in Figure 17 (Korkealaakso, 2017).

Figure 17. JibBoom with $E = 2 \times 10^{11} Pa$ and $E = 2 \times 10^{10} Pa$

### 3.3 Tutorial production

The initial plan of Mevea Ltd. was to create the tutorial about creation of flexible body. However, as seen from the Section 3.2 in comparison to Section 3.1, this is a rather simple and quick process. Therefore, the initial structure of the tutorial was changed to the following:

1. Introduction
2. Creation of Flexible Body
    a. Adding Flexibility to Jib_Boom
3. Adjusting Solver Window
    a. Changing Basic Parameters
    b. Recording Input
    c. Creating Plots
    d. Saving and Loading Workspace

The tutorial, therefore, first introduces the student to the concept of flexible bodies and how to add one to the existing model. After that different ways to customise Mevea Solver window are described. They are listed in such an order, that if the student follows all of the paragraphs one by one, he or she is able to learn how to easily change and track the parameters of the simulated model. Tracking and changing the parameters is a useful feature in research and development.

As the creation of flexible body was already described in previous section of the thesis, only part regarding Mevea Solver window adjustments will be described here.

### 3.3.1 Changing Basic Parameters

Mevea Modeller offers a wide range of parameters that can be adjusted in order to produce a correct reliable model. Some parameters cannot be altered while running the dynamic simulation. With the following instructions, though, the most important parameters can be changed while the simulation is still running.

Parameters can be changed by going to *Control* -> *Parameter view*. Under three tabs *Hydraulics*, *Bodies* and *Parameters*, some of the respective parameters can be modified. In case of the tutorial developed for the thesis, it was determined that the most beneficial and interesting parameter to adjust is the mass of the *Bag*, as the mass will affect the deflection of *JibBoom* – the flexible body. The student is advised to change the mass of *Bag* to a higher value than default, so that it is possible to see bigger deflection of the *JibBoom*. Once the parameter is changed in the *View/Modify Parameters* window, it affects the dynamic simulation immediately, and the user is able to see the response right away, as seen in Figure 18.

Figure 18. Before and after changing *Bag* mass in parameter window

### 3.3.2 Recording input

Changing a parameter is a good first step for customising the simulation, however it might be rather useless if it is not possible to clearly compare the output. The next step for quality analysis is therefore recording the input, as this will allow the user to see how the parameters that are changed in Solver or Modeller affect the model that is controlled by the same signals.

The user can record input created by joystick or keyboard-emulated joystick by going to *Control -> Input control -> Start record*. The input is stored in Mevea-specific file format .iob, and contains only the information about the signals given to the specified inputs (Input_Slew, Input_Lift and Input_Jib in case of the model presented in tutorial).

### 3.3.3 Creating plots

Plotting is a very important tool for research and development, as it allows the user to clearly see the changes in output during the simulation, as well as to compare how the change in design parameters affects the model, especially when it is possible to compare the model in the same condition.

Mevea modeller allows the user to create the plots of any printable variable that is specified in Mevea Solver Reference Manual in the respective table in the end of each of the component's description. Plots are produced in real time, and it is possible to check the value of the parameters by hovering over the plot. It is also possible to write the plot data to .txt file or save the plot as .tga image. One of the plots that Mevea Solver is able to produce is shown in Figure 19.

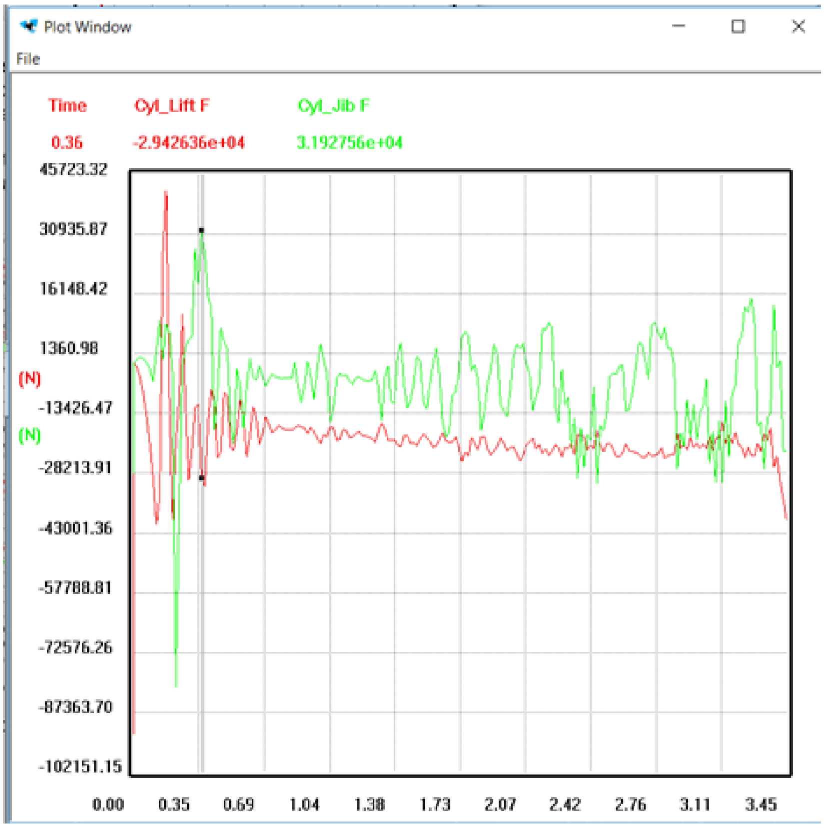
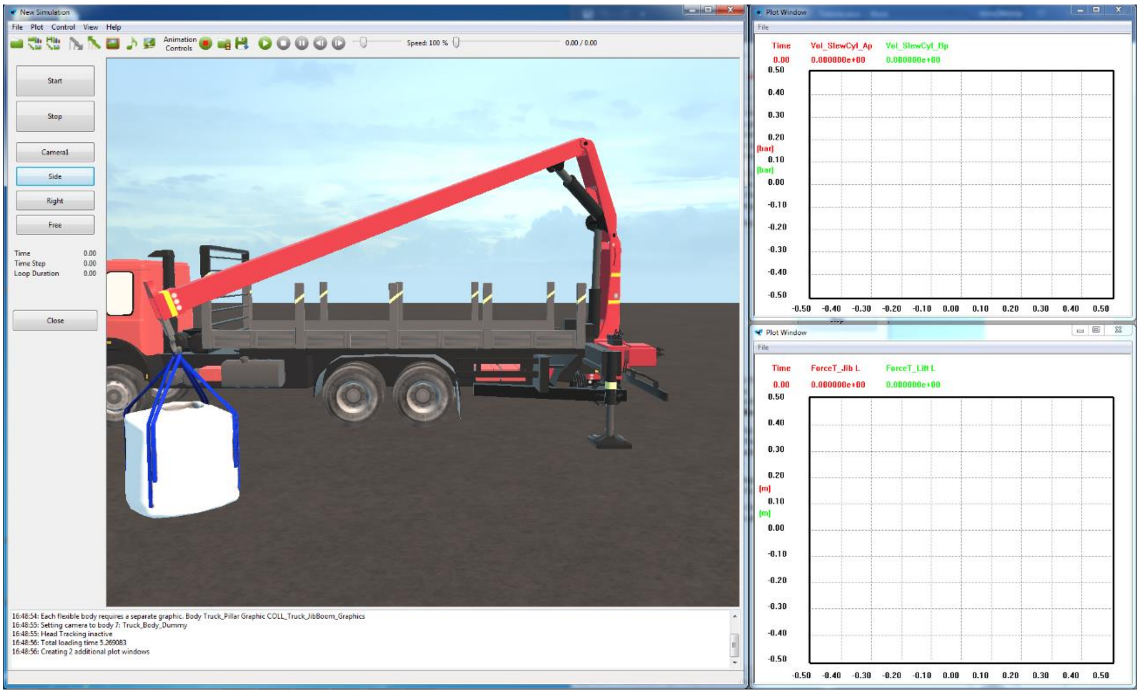


