Tomás Correia Coimbra Mano
Licenciado em Ciências da Engenharia Mecânica

Real-time Simulation of a Mobile Machine

Dissertação para obtenção do Grau de Mestre em Engenharia Mecânica

Orientador: Professor Doutor Alberto Martinho,
Professor Auxiliar, FCT-UNL
Co-orientador: Professor Doutor Aki Mikkola,
Professor, LUT

Júri:

Presidente:	Prof. Doutor António José Freire Mourão
Arguente:	Prof. Doutor Tiago Alexandre Narciso da Silva
Vogal:	Prof. Doutor Alberto José Antunes Marques Martinho

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Outubro, 2019

Real-time simulation of a mobile machine

Acknowledgements

This work was made possible due to the efforts of some people I would like to acknowledge.

First, to all the people I got to know at LUT, for the help given, for the possibility to develop this thesis and for the chance to learn more about real-time simulation. To professor Aki, who made it possible for me to develop my work abroad, for all the help given related with the work and with life in Finland. To Qasim and Nima, for making it easier to understand the software and to perform the necessary tests.

To all my friends, who were always a phone call away, to answer any questions and for the support given throughout these past months when this work seemed to be impossible to complete. For my friends, developing their works at the same time as me, supporting is key, and thank you for all the love.

To my supervisor in Portugal, professor Alberto Martinho, for all the help given, for going along in this journey with my, always available to answer my questions.

Finally, to my family, my mother and father, my grandparents, and all the others involved in making my time in Finland possible, and to my brothers, in spite of not being there with me, were always there for me.

Abstract

Real-time simulation is becoming a useful tool to improve products and efficiency in product development. Being able to perform real-time dynamic analysis on a virtual prototype helps engineers improve products faster, with a reduction in the associated costs. The possibility to analyse complex systems using powerful software that enable the creation of simpler models which can accurately represent these systems, allows the use of tools to perform numerical integrations and dynamic analysis in a faster, simpler, and easier way.

With the growth of real-time simulation, gamified simulations are also improving and becoming a tool for product development. The goal of turning mechanical simulations into game-like environments is to encourage users to test products, providing important information to manufacturers, and making the product development to be more customer oriented.

This thesis explores two topics. The first part consists of developing a dynamic analysis of a forklift, using the existing theoretical definitions and implementing them using Mevea software. The system's behaviour will be studied by imposing different conditions and comparing the outcomes (differences in the behaviour of the machine). The second part aims at using the developed model to create a gamified simulation. The simulation should become user-friendly with a goal and tasks to be performed, encouraging users to test the created model. All this is achieved by adding different game elements. Ultimately, the goal of the simulation is to obtain information from the users about their experience, to improve the model and, consequently, the forklift, to suit the needs of the users.

The goal of this thesis is to understand the benefits of real-time dynamic analysis, as well as to evaluate gamified simulations as a tool for engineers to get important information during product development.

Keywords: real-time simulation, dynamic analysis, gamified simulation, game elements, users

Resumo

A simulação em tempo real tem vindo a tornar-se uma ferramenta útil na melhoria de produtos e no aumento da eficiência na fase de desenvolvimento de um produto. A possibilidade de realizar uma análise dinâmica em tempo real a um protótipo virtual permite aos engenheiros melhorar os produtos mais rapidamente, e com uma redução de custos associada. A possibilidade de analisar sistemas complexos utilizando software que permite a criação de modelos simplificados que conseguem representar o sistema de maneira eficaz, permite o uso de ferramentas para realizar as integrações numéricas e a análise dinâmica de maneira mais eficiente, simples e rápida.

Com o crescimento da simulação em tempo real, as simulações estilo "jogo" também têm vindo a melhor, tendo se tornado uma ferramenta útil para o desenvolvimento de produtos. O objetivo de transformar as simulações em ambientes semelhantes a jogos é encorajar os utilizadores a testarem os produtos e a fornecer feedback aos fabricantes, permitindo que o produto seja desenvolvido tendo em conta as necessidades dos utilizadores.

Esta dissertação explora dois tópicos. A primeira parte consiste em realizar uma análise dinâmica a uma empilhadora, recorrendo a formulações teóricas existentes, e implementando-as com recurso ao software Mevea. O comportamento da empilhadora deve ser estudado, considerando diferentes casos e comparando os resultados (alterações no comportamento da máquina). A segunda parte consiste em utilizar o modelo desenvolvido na criação de uma simulação estilo "jogo". Esta simulação deve ser centrada no utilizador e deve conter um objetivo claro e as instruções necessárias, encorajando os utilizadores a testarem o modelo. Tudo isto é feito através da introdução de elementos de jogo. Em última análise deverá ser possível avaliar as experiências dos utilizadores e utilizar a informação proveniente das simulações para melhorar a empilhadora.

Este trabalho tem como objetivo a perceção dos benefícios da análise dinâmica em tempo real, bem como avaliar como as simulações estilo "jogo" podem fornecer informação importante aos projetistas.