Figure 19. Plot example produced in Mevea Solver

### 3.3.4 Saving and Loading Workspace

Sometimes the user needs a lot of plots to be created, and the user needs them every time when new dynamic simulation using Mevea Solver is run. Creating a lot of new plot windows every time can be quite challenging and unrewarding task, that is why it is possible to save the workspace in Mevea Solver: position of simulation window, variables plotted and the position of the plots. The workspace files are model specific, so they can only be used with the model for which it was created. It is also important to save the workspace files in the model root folder, as saving it in another location breaks the links between the files.

Figure 20 shows the Mevea Solver window that was just loaded from the external file and is ready to run.



```
16:48:54: Each flexible body requires a separate graphic. Body Truck_Pillar Graphic COLL_Truck_JibBoom_Graphics
16:48:55: Setting camera to body 7: Truck_Body_Dummy
16:48:55: Head Tracking inactive
16:48:56: Total loading time 5.269083
16:48:56: Creating 2 additional plot windows
```

Figure 20. Mevea Solver window with workspace loaded from the saved file

Section about saving and loading workspace concludes the tutorial. After completing this tutorial, the user should be able to add flexibility to bodies already created in Mevea Modeller, as well as adjust Mevea Solver window in a way that would be beneficial for research and development associated with the developed model.

# 4 Discussion

The process of tutorial creation consisted of two main parts: model creation and tutorial writing, that consisted of smaller sub-tasks. On both of these major steps, Mevea Ltd., as the commissioner of this thesis, was consulted to make sure that the model and tutorial developed are in line with the company needs. Because of that, the model and tutorial are not exactly the same as was presented in the first meeting with the company. Even though, the final goal – create a demo model and a tutorial that are profitable for the company – was achieved.

The model created might seem rather complicated, however it is not totally a workable solution to represent a real machine, but rather a demo model. Nevertheless, this model is enough to be used in the tutorial, illustrate the basic concepts, and encourage the students to implement this knowledge in the future.

Tutorial was created based on the best practices in tutorial creation found over the Internet, taking into account the background of supposed students. The outline was discussed with Mevea Ltd. to make it suit their needs best. The biggest emphasis in the tutorial writing process was put on the language and formulations, because Mevea's customers are located all around the world, therefore they might have different level of English and Mevea software knowledge.

## 4.1 Tutorial demonstration with Saimaa UAS students

As was mentioned in Section 2.5, one important step in tutorial creation is testing it with person or people, who have not seen the tutorial before. That is why a demo session with Saimaa UAS students was held. The session happened during the lecture *Konetekniikan simulaatiot* (Simulations in Mechanical Engineering) for third year students of degree programme in Mechanical Engineering. This group was chosen, because they have already studied some basics of Mevea Modeller and Solver, but are not proficient with Mevea software yet.

Feedback forms were distributed to the students in the end of the session so that they can assess the performance of the author and give some hints and recommendations for future development.

Feedback was anonymous, and Google Forms was used to collect the responses. Around 20 students attended the session, however it was possible to run Mevea only for on 13 machines, therefore only 13 feedback forms were submitted in the end. The complete feedback forms can be found in Appendix 2. Feedback form for tutorial demonstration with Saimaa UAS students of this thesis. Feedback was mostly quantitative, with questions asking to rate certain parameters on the scale from 1 to 5 where 1 stands for "Strongly disagree with the statement" and 5 for "Strongly agree with the statement", however the students had the possibility to leave the comments in respective fields.

The statistical summary of quantitative responses is presented in Appendix 3. Summary of feedback form responses of the thesis. Only the most notable outcomes are outlined in this section.

In general, the students rated high the easiness, language, logic and usefulness of the tutorial. The purpose of the tutorial, according to them, was explained well from the beginning. Most of the students were able to understand the steps of the tutorial from the first time, even though only very few knew at least something about flexible bodies and ways of customizing Mevea Solver window.

Even though a lot of students did not know before the demonstration the concept of flexible bodies, after it almost all of them were able to tell the difference between flexible and rigid bodies. One student pointed out that the way of adding flexibility to the existing rigid body was not explained that well, which points out that probably a more detailed explanation is needed for that part. The way of flexible files creation was out of the scope of the tutorial and the demonstration, and only five students showed any interest in this topic. In the end, half of the group expressed their belief in using flexible bodies in the future simulations.

The students admitted that Mevea Solver window customisation is a useful feature, but only a little less than a half of them expressed the interest in learning more about how this works.

Most of the students are unsure if they are going to use Mevea software in the future, or even continue working with any kind of real-time simulation software, therefore half of them was not sure if they would be able to use the knowledge obtained during the demonstration in the future.

The general purpose of the tutorial is to teach the students to replicate the results in the future for their own models, and only two students were unsure if they would be able to do so.

## 4.2 Possibilities for improvement and future topics for tutorials

The demonstration with real students showed that the tutorial produced is good in language and content in general. The free-form comments pointed out couple of minor spelling mistakes. Even though the students saw the tutorial useful, they were unsure if they would be able to use the knowledge in the future, most probably because they were not sure about their future in general. The author believes that this parameter would go up in the professional community where the students follow the tutorial to improve their professional skills. The author also believes that the main goal of having the tutorial was achieved: the students admitted that they could reproduce the results in the future.

Feedback form also had some general questions that would help in defining the possible ways to develop Mevea tutorial base in the future. One of the questions was to suggest topics for possible future tutorials, but unfortunately only two responses were collected. One option was “trusses”, which is not a topic that would be suitable for Mevea software in the author’s opinion, as this area is better suited for FE analysis. Another answer was “car suspension system”, which seems to be a topic more suitable for MBS approach. Even though there was only two proper responses, one can draw a conclusion that students seem to be more interested in examples of concrete structures, rather than an abstract machine.

Another question concerned the way the students like receiving tutorials. Almost all of the students noted that they prefer to get tutorials in a form of direct teaching in the classroom. The second most popular option was PDFs directly embedded in software, followed by short 2-5 minute videos on a single topic. Some people admitted preferring long videos covering several topics, and the same amount of students said that they want tutorials to be as a webpage that can be easily found using a search engine. The least popular option was a forum thread with possible solutions discussion.

Mevea already uses direct teaching and PDFs embedded in software at the moment, which also seem to be the most popular options at least among this group of students. However, new ways of teaching and especially distant teaching are coming up, that is why video tutorials got several votes, though the percentage in the end was not as high as expected. At the same time, it was thought that short videos are preferred over the long ones, which seemed to be true.

Considering Mevea’s willingness to expand the customer base, the recommendation is to consider creating video tutorials as short videos and test this concept to see the response. If the response is positive, then this concept should be developed even further. It is also recommended to keep the PDF-based documentation up-to-date, as it is still one of the most preferred ways of getting the information. Direct teaching, as was pointed out by Mevea management, is becoming more and more complicated due to growing customer base, so these two options seem to be the most feasible ones to develop the distant teaching.

Other possible option that was not discussed with students, is the possibility of organising webinars – online seminars. Mevea employee would conduct direct teaching as if the students are in the classroom, but his actions would be streamed online to the students. The students would be able to stop the presenter and ask questions just as in direct teaching method. Webinars, however, require additional training and equipment on Mevea side, so this is the solution that should be carefully investigated in the beginning, before investing recourses into this option.

# 5 Conclusion

The objective of this thesis was to create a tutorial for Mevea Modeller – real-time simulation software – about flexible bodies creation. It was important to create the tutorial that would be beneficial for Mevea Ltd. as a company in the future, therefore there were minor adjustments made to the tutorial outline during the process of tutorial development.

The process of tutorial creation was described in the present thesis. However, the steps that are taken to create the base model for the tutorial are already described in the existing Mevea software tutorials and the topic of flexible body creation was not big enough to compromise the full tutorial. Therefore, it was decided to include customisation of Mevea Solver window as an additional part to the tutorial. To check the quality of the tutorial and see possible ways to improve it, the demo session with Saimaa University of Applied Sciences students was held. The feedback given by the students was taken into account to develop recommendations for the development of Mevea tutorial base: keep PDF documents up-to-date, develop short tutorial videos and focus on concrete rather than abstract machines.

On the whole, the thesis topic proved to be rather challenging, as it required quite good understanding of MBS and knowledge of Mevea software at the same time. However, the demo session held with Saimaa UAS students proved that tutorial was easy and clear enough to follow, which are good qualities for a basic level tutorial.

# References

Averill M. Law, W. D. K., 2000. *Simulation modeling and analysis.* 3rd ed. Boston [MA]: McGraw-Hill.

Baharudin, M. E., 2016. *Real-time simulation of multibody systems with applications for working mobile vehicles.* Lappeenranta: Lappeenranta University of Technology.

Blender Foundation, 2017. *Importing & Exporting Files - Blender Manual.* [Online] Available at: https://docs.blender.org/manual/en/dev/data_system/files/import_export.html [Accessed 7 12 2017].

Briot, S. & Khalil, W., 2015. *Dynamics of Parallel Robots: From Rigid Bodies to Flexible Elements.* Cham: Springer International Publishing.

Hartl, M., 2015. *Learn Enough Tutorial Writing to Be Dangerous.* [Online] Available at: https://www.learnenough.com/tutorial-writing-tutorial [Accessed 16 December 2017].

Haug, E. J. & Deyo, R. C., 1991. *Real-time integration methods for mechanical system simulation.* New York, Springer, p. 352.

Korkealaakso, P., 2017. [Interview] (18 December 2017).

Mevea Ltd., 2016. *About | Mevea.* [Online] Available at: https://mevea.com/about/ [Accessed 15 December 2017].

Mevea Ltd., 2016. *Software for Real-Time Simulation | Mevea.* [Online] Available at: https://mevea.com/products/software/ [Accessed 15 December 2017].

Mevea Ltd., 2017. *Reference manual for Solver Library 7.70.* Lappeenranta: Mevea Ltd..

Meylah, 2009. *8 Tips to Write a Tutorial That Gets You Noticed.* [Online] Available at: https://meylah.com/meylah/8-tips-to-write-a-tutorial-that-gets-you-noticed
[Accessed 16 December 2017].

Mikkola, A., 2016. Simulation of a Mechatronic Machine. Lappeenranta University of Technology. *Lecture notes.*

OpenSceneGraph, 2017. *Features.* [Online] Available at: http://www.openscenegraph.org/index.php/about/features
[Accessed 7 12 2017].

Pfeiffer, F., 2008. *Mechanical System Dynamics.* Corr. 2nd pr. ed. Berlin: Springer.

Princeton University, 1998. *Definition of viscosity.* [Online] Available at: https://www.princeton.edu/~gasdyn/Research/T-C_Research_Folder/Viscosity_def.html
[Accessed 13 December 2017].

Quach, Q., 2008. *11 Essential Tips to Writing the Ultimate Tutorial.* [Online] Available at: https://www.dailyblogtips.com/11-essential-tips-to-writing-the-ultimate-tutorial/
[Accessed 16 December 2017].

Shabana, A., 1998. *Dynamics of multibody systems.* 2nd ed. Cambridge: Cambridge University Press.

Sopanen, J., 2004. *Studies of rotor dynamics using a multibody simulation approach.* Lappeenranta: Lappeenrannan teknillinen yliopisto.

Southpaw, A., 2014. *Writing technical tutorials.* [Online] Available at: http://www.andrewsouthpaw.com/2014/12/23/writing-technical-tutorials-that-dont-suck/
[Accessed 16 December 2017].

Watton, J., 1989. *Fluid power systems.* New York: Prentice Hall.