Palavras chave: simulação em tempo real, Análise dinâmica, simulação estilo "jogo", elementos de jogo, utilizadores

Table of contents

List of Figures

List of tables

List of Symbols and Abbreviations

Initials and acronyms

DAE – Differential algebraic equations

ODE – ordinary differential equations

DOF’s – Degrees of Freedom

CAD – Computer Aided Design

CAM – Computer Aided Manufacturing

CASE – Computer Aided Software Engineering

CAS – Computer Aided Simulation

VR – Virtual Reality

Vectors/matrix

$\mathbf{q}/\mathbf{r}_{I_i}$ – Global position of a particle

$\dot{\mathbf{q}}/\dot{\mathbf{r}}_{I_i}$ – Velocity of a particle

$\ddot{\mathbf{q}}/\ddot{\mathbf{r}}_{I_i}$ – Acceleration

$\mathbf{R}_I$ – Global position of a body

$\dot{\mathbf{R}}_I$ – Velocity of a body

$\ddot{\mathbf{R}}_I$ – Translational acceleration of a body

$\bar{\mathbf{u}}_{I_i}$ – Position a particle relatively to the local coordinate system

$\mathbf{A}_i$ – Coordinates transformation matrix

$\dot{\mathbf{A}}_i/\mathbf{A}_{\theta,A}$ – Derivative of rotation matrix

$\boldsymbol{\omega}_i$ – Angular velocity of a body

$\dot{\boldsymbol{\omega}}_i$ – Angular acceleration of a body

$\Delta\mathbf{q}$ – displacement

$\delta\mathbf{r}_A$ – Virtual displacement

$\delta\mathbf{W}_A$ – Virtual work

$\dot{\mathbf{q}}^*$ - Virtual velocity

$\boldsymbol{\omega}^*$ - Virtual angular velocity

$\boldsymbol{M}$ – Mass Matrix

$\mathbf{I}_n$ – Identity matrix of index n

$\mathbf{J}_\mathbf{0}$ – Rotation inertia tensor

C – Quadratic velocity vector

F –Force vector

Q – Generalized forces vector

T - Torque

$\mathbf{W}^*$ - Virtual Power

$\boldsymbol{\lambda}$ – Lagrange multipliers

$\boldsymbol{\Phi}_\mathbf{q}/\boldsymbol{\Phi}(\mathbf{q},t)$ – Displacement constraints

$\dot{\boldsymbol{\Phi}}(\dot{\mathbf{q}},t)/\dot{\boldsymbol{\Phi}}(\mathbf{q}\dot{,}\mathbf{q},t)$– Velocity constraints

$\mathbf{CqC}_\mathbf{q}$ – Jacobian matrix

$\mathbf{C}_\mathbf{q}$ – Jacobian matrix

$\dot{\boldsymbol{\Phi}}(\dot{\mathbf{q}},t)$ – velocity constraints

R – Velocity transformation matrix

$\dot{\mathbf{R}}$ – Time derivative of R

$\dot{\mathbf{z}}$ – Relative joint velocity

$\ddot{\mathbf{z}}$ – Relative joint acceleration

$\dot{\mathbf{z}}^*$ - Independent velocities vector

$\mathbf{F}_\mathbf{n}$ – Normal forces

$\mathbf{F}_\mathbf{t}$ – Tangential forces

d – location of the collision

$\mathbf{V}_n$ – Relative velocity of a collision

n – normal vector

$\mathbf{F}_\mathbf{\mu}$ – Friction Force

Scalar values

θ_i – Angle between the local coordinate and the global coordinate systems

$\dot{\theta}_i$ – Angular velocity of a particle

$\ddot{\theta}_i$ – Angular velocity of a particle

$\boldsymbol{\omega i}$ – skew-symmetric matrix of ωix – Displacement along the X axis

y – Displacement along the Y axis

t_i – Time at instant i

Δt – time step

$\dot{x}^*$ - Virtual velocity along the X axis

$\dot{y}^*$- virtual velocity along the Y axis

m - number of constraint equations

n – Number of dependent coordinates

f – Number of degrees of freedom

$O(h^{p+1})$ – local truncation error

$O(h^p)$ – method's order

ρ - Density

B – Bulk modulus

B_e – Effective bulk modulus

Q_i – Flow rate

v - Velocity

k – Spring coefficient

s – Damping coefficient

V - Volume

dV – Volume change

dp – Pressure change

Re – Reynolds number

D_h - Hydraulic diameter

υ - Viscosity

$\dot{m}$ – mass flow rate

C_v – Flow rate constant

X_0 – Relative spool position

U_{ref} – Reference voltage signal

τ – Time constant

Note: All the vector and matrixes presented throughout this work are presented in normal bold characters while the scalar values are presented in italic without bold

1 *Introduction*

The first chapter starts by introducing the work that will be developed, the approach to the problem, the objectives proposed and how this thesis will be developed.

1.1 Context

Mechanical engineering is often associated with innovation. With industrial growth and development, productivity, efficiency, and competitiveness are now issues that engineers must face. It is necessary to improve how engineers work and how projects and products are developed and tested. It is necessary to work with the existing tools, but in a better way, to boost productivity levels. The processes need to be fully understood in order to increase the efficiency of the project and, consequently, of the product (Larsson, 2001).

Computational mechanics, and more specifically multibody system dynamics, is one of the available tools used for performing dynamic analysis of complex systems (where the system's behaviour is analysed, having the time as a constraint). This area is responsible for dealing with the kinematics and the dynamics of systems, numerical methods, optimization, and control issues (Schiehlen, 2006).

Combining multiple areas in computational mechanics, like multibody dynamics and computer-aided design (CAD) makes it possible for engineers to work faster and in a more efficient way. The possibility to use a tool to build a trustworthy model (to represent a product) together with the possibility to test that model as a virtual prototype allows engineers to obtain a better and faster understanding of most of the problems that might exist when testing different solutions for a project. This way, virtual models become virtual prototypes that can represent the properties of a physical prototype accurately, and the analysis becomes faster and can commit fewer errors, compared with traditional methods.

In the industry field, the demand for better, safer, and more efficient solutions made engineers change the way they develop and test solutions. The possibility of a faster and more efficient analysis (using computational tools) with a model made engineers turn to real-time simulations. This type of simulation appears as a solution because it allows the model to be tested in real time, eliminating time-delay as a constraint.

Real-time simulation reduces the necessity of creating many physical prototypes, which most of the times are tested with the wrong parameters and have the wrong properties. Physical prototypes often need multiple versions to perform different tests, until the best version is achieved. A virtual prototype allows testing the different parameters and conditions without the need of creating numerous versions with different physical properties, eliminating the costs associated with the process and achieving the same outcome faster. It also allows for a better understanding of how the different systems interact, making it possible to make more efficient products. The three major benefits related to efficiency are reduced costs, accelerated schedule, and improved product quality (US Army, 2013).

Figure 1.1 shows the benefits of modelling and simulation when applied to product development. The major benefits regarding the project are testing, user-engagement, and safety.

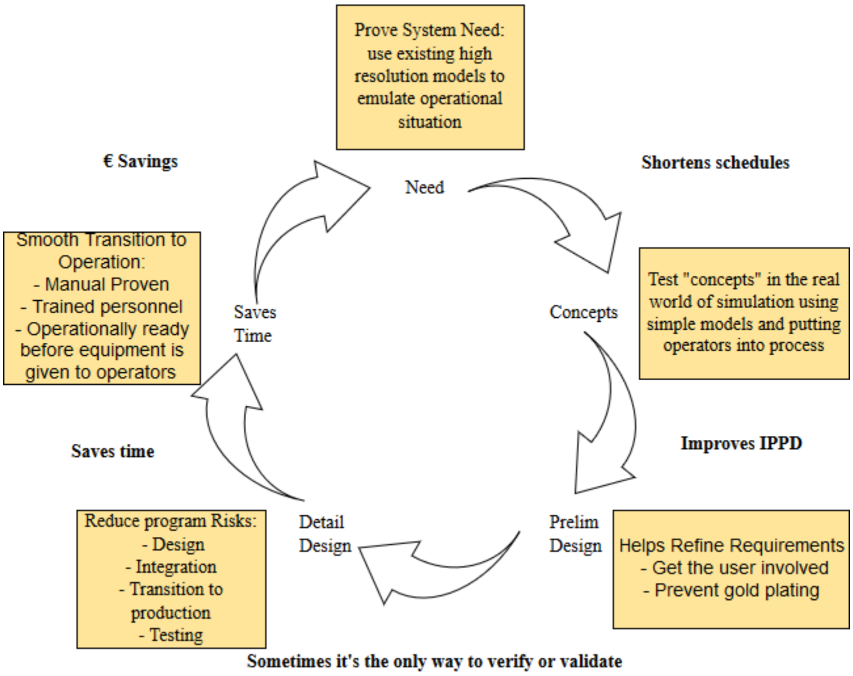

Prove System Need: use existing high resolution models to emulate operational situation
€ Savings
Shortens schedules
Need
Smooth Transition to Operation:
- Manual Proven
- Trained personnel
- Operationally ready before equipment is given to operators
Saves Time
Concepts
Test "concepts" in the real world of simulation using simple models and putting operators into process
Improves IPPD
Saves time
Reduce program Risks:
- Design
- Integration
- Transition to production
- Testing
Detail Design
Prelim Design
Helps Refine Requirements
- Get the user involved
- Prevent gold plating
Sometimes it's the only way to verify or validate

Figure 1.1 Advantages of modelling and simulation, adapted from (US Army, 2013)

Mobile machines of every type are essential in modern industry. Every machine needs to keep improving the safety standards and its efficiency in order to satisfy the rules and safety parameters. The demand leads to the necessity of developing better ways to improve mobile machines, simplifying its controls and improving overall features.

To help developing better machines and better professionals, simulation models with real-time simulation applications are being developed. It becomes easier to evaluate the machine due to its complexity and to improve the design or functioning features.

Mechanical real-time simulation has one problem. It can become boring when testing the products. If the simulation lacks interaction with whoever tests it, it can lead to a loss of interest, producing average results and not representing a sucessful test. The best way to create a simulation that does not become boring is to keep improving the interaction features (game elements or motivational factors).

Because the simulations can be tested by users to get feedback and to improve the product in a more customer-oriented way, they should be appealing and keep users interested in them. If these simulations are boring, users might not feel interested in testing them, making the feedback available less trustful, delaying the improvement process. Engineers, to solve this problem, started introducing game elements to the simulations, making them more appealing but maintaining realism. The introduction of game elements in simulations allows better feedback from users, making the simulation more engaging, motivating users to perform any given tasks. This way, getting more and better information about the products, engineers can evaluate the products from the customer´s point of view, being able to think of products for a more customer-oriented production.

Since the beginning of the 21st century, articles about computationally aided simulation and real-time simulation, as well as game-like environment simulation, have been published, proving the growing interest in this area.

1.2 Objectives

This thesis' objective is to understand how multibody dynamic analysis works, to analyse a mobile machine (a forklift) and simulate it, using real-time simulation software, MEVEA, and add game elements to the simulation, using UNITY, in order to comprehend how different bodies interact with each other, and how gamification helps getting better and more accurate information about a product.

Some questions like the ones presented below should be answered at the end of this thesis:

- What are the benefits of real-time dynamic analysis?
- What game elements can be added so that the simulation is effective and motivates users to complete tasks?
- How manufacturers can use the information obtained through game-like simulations?

To help understand and answer these questions, this thesis includes a methodology section where the key concepts, as well as the theoretical definitions and related work are presented before addressing the forklift study case. After the results are presented and discussed, it should be possible to answer the questions presented.

1.3 Research Methodology

Figure 1.2 describes the existing relations involving all necessary systems to build a simulation that truly represents a mobile machine. The external forces acting in the system, as well as the internal forces, make the system behave in a certain way. Position, velocity, and acceleration are necessary parameters to describe this behaviour and to predict what actuators need to do. All this information can be passed to a control panel that enables the user to see and control the system.