## List of figures

Figure 1. Simulation studio in Lappeenranta University of Technology................8
Figure 2. Ways to study a system (Averill M. Law, 2000, p. 4) ............................ 10
Figure 3. Simulation functional diagram (Haug & Deyo, 1991, p. 5) .................. 12
Figure 4. a) Cam-shaft mechanism b) spring-mass system ................................ 13
Figure 5. Recursive body position description of multibody system (Baharudin, 2016, p. 31)............................................................................................................ 16
Figure 6. Global and body reference coordinate systems (Shabana, 1998, p. 5) ........................................................................................................................... 18
Figure 7. Rigid body mechanics (Shabana, 1998, p. 11) ..................................... 19
Figure 8. System according to lumped fluid theory (Watton, 1989, pp. 96-97) . 22
Figure 9. Kinematic viscosity dependence on pressure and temperature (Watton, 1989, p. 27)............................................................................................................ 23
Figure 10. Rigid container with two volumes ........................................................ 24
Figure 11. Deformable body coordinates (Shabana, 1998, p. 16) ...................... 29
Figure 13. Initial model in tutorial ............................................................................. 34
Figure 14. a) Body graphics for Truck_Body b) Collision graphics for Truck_Body ........................................................................................................................... 36
Figure 15. Bodies of Truck model ............................................................................ 37
Figure 16. Mass properties of body Truck_Pillar ................................................... 38
Figure 17. Topology of hydraulic circuit for Truck model (courtesy of Timo Julkunen, Etteplan)............................................................................................. 46
Figure 18. JibBoom with $E = 2 \times 1011Pa$ and $E = 2 \times 1010Pa$......................... 53
Figure 19. Before and after changing Bag mass in parameter window .............. 54
Figure 20. Plot example produced in Mevea Solver ............................................. 56
Figure 21. Mevea Solver window with workspace loaded from the saved file .. 57

## List of tables

Table 1. Summary of joint types (Mevea Ltd., 2017, pp. 20-27) .......................... 21
Table 2. Selected hydraulic components and associated parameters (Mevea Ltd., 2017, pp. 57-91) .......................... 27
Table 3. Parameters of the body Truck_Pillar .......................... 37
Table 4. Coordinate Systems for joint between Ground and Truck_Pillar .......................... 39
Table 5. Names and joint types in Truck model .......................... 40
Table 6. Parameters of constraint Body_Pillar .......................... 40
Table 7. Parameters of ForceT_Lift .......................... 41
Table 8. Parameters of ForceR_Pillar .......................... 42
Table 9. Parameters of rotational forces ForceR .......................... 42
Table 10. Parameters of Dummy_PillarCylinder .......................... 44
Table 11. Parameters of collision Ground_Bag .......................... 44
Table 12. Parameters of Collision_Friction .......................... 45
Table 13. Parameters of Vol_MainPump .......................... 47
Table 14. Parameters of Hose_Slew_A .......................... 47
Table 15. Parameters of L90Slew .......................... 48
Table 16. Parameters of SlewCyl_CraneSlew .......................... 48
Table 17. Parameters of Spline_CylinderFriction .......................... 49
Table 18. Parameters of Cyl_Lift .......................... 49
Table 19. Parameters of load sensing pressures .......................... 50
Table 20. Parameters of PRV_MainPump .......................... 50
Table 21. Parameters of MainPump .......................... 51
Table 22. Parameters of Input_Slew .......................... 51
Table 23. Parameters of JibBoom responsible for flexibility of the body .......................... 52

# Appendix 1. Mevea tutorial

## Introduction

This tutorial introduces flexible body concept and how flexible bodies are defined in Mevea Modeller. At the same time, this tutorial covers the creation of plots and editing of modelling parameters that are used in research and development.

It is recommended to complete at least Tutorial 1 and understand the features presented there. The background knowledge of flexible body concept is beneficial, but not required to complete this tutorial. This tutorial does not explain the theory behind flexible body concept, but instead shows how flexible bodies are created in Mevea Modeller and how they behave in Mevea Solver.

You can find a model for this tutorial in the folder Mevea_truck called *Truck.mvs*. This file is used as the initial model for this tutorial.

*Figure 1: Tutorial model preview in Modeller's Preview Window*

# Creation of flexible body

Before starting the process of flexible body creation, it is important to understand what is flexible body and what is the difference between flexible bodies and rigid bodies.

By default, all bodies created in Mevea Modeller are rigid bodies. This means that small deformations and rotations are not taken into account when running the simulation. Flexible bodies allow to consider these transformations in the analysis.

To run flexible body model in Mevea Solver, the user needs to specify Flexible Files. These files will replace definition of mass, moments and products of inertia, centre of mass and inertia definition frame for a rigid body.

It is important to understand that the flexible bodies simulation requires a lot of computational power and may heavily affect the output. Therefore, flexible bodies should be used carefully and only when needed. In this tutorial, only the crane boom is defined as a flexible body, while other bodies remain rigid, because the deformation of those bodies is not relevant to the study of the boom deformation.

## Adding flexibility to the Jib_Boom

Open file *Truck.mvs* in Mevea Modeller. You can run the Dynamic Simulation by pressing *Ctrl+D* in order to check the model. All the body definitions, forces, joints, hydraulic components and inputs are already included in the model. The only difference to the final model is that the crane boom (Jib_Boom) acts as a rigid body.

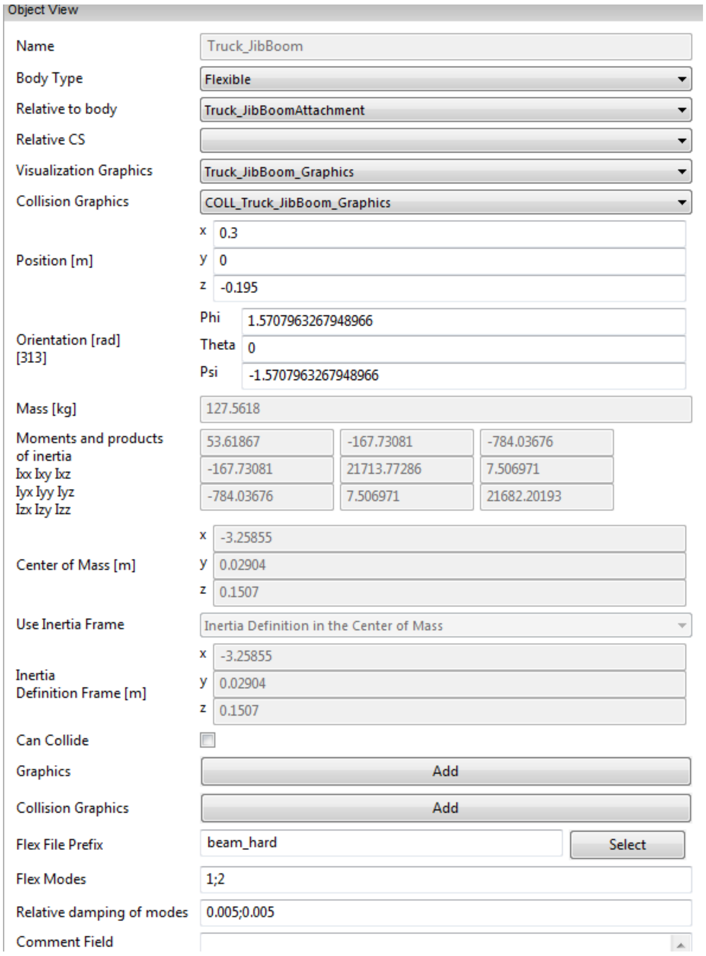
To add the flexibility to *Jib_Boom* body, do the following:

1. In Mevea Modeller go to **Bodies** and double click **Jib_Boom**.
2. Change parameters in **Object View** as follows:

| | |
|---|---|
| Body type | Flexible |
| Flex File Prefix | beam_hard |
| Flex Modes | 1;2 |
| Relative damping of modes | 0.005;0.005 |

*Table 1: Flexible body parameters*

3. Leave the rest of the parameters unchanged.
4. Double click on the name of the body to update the parameters.



Name: Truck_JibBoom
Body Type: Flexible
Relative to body: Truck_JibBoomAttachment
Relative CS:
Visualization Graphics: Truck_JibBoom_Graphics
Collision Graphics: COLL_Truck_JibBoom_Graphics
Position [m]: x 0.3, y 0, z -0.195
Orientation [rad] [313]: Phi 1.5707963267948966, Theta 0, Psi -1.5707963267948966
Mass [kg]: 127.5618

| Moments and products of inertia<br>Ixx Ixy Ixz<br>Iyx Iyy Iyz<br>Izx Izy Izz | | |
|---|---|---|
| 53.61867 | -167.73081 | -784.03676 |
| -167.73081 | 21713.77286 | 7.506971 |
| -784.03676 | 7.506971 | 21682.20193 |

Center of Mass [m]: x -3.25855, y 0.02904, z 0.1507
Use Inertia Frame: Inertia Definition in the Center of Mass
Inertia Definition Frame [m]: x -3.25855, y 0.02904, z 0.1507
Can Collide: ☐
Graphics: Add
Collision Graphics: Add
Flex File Prefix: beam_hard (Select)
Flex Modes: 1;2
Relative damping of modes: 0.005;0.005
Comment Field:

*Figure 2: Parameters of body Jib_Boom after adding flexibility*

Now you can try running the simulation again. Flexible files are already included for this tutorial model, however whenever saving the model with **Body type: Flexible** you need to make sure that the flexible files are specified in the respective field (*Flex File Prefix*) and they are in the same folder as your simulation model.

The model should not change its graphical appearance. In case some of the textures on some of your bodies were replaced by default violet, do the following:

1. Go to Graphics -> name of the body that lost the texture
2. Scroll down the Object View and change the following parameters:

| | |
|---|---|
| Shader name | Shaders/self_contained/dsn |
| Link default shaders | no |

*Table 2: Changing shaders parameters*

3. If there are more bodies that lost the texture, repeat the procedure with them.

The model in dynamic simulation mode should look as in Figure 3 presented below:

*Figure 3: Initial position of the model in Mevea Solver*

You can now observe the changes in the Jib_Boom behaviour. You can notice that the boom bends in both horizontal and vertical planes while lifting and moving the bag.

In the following chapters of the tutorial, ways to change the basic parameters and improve the workspace are discussed.

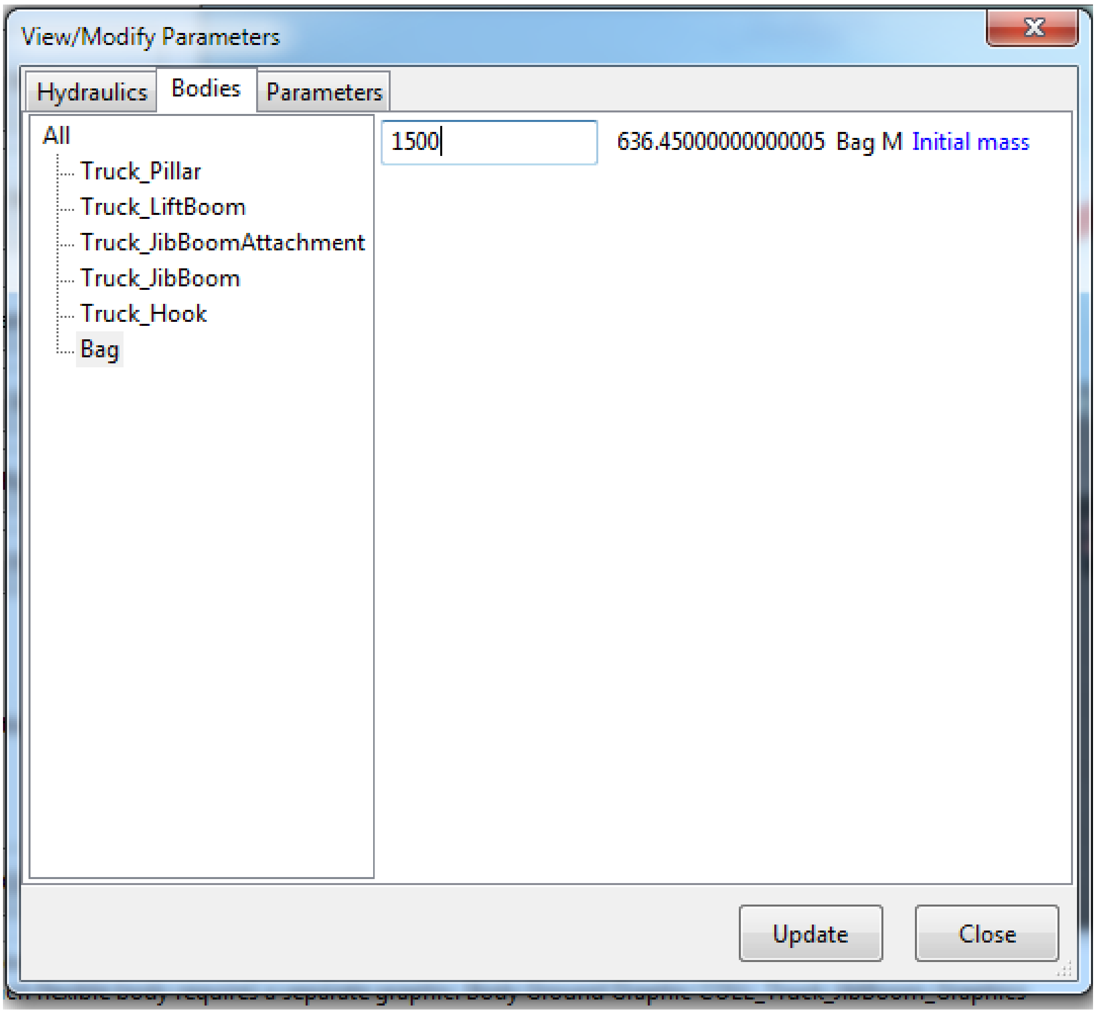
## Adjusting Solver Window

In this part of the tutorial, different ways to customise Mevea Solver output are discussed. The first part explains the way to change basic model parameters. The second part shows how to record the control input. The third part defines the process of plots creation. Finally, the way to save created customised Solver window is described.