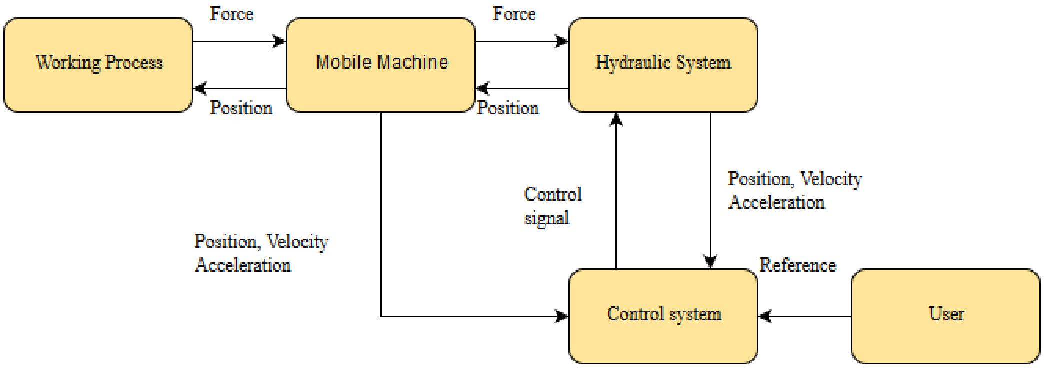


Figure 1.2 Systems needed to build a simulation for a mobile machine, adapted from (Mikkola, Lecture1_m, 2018)

To understand how a complex system behaves, it is necessary to comprehend how the different bodies behave individually and as a whole. It is also necessary to understand what joints can be used and what constraints (and consequently the existing DOF's) are associated with each one of them. This way, it is possible to understand how to perform a dynamic analysis of the system. The existing theoretical definitions necessary to perform the dynamic analysis are available in most literature. Because the definitions derive from the basic physics and mathematical equations, they are accepted by most theorists and do not need much discussion. It is only necessary to comprehend such definitions and how to use them.

It is also mandatory to comprehend what real-time simulation is, how to create a model to represent a system, and what game elements are and how they can help when building a simulation. Game elements have been introduced to real-time simulation recently, and the articles published about the subject suggest they can help to turn real-time simulation more interesting, being worth exploring. There are many ways of creating a model and a simulation, with different game elements, and because of that some research is needed, to decide which game elements have the highest impact and how to build an efficient real-time simulation.

This work is done using two software tools: Mevea software and Unity game-engine. The first one can be used to build a trustworthy representation of a system's behaviour, making it possible to represent different parts, joints, and actuators. The second one can be used to add game elements to a simulation, as well as for visual aspects, making it more real. Both tools can be used to store data, which later can be used to comprehend how systems can work together to increase efficiency. It is essential to understand how different software programs work together using interfaces. The validation of the results obtained will be done using two different ways: the simulation part will be evaluated according to reference values provided for the forklift, and by comparing the results for two different situations, while the simulation and gamification part will be evaluated according to a loop-time analysis, and game-creation features, respectively.

1.4 Thesis Structure

This thesis is divided into seven chapters, each one broken down in smaller sub-chapters.

The first chapter is an introduction to the thesis. It contains the contextualization of the work, the objectives proposed, and how it should be developed.

The second and third chapters are about the methodology and the gamification process. Some concepts used, like multibody dynamics, gamification and simulation are addressed, as well as the different definitions used for the dynamic analysis. Finally, some recent articles are summarized, with the most important parts related to this work being highlighted.

The fourth chapter addresses the practical study case for the work developed in the thesis. The system under investigation is introduced with its configuration. Each part is described and analysed. The model used in the simulation is also introduced in this chapter, together with the load used for the tests. The gamification part is also introduced, with the different game-elements presented, with their objective being explained.

The fifth chapter presents the outcome achieved with a brief explanation while the sixth chapter includes a discussion about the outcome achieved. The research questions should be addressed, as well as negative and positive considerations about the work produced.

The seventh and final chapter is a conclusion. A summary of the work is presented, with some details and curiosities about some aspects of the work developed.

2 *Methodology*

This chapter starts by addressing some concepts like what multibody systems, dynamic simulation, and Computer-aided simulation are. The definitions used to perform the dynamic analysis are addressed with a detailed explanation about them.

2.1 Multibody Systems

A multibody system is a group of rigid or flexible bodies connected to each other. Bodies interact with each other through joints. Systems can be analysed as a whole, or different bodies can be analysed as separate parts. Figure 2.1 shows a multibody system with two bars and one body connected using three joints. The bars' movements are conditioned through the existing constraints introduced by the joints used. For this example, a planar system is used. Using a planar system allows a simpler and better understanding of what DOF's are, as well as constraints, and ultimately, how multibody analysis works. For spatial systems (3D), the definition of the same DOF's, constraints, and other important information can be inferred from the following literature: (de Jalón & Bayo, 2009); (Shabana, 2001); (Nikravesh, Computer-Aided Analysis of Mechanical Systems, 1988).

To perform any analysis, it is necessary to understand what the existing constraints are and their type. Using figure 2.1 as an example, it is possible to see the movements the three bodies can perform: the first bar can only rotate around the joint on the left, while the second bar can rotate around the joint on its left, but its centre of mass can have a translational movement, and so on. The existing constraints can be expressed using differential and algebraic equations. After the number of constraints is established, it is possible to calculate the DOF's of the system. In dynamic problems, the number of DOF's is always greater than one (DOF's > 1).

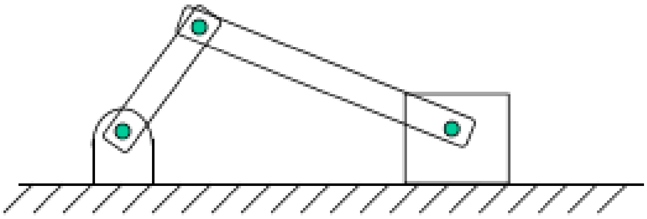


Figure 2.1 Multibody System, adapted from (de Jalón & Bayo, 2009)

Dynamic analysis of any multibody system also relies on a set of parameters, which can be described either in local or global coordinates. These two types of coordinates are explained next.

2.2 Description of rigid bodies

All systems require a set of parameters that uniquely define their position, velocity, and acceleration during the period of analysis. These parameters can be referred to as generalized coordinates. (Roupa, Gonçalves, & Tavares da Silva, 2018). A planar system (2D) consisting of *i* bodies will need 3n generalized coordinates to describe the system while a three-dimensional system (3D) consisting of n bodies will need 6n generalized coordinates to be described. (Mikkola, week2-18mm, 2018).

When studying complex systems, there are two systems of coordinates that can be used as a reference when defining the position, velocity, and acceleration of the bodies at all times. The first one is a global coordinate system, and the second is a local coordinate system. A global coordinate system is a reference for all bodies that constitute the system, while a local coordinate system is unique and different for each body. The more bodies are part of a system, the more local coordinate systems exist. Figure 2.2 gives an example of the difference between the two coordinate systems, where the global coordinate axes are defined by X and Y and the local ones by $\bar{X}_A$ and $\bar{Y}_A$.

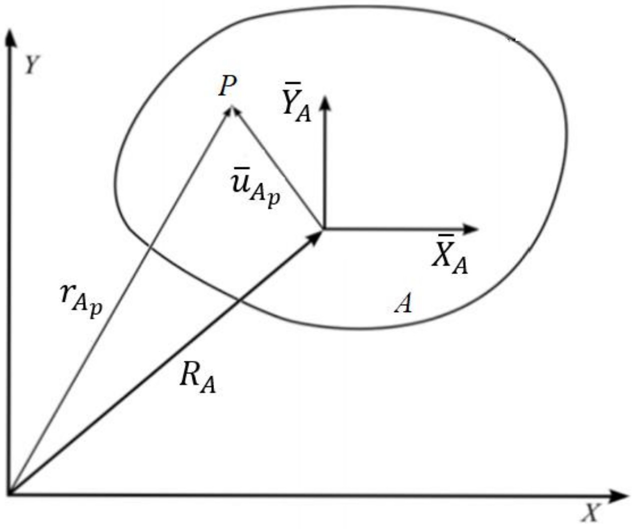


Figure 2.2 Global vs. Local coordinate system, adapted from (Nikravesh, Computer-Aided Analysis of Mechanical Systems, 1988)

The position of a particle P, belonging to the rigid body A, in the global coordinate system, can be expressed in the following way:

$$\mathbf{r}_{A_p} = \mathbf{R}_A + \mathbf{A}_A\bar{\mathbf{u}}_{A_p} \tag{1}$$

where $\mathbf{r}_{A_p}$ is the global position of a particle P, $\mathbf{R}_A$ is the global position of the body where P belongs, $\mathbf{A}_A$ is the transformation matrix, and $\bar{\mathbf{u}}_{A_p}$ is the position of P relatively to the local coordinate system. The vector $\bar{\mathbf{u}}_{A_p}$is considered constant because the body A is considered as a rigid body.

The transformation matrix, $\mathbf{A}_A$, can be expressed as:

$$\mathbf{A}_A \;=\begin{bmatrix} cos\theta_A & -sin\theta_A \\ sin\theta_A & cos\theta_A \end{bmatrix} \tag{2}$$

where θ_A is the angle between the local coordinate system and the global one. The first and second columns of matrix $\mathbf{A}$ represent the axes $\bar{x}$ and $\bar{y}$, respectively.

From the position of a particle, it is possible to obtain its velocity by differentiating the vector $\mathbf{r}_{A_p}$ with respect to time:

$$\dot{\mathbf{r}}_{A_p} = \dot{\mathbf{R}}_A + \dot{\mathbf{A}}_A\bar{\mathbf{u}}_{A_p} \tag{3}$$

where $\dot{\mathbf{R}}_A$ becomes the translational velocity of the body, and $\dot{\mathbf{A}}_A$ is the time derivative of the rotation matrix, expressed as:

$$\dot{\mathbf{A}}_A \;=\begin{bmatrix} -\sin{(\theta_A)}\dot{\theta}_A & -\cos{(\theta_A)}\dot{\theta}_A \\ \cos{(\theta_A)}\dot{\theta}_A & -\sin{(\theta_A)}\dot{\theta}_A \end{bmatrix} = \dot{\theta}_A\mathbf{A}_{\theta,A} \tag{4}$$

$\dot{\theta}_A$ is the angular velocity of the particle. The particle's velocity can then be expressed as follows:

$$\dot{\mathbf{r}}_{A_p} = \dot{\mathbf{R}}_A + \dot{\theta}_A\mathbf{A}_{\theta,A}\bar{\mathbf{u}}_{A_p} \tag{5}$$

From the velocity it is also possible to obtain the acceleration, once again, by differentiating the previous equation with respect to time:

$$\ddot{\mathbf{r}}_{A_p} = \ddot{\mathbf{R}}_A - \dot{\theta}_A^2\mathbf{A}_{\theta,A}\bar{\mathbf{u}}_{A_p} + \ddot{\theta}_A\mathbf{A}_{\theta,A}\bar{\mathbf{u}}_{A_p} \tag{6}$$

where $\ddot{\mathbf{R}}_A$ is the translational acceleration component, $\dot{\theta}_A^2\mathbf{A}_{\theta,A}\bar{\mathbf{u}}_{A_p}$ is the normal acceleration component, and $\ddot{\theta}_A\mathbf{A}_{\theta,A}\bar{\mathbf{u}}_{A_p}$ is the tangential acceleration component of the acceleration.