### Changing Basic Parameters

Mevea Solver allows changing basic model parameters directly in Solver window without having to edit them in Modeller and reloading the dynamic simulation. In case of this tutorial, the bag mass is a parameter that might be interesting to change. To change the mass of the bag, do the following:

1. Open the model in Mevea Solver.
2. In the top bar of Solver window go to *Control -> Parameter View.*
3. Click **Bodies** tab and choose **Bag**.
4. Change the mass of the bag to 1500 in the respective field.
5. Click **Update** to save the changes and **Close** to close the window.



1500 636.45000000000005 Bag M Initial mass

*Figure 4: Changing bag mass in Solver's parameter view*

6. You will be able to see the changes immediately after clicking **Update**.

*Figure 5: Flexible boom with default bag mass and 1500kg*

Note that changing the parameter in Parameter View window will not change the parameter value in the Modeller file. If the parameter value was changed, it will be lost after closing Solver window.

## Recording Input

Another useful feature of Solver is recording input signals. This option can be used to test the model with the same input but different parameters, for instance, after changing the mass of the bag or pressure in the system.

To record the input, follow these steps:

1. Open the simulation in Mevea Solver
2. Click **Start**
3. In the top bar, go to *Control -> Input Control -> Start record*

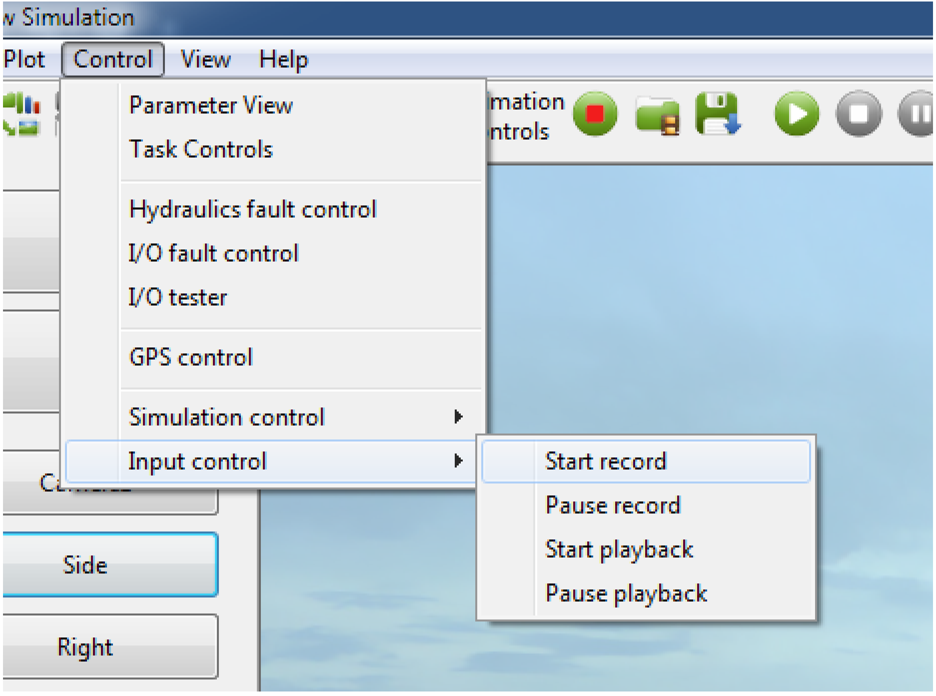


*Figure 6: Recording input*

4. In the pop-up window, specify the name of the file for input record and its location
5. After you have performed all the desired actions, go to *Control -> Input Control -> Start record* again.
6. Now you can replay the record by clicking *Control -> Input Control -> Start playback*. Specify the location of the record file and hit *“Open”*.

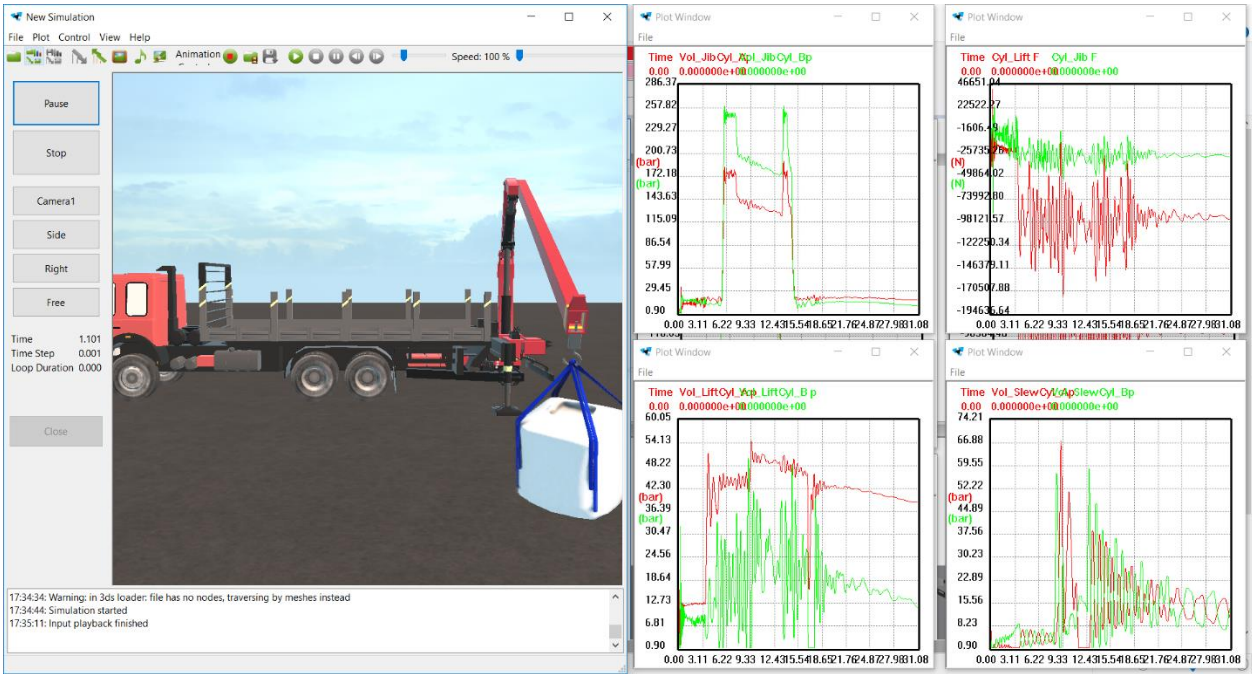
Note that this way only the input signals are recorded, not the actual locations. Therefore, if the initial position of the body before playback is different from the initial position of the body during recording, the resulting simulations will not be the same. That is why it is advised to start the recording and playback from the same initial position, for example after the simulation is restarted.