It is necessary to establish whether the study is to be made using dependent or independent coordinates. According to de Jalón and Bayo (2009), the first dilemma encountered when choosing a system of coordinates which may describe the motion by position, velocity, and acceleration is the problem of either adopting a set of independent coordinates, whose number coincides with the number of degrees of freedom (DOF) and is thereby minimal or adopting an expanded system of dependent coordinates.

The problem can be solved, no matter what type of coordinates is chosen, but the solutions obtained are not similar. Depending on what the problem and the specifications are, using dependent or independent coordinates will have an impact on the efficiency, computational cost, and easiness of the solutions proposed. For example, independent coordinates change all the time because of the movement, and, although the number of equations is minimal, the need to always find the most appropriate set of coordinates can affect the computational effort. Dependent coordinates, on the other hand, although giving a bigger number of equations, do not change for every time instant considered, and the same set of coordinates can be used during all the analysis.

Constraint joints

Normally, in multibody systems, many different joints can be used to model the connection between bodies. The most common joints used are revolute and translational joints.

In planar systems, as the name says, translational joints only allow relative translational movement along one axis between the two bodies, meaning that only one DOF is available. The axis of relative translational movement is called the *line of translation* (Nikravesh, Planar Multibody Dynamics: Formulation, Programming and Applications, 2008). Revolute joints also allow only one DOF, but instead of being the movement along one axis, it is the rotation along one axis.

2.3 Kinematic Analysis

To perform any analysis of a system, the first thing to do is a kinematic analysis, where the focus is on the geometric aspects of the motion, regardless of the forces acting on the system. When performing a kinematic analysis, the input elements or input coordinates should be identified. Input elements are the ones whose position or motion is described, and input coordinates can be an angle or a distance to a reference point (de Jalón & Bayo, 2009).

To perform this analysis, the system's DOF should first be identified, in order to establish and describe the kinematic relationships in terms of the DOF and their time derivatives. Knowing the existing constraints is also a critical part before performing this analysis.

The number of constraint equations is equal to the difference between the number of dependent coordinates and the number of DOF's $(m = n - f)$ (de Jalón & Bayo, 2009). Therefore, for this system's analysis, a formulation in dependent coordinates should be adopted.

For the system's analysis in dependent coordinates, it is necessary to describe how the constraint conditions should be represented:

$$\boldsymbol{\Phi}(\mathbf{q}, t) = 0 \tag{7}$$

Where $\mathbf{q}$ represents the displacement's matrix and t represents the instant considered. When imposing a displacement change to any constraint equation, it follows that:

$$\boldsymbol{\Phi}(\mathbf{q} + \Delta\mathbf{q}, t) = \mathbf{0} \tag{8}$$

and by using the Taylor series development, the residual (displacement change) added to the generalized coordinates can be written as:

$$\boldsymbol{\Phi}(\mathbf{q} + \Delta\mathbf{q}, t) = \boldsymbol{\Phi}(\mathbf{q}, t) + \frac{\boldsymbol{\partial\Phi}(\mathbf{q},t)}{\boldsymbol{\partial}\mathbf{q}}\Delta\mathbf{q} + \frac{1}{2}\frac{\boldsymbol{\partial}^{\mathbf{2}}\boldsymbol{\Phi}(\mathbf{q},t)}{\boldsymbol{\partial}\mathbf{q}^{\mathbf{2}}}\Delta\mathbf{q}^{\mathbf{2}} + \cdots \tag{9}$$

the first order terms from the Tayler series expansion, because it is assumed to be sufficiently accurate, can be used to give a similar expression to eq (9):

$$\boldsymbol{\Phi}(\mathbf{q} + \Delta\mathbf{q}, t) = \boldsymbol{\Phi}(\mathbf{q}, t) + \frac{\boldsymbol{\partial\Phi}(\mathbf{q},t)}{\boldsymbol{\partial}\mathbf{q}}\Delta\mathbf{q} = 0 \tag{10}$$

$\frac{\boldsymbol{\partial\Phi}(\mathbf{q},t)}{\boldsymbol{\partial}\mathbf{q}}$ is denoted as $\mathbf{C_q}$ and is called the Jacobian Matrix. For any given system, the Jacobian matrix is an $m * n$ sized matrix:

$$\mathbf{C_q} = \left[\begin{pmatrix} \frac{\partial C_1}{\partial q_1} & \frac{\partial C_1}{\partial q_2} & \cdots & \frac{\partial C_1}{\partial q_n} \\ \frac{\partial C_2}{\partial q_1} & \frac{\partial C_2}{\partial q_2} & \cdots & \frac{\partial C_2}{\partial q_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial C_m}{\partial q_1} & \frac{\partial C_m}{\partial q_2} & \cdots & \frac{\partial C_m}{\partial n} \end{pmatrix}\right] \tag{11}$$

where C_i stands for any constraint equation and q_i represents the parameter chosen.

The first step of the kinematic analysis is determining the locations and orientations of the different bodies, also known as position analysis. The position can be represented using a $\mathbf{r}$. The second step, after knowing the displacement variables, is a velocity analysis. It is used to determine the velocities of the different bodies that make the system as functions of the time rate and the DOF's. Velocity equations can be achieved by differentiating the position equations in order of time, being represented by $\dot{\mathbf{r}}$. The third and final step is to perform the acceleration analysis. This analysis is also done by differentiating the previous equations in order of time to get the accelerations of the different bodies. Acceleration is represented by $\ddot{\mathbf{r}}$. (Shabana, 2001).