## Creating Plots

Recorded inputs allow the user to repeat exactly the same actions, however it is impossible to see how the internal design parameters change during the simulation. Mevea Solver lets the user to plot the values and save them for later use. To enable plotting, follow the steps:

1. Open your model in Mevea Solver.
2. Click *Plot -> Create plot* in the top bar or icon in the top bar or *Ctrl+P*.
3. Tick the values that you would like to show on the same plot. The list of values that are available for plotting is listed in Reference Manual after each type of element in table Printable variables.
4. All the values that you tick before clicking **Apply** are plotted on the same graph. To create more plots, tick desired values and click **Apply** again.

5. Click **OK** to close the window and go back to simulation window.
6. You can rearrange the plot windows as you wish. The plotting will start together with the simulation.

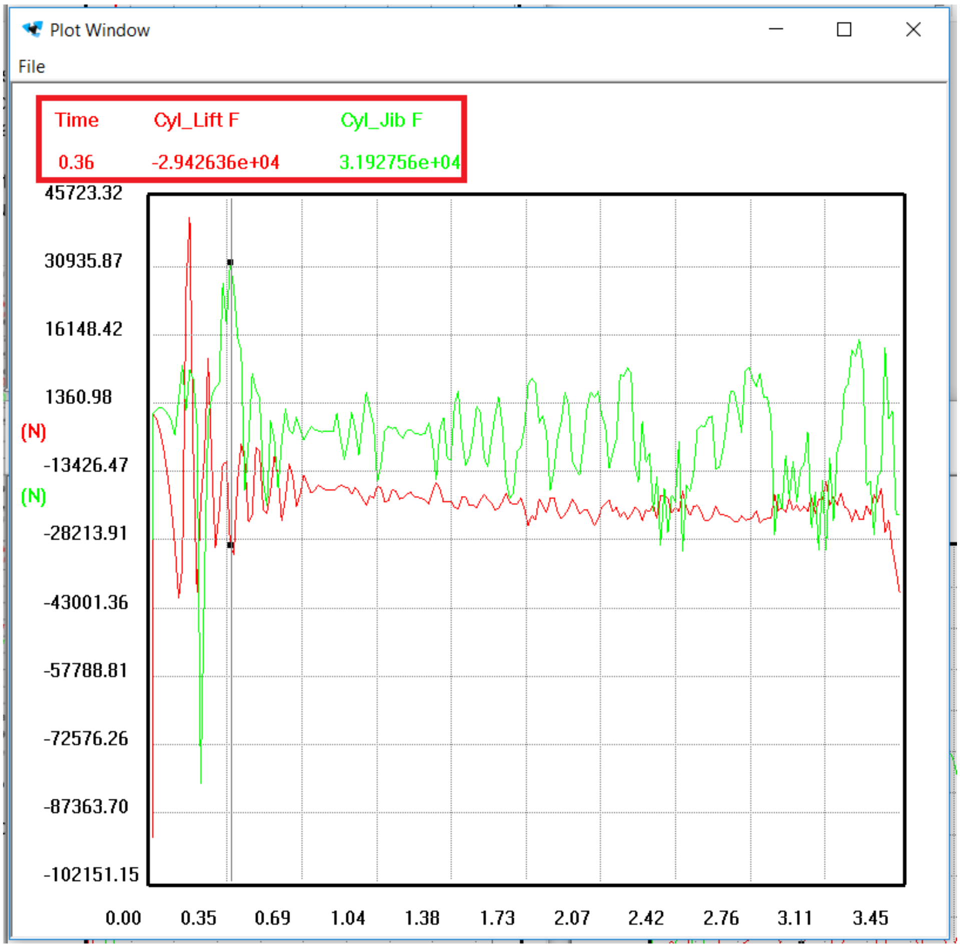


Time 1.101
Time Step 0.001
Loop Duration 0.000

17:34:34: Warning: in 3ds loader: file has no nodes, traversing by meshes instead
17:34:44: Simulation started
17:35:11: Input playback finished

*Figure 7: Simulation window and four plotting windows in Solver*

7. In the plot, you can hover over a specific point to find the value in that point.



| Time | Cyl_Lift F | Cyl_Jib F |
|---|---|---|
| 0.36 | -2.942636e+04 | 3.192756e+04 |

*Figure 8: Cylinder lift and cylinder jib forces at 0.36s after the start of the simulation*

## Saving and Loading Workspace

Plots are useful tool for research and development of the new product, especially in real-time simulation, as they allow to change the parameters of the internal components and see the changes in the plots immediately. However, opening the same plot windows each time the simulation is started, may take a lot of additional time. Mevea Solver allows the user to save the simulation window, the plot types shown and the position of these windows for later use. To use this feature, follow the steps:

1. Close all previously opened **Mevea Modeller** and **Mevea Solver** windows.
2. Open the model with **Mevea Solver**.
3. Create the desired plot windows.
4. Rearrange the windows in the desired positions.
5. In the top bar, click *File -> Save workspace* or “Save workspace” icon

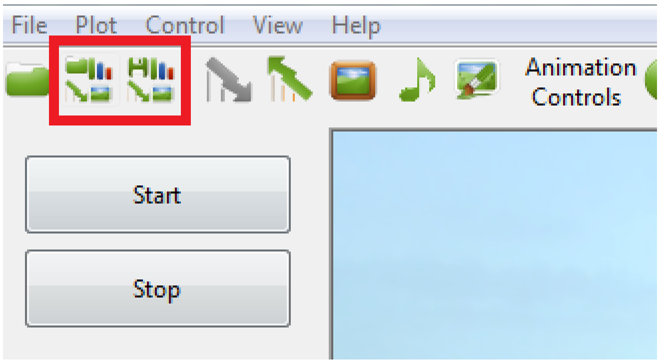


*Figure 9: Saving and loading workspace icons*

6. Specify the name and the location for the configuration file. Click **Save** to finish the process.
7. To load the workspace, close all previously opened **Mevea Solver** windows first.
8. Open **Mevea Solver** with a desktop or Start Menu shortcut.
9. Click *File -> Load workspace* or “Load workspace” icon
10. Select the workspace file in a pop-up window. Click **Open** to load the workspace.
11. The loaded workspace will have the simulation window together with all the plots created before.