The main difference between kinematic analysis and dynamic analysis is that the first one does not take into consideration the forces applied to the system while the second one does. Next, dynamic analysis is introduced.

2.4 Dynamic Analysis

A truly dynamic problem can be described as having more unknown dependent variables than independent geometric and driving constraint equations. Because of this, it is necessary to establish a dynamic equilibrium condition, taking into consideration the existing loads applied on the system. This dynamic condition leads to a system of second-order differential equations known as the equations of motion (de Jalón & Bayo, 2009).

All descriptions of forces acting on bodies can be associated to Newton (1642-1727), who wrote the three basic laws that guide dynamics:

1. principle of inertia: a particle, where the total sum of the forces applied is equal to zero, tends to maintain its state;

2. principle of force: the acceleration of a particle is proportional to the force applied, with the same direction as the applied force;

3. principle of action/reaction: for every force applied in one of a pair of interacting bodies, there is a reaction force with the same magnitude, opposite direction and collinear happening on the other body of the pair.

The author Sol proposes three steps for performing any dynamic analysis. First, the mass, the inertia tensor, and the body’s momentum need to be discussed. Then follows a description of the external loads and torques imposed on the system. And finally, the equations of motion are given (Sol, 1983). Sol also proposes three ways to get to the equations of motion: The Newton-Euler laws, the virtual work principle of d’Alembert, and the equations of Lagrange (Sol, 1983).

There has been a lot of research about the pros and cons of the different definitions, regarding two major ways of performing the dynamic analysis, one using redundant coordinates and the other, using the system’s DOF’s. The first way leads to a bigger number of equations but gives a more general idea of the formulation and can achieve a sparse matrix structure (where most elements are equal to zero), while the second way leads to a much smaller number of equations, but the complexity increases with the appearance of different coefficients (Shabana, 2001).

Using either the Newton-Euler equations, the method of virtual work or Lagrange’s equations, it is possible to achieve the same result, which is the equations of motion that describe the body’s behaviour.

Principle of virtual work

First, it is useful to understand what the principle of virtual work is. This principle is used together with the principle of virtual displacement, and both are essential to derivate the equations of motion of a system. When applying a virtual displacement, the change of time, *dt*, is not taken into consideration.

When applying a virtual displacement to a body under the action of a force, a new vector appears, called the generalized external forces vector, expressed as Q. The virtual work done by the force applied to the body can be calculated as:

$$\delta \mathbf{W}_{ext,A} = \delta \mathbf{r}_A^T \mathbf{F}_A \tag{12}$$

$$\delta \mathbf{r}_A^T = \left(\frac{\delta \mathbf{r}_A}{\delta \mathbf{q}_A}\right) \delta \mathbf{q}_A \tag{13}$$

where A stands for a given body where the force is applied. The relation $\delta \mathbf{r}_A^T = \left(\frac{\delta \mathbf{r}_A}{\delta \mathbf{q}_A}\right) \delta \mathbf{q}_A$ can be written in a different way:

$$\delta \mathbf{r}_A^T = [\mathbf{I} \quad \mathbf{A}_A \bar{\mathbf{u}}_A] \begin{bmatrix} \delta \mathbf{R}_A \\ \delta \boldsymbol{\theta}_A \end{bmatrix} \tag{14}$$

It is possible to rewrite eq (12) in a simpler way, using the relations given:

$$\delta \mathbf{W}_{ext,A} = \begin{bmatrix} \delta \mathbf{R}_A^T \\ \delta \boldsymbol{\theta}_A \end{bmatrix}^T [\mathbf{I} \quad \mathbf{A}_A \bar{\mathbf{u}}_A]^T \mathbf{F}_A = \begin{bmatrix} \delta \mathbf{R}_A^T \\ \delta \boldsymbol{\theta}_A \end{bmatrix}^T \begin{bmatrix} \mathbf{F}_A \\ \mathbf{A}_A \bar{\mathbf{u}}_A \mathbf{F}_A \end{bmatrix} = \begin{bmatrix} \delta \mathbf{R}_A^T \\ \delta \boldsymbol{\theta}_A \end{bmatrix}^T \mathbf{Q}_{e,A} \tag{15}$$

vector $\mathbf{Q}_{\boldsymbol{e},\boldsymbol{A}}$ is called the generalized forces vector of the force $\mathbf{F}_{\boldsymbol{A}}$. Vector $\mathbf{Q}_{\boldsymbol{e},\boldsymbol{A}}$ includes the translational and rotational components.

Using the same principle, it is also possible to characterize the inertial forces applied to the body. They can be expressed as:

$$\mathbf{F}_{iner,A} = \int_{V_A} \rho \ddot{\mathbf{r}}_A dV_A \tag{16}$$

The virtual work done by the inertial forces can be written as:

$$\delta \mathbf{W}_{iner,A} = \delta \mathbf{r}_A^T \mathbf{F}_{iner,A} = \int_{V_A} \rho \delta \mathbf{r}_A^T \ddot{\mathbf{r}}_A dV_A \tag{17}$$

Using eq (13) to write the virtual displacement, eq (17) can be changed to:

$$\delta \mathbf{W}_{iner,A} = \int_{V_A} \rho \delta \mathbf{q}_A \left(\frac{\delta \mathbf{r}_A}{\delta \mathbf{q}_A}\right) \ddot{\mathbf{r}}_A dV_A = \delta \mathbf{q}_A^T \mathbf{Q}_{iner,A} \tag{18}$$

From eq (18), the virtual displacement of a body can be described as the product of the transposed virtual displacements matrix $\delta\mathbf{q}_A^T$ with the generalized inertial forces vector acting on the body, $\mathbf{Q}_{iner,A}$. The inertial forces (and consequently the inertial work) acting on these types of systems can be despised when compared with the external forces applied.