Note that workspace file should be saved in the root folder of the model to avoid possible location conflicts.

Now you have the tools to run flexible body model, change basic parameters and track how these changes affect the model.

# Appendix 2. Feedback form for tutorial demonstration with Saimaa UAS students

## Overall impression:

| Question: | Answer type: |
|---|---|
| The purpose of the tutorial was well-explained | Linear scale 1-5 |
| The tutorial was easy to understand | Linear scale 1-5 |
| I was able to understand all the steps of the tutorial from the first time | Linear scale 1-5 |
| The tutorial steps were in logical order | Linear scale 1-5 |
| I will be able to utilise the knowledge obtained in this tutorial for future Mevea software usage | Linear scale 1-5 |
| The language of instructions was clear and understandable | Linear scale 1-5 |
| I will be able to reproduce the same result myself without additional guidance | Linear scale 1-5 |
| I found this tutorial useful | Linear scale 1-5 |
| Any comments on this section or tutorial in general? Something that you liked/disliked in general? | Long-answer text |

## Flexible body implementation:

| Question: | Answer type: |
|---|---|
| I knew about flexible body concept before this tutorial | Linear scale 1-5 |
| I understand the difference between flexible and rigid bodies simulation after doing the tutorial | Linear scale 1-5 |
| It was easy to understand the way the flexibility is added to already existing rigid body in Mevea Modeller | Linear scale 1-5 |
| I would like to know more about the way flexible files are created | Linear scale 1-5 |
| I believe I will use flexible bodies in my future simulations with Mevea software | Linear scale 1-5 |
| Any comments/suggestions? Something that you liked/disliked about the flexible body part of the tutorial? Anything you would like to add to this part of the tutorial? | Long-answer text |

## Solver window customisation:

| Question: | Answer type: |
|---|---|
| I knew about the ways of customising Mevea Solver window before taking this tutorial | Linear scale 1-5 |
| I find the explained customisation features useful for future modelling with Mevea | Linear scale 1-5 |
| I would like to learn more about Solver window customisation | Linear scale 1-5 |

| | |
|---|---|
| Any comments/suggestions? Something that you liked/disliked about the Solver window customisation part of the tutorial? Anything you would like to add to this part of the tutorial? | Long-answer text |

## Closing paragraphs (general questions):

| **Question:** | **Answer type:** |
|---|---|
| I am... (name the study year) | Multiple choice |
| Did you use Mevea software before this tutorial? | Yes/No |
| If yes, how long have you used it? | Multiple choice |
| I prefer to get tutorials (you can choose multiple options): | Checkboxes |
| I believe I will use Mevea software in the future | Linear scale 1-5 |
| I would like to continue working with real-time simulation software (doesn't have to be Mevea) | Linear scale 1-5 |
| Any ideas for future tutorials for Mevea software? | Long-answer text |
| Final comments/thoughts/ideas? | Long-answer text |

# Appendix 3. Summary of feedback form responses

| **Question** | **Average** | **Median** | **Min** | **Max** | **Number of responses per each option** |
|---|---|---|---|---|---|
| The purpose of the tutorial was well-explained | 3.77 | 4 | 2 | 5 | 5: 2<br>4: 7<br>3: 1<br>2: 1<br>1: 0 |
| The tutorial was easy to understand | 4.46 | 4 | 4 | 5 | 5: 6<br>4: 7<br>3: 0<br>2: 0<br>1: 0 |
| I was able to understand all the steps of the tutorial from the first time | 4.38 | 5 | 2 | 5 | 5: 8<br>4: 3<br>3: 1<br>2: 1<br>1: 0 |
| The tutorial steps were in logical order | 4.31 | 4 | 3 | 5 | 5: 6<br>4: 5<br>3: 2<br>2: 0<br>1: 0 |
| I will be able to utilise the knowledge obtained in this tutorial for future Mevea software usage | 3.85 | 4 | 3 | 5 | 5: 4<br>4: 3<br>3: 6<br>2: 0<br>1: 0 |
| The language of instructions was clear and understandable | 4.96 | 5 | 4 | 5 | 5: 9<br>4: 4<br>3: 0<br>2: 0<br>1: 0 |
| I will be able to reproduce the same result myself without additional guidance | 4.15 | 4 | 3 | 5 | 5: 4<br>4: 7<br>3: 2<br>2: 0<br>1: 0 |
| I found this tutorial useful | 4 | 4 | 2 | 5 | 5: 3<br>4: 8<br>3: 1<br>2: 1<br>1: 0 |
| I knew about flexible body concept before this tutorial | 2 | 2 | 1 | 4 | 5: 0<br>4: 3<br>3: 0<br>2: 4 |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | 1: 6 |
| I understand the difference between flexible and rigid bodies simulation after doing the tutorial | 4.31 | 4 | 3 | 5 | 5: 5<br>4: 7<br>3: 1<br>2: 0<br>1: 0 |
| It was easy to understand the way the flexibility is added to already existing rigid body in Mevea Modeller | 3.92 | 4 | 2 | 5 | 5: 4<br>4: 5<br>3: 3<br>2: 1<br>1: 0 |
| I would like to know more about the way flexible files are created | 3.31 | 3 | 2 | 5 | 5: 1<br>4: 4<br>3: 6<br>2: 2<br>1: 0 |
| I believe I will use flexible bodies in my future simulations with Mevea software | 3.38 | 4 | 1 | 5 | 5: 2<br>4: 5<br>3: 3<br>2: 2<br>1: 1 |
| I knew about the ways of customising Mevea Solver window before taking this tutorial | 1.77 | 1 | 1 | 4 | 5: 0<br>4: 2<br>3: 0<br>2: 4<br>1: 7 |
| I find the explained customisation features useful for future modelling with Mevea | 4 | 4 | 2 | 5 | 5: 3<br>4: 8<br>3: 1<br>2: 1<br>1: 0 |
| I would like to learn more about Solver window customisation | 3.15 | 3 | 1 | 5 | 5: 1<br>4: 4<br>3: 5<br>2: 2<br>1: 1 |
| I believe I will use Mevea software in the future | 3 | 3 | 2 | 4 | 5: 0<br>4: 4<br>3: 5<br>2: 4<br>1: 0 |
| I would like to continue working with real-time simulation software (does not have to be Mevea) | 3.31 | 3 | 2 | 5 | 5: 1<br>4: 5<br>3: 4<br>2: 3<br>1: 0 |