Principle of virtual power

This principle is similar to the principle of virtual work, with the only difference being the use of a virtual velocity instead of a virtual displacement. It can be used to solve the static equilibrium position problem. On the final equilibrium position, the virtual power of all the forces acting on the multibody should be zero.

First, a virtual velocity vector needs to be defined. It can be defined as a set of imaginary velocities at a stationary time that is consistent with the homogeneous form of the velocity constraint equations. This vector will have the denomination $\dot{\mathbf{q}}^*$. The velocity constraint equations can be driven from the equations of virtual displacements and imposing the virtual variation of the constraint to be zero:

$$\dot{\mathbf{\Phi}} = \mathbf{\Phi}_{\mathbf{q}}^T\dot{\mathbf{q}}^* = [2x \quad 2y]\begin{Bmatrix}\dot{x}^*\\ \dot{y}^*\end{Bmatrix} = 0 \tag{19}$$

where $\dot{\mathbf{\Phi}}$ represents the virtual velocity constraints**,** $\mathbf{\Phi}_{\mathbf{q}}^T$ is the displacement constraints transposed matrix, and $\dot{x}^*$and $\dot{y}^*$ are the virtual velocities according to the X and Y axis, respectively.

Using $\dot{\mathbf{q}}^*$ as a set of n dependent virtual velocities, the principle of virtual power can be presented as:

$$\mathbf{W}^* \ = \sum_{i=1}^{n} \mathbf{F}_i\dot{\mathbf{q}}_i^* = \dot{\mathbf{q}}^{*T}\mathbf{F} = 0 \tag{20}$$

where **F** is the vector of all the forces that produce virtual power, including the inertial ones:

$$\mathbf{F} = \mathbf{M}\ddot{\mathbf{q}} - \mathbf{Q} \tag{21}$$

M is the mass matrix of a system:

$$\mathbf{M}^i = \begin{bmatrix}\mathbf{m}_{RR}^i & \mathbf{m}_{R\theta}^i\\ \mathbf{m}_{R\theta}^i & \mathbf{m}_{\theta\theta}^i\end{bmatrix} \tag{22}$$

The term $\mathbf{m}_{RR}^i$ is the one associated with the body's translational movement and is a constant diagonal matrix:

$$\mathbf{m}_{RR}^i = \begin{bmatrix}m^i & 0 & 0\\ 0 & m^i & 0\\ 0 & 0 & m^i\end{bmatrix} \tag{23}$$

where m^i represents the total mass of the body. The next term, $\mathbf{m}_{R\theta}^i$, represents the inertia coupling between the translation and rotation of the body and can be defined as:

$$\mathbf{m}^{i}_{R\theta} = -\mathbf{A}_{\mathrm{A}} \int \rho^{i} \tilde{\mathbf{u}}^{i} dV^{i}\, \bar{\mathbf{G}}^{i} \tag{24}$$

where $\mathbf{A}_{\mathrm{A}}$ is the transformation matrix, $\tilde{\mathbf{u}}^{i}$ is the distance between the coordinate system and the body's centre of mass and $\bar{\mathbf{G}}^{i}$ is a matrix that depends on the Euler parameters. The Euler parameters are used to describe a finite rotation from the body along an arbitrary axis, its values are usually low (Mevea assumes the rotation to be minimum), the matrix $\mathbf{m}^{i}_{R\theta}$ tends to be a null matrix (full of zeros).

Finally, the term $\mathbf{m}^{i}_{\theta\theta}$ is associated with the rotational coordinates of the body reference, and can be written as:

$$\mathbf{m}^{i}_{\theta\theta} = \bar{\mathbf{G}}^{i\mathrm{T}} \bar{\mathbf{I}}^{i}_{\theta\theta} \bar{\mathbf{G}}^{i} \tag{25}$$

where $\bar{\mathbf{I}}^{i}_{\theta\theta}$ is the inertia tensor of the rigid body, defined as:

$$\bar{\mathbf{I}}^{i}_{\theta\theta} = \begin{bmatrix} i_{11} & i_{12} & i_{13} \\ & i_{22} & i_{23} \\ sym & & i_{33} \end{bmatrix} \tag{26}$$

Note: This mass matrix definition is for a spatial system. Although using, for all the kinetic and dynamic analyses, planar systems, the system considered as a study case is a spatial one, hence the necessity of introducing the mass matrix for spatial systems. For planar ones, the final mass matrix will be an index four square matrix.

Equation (21) can lead to a set of equilibrium equations where the internal constraint forces are not considered, although these equations should be considered in a state of equilibrium. In order to take into consideration these equations, it is necessary to add a set of forces in the direction of the constraint violations ($\mathbf{\Phi}^{T}_{\mathbf{q}}\boldsymbol{\lambda}$) where the columns of $\mathbf{\Phi}^{T}_{\mathbf{q}}$ (rows of $\mathbf{\Phi}_{\mathbf{q}}$) give the direction of the constraint forces and $\boldsymbol{\lambda}$ is the vector of their magnitudes. Because the virtual velocity vector $\dot{\mathbf{q}}^{*}$ belongs to the null space of $\mathbf{\Phi}_{\mathbf{q}}$, the product ($\dot{\mathbf{q}}^{*}\mathbf{\Phi}^{T}_{\mathbf{q}}\boldsymbol{\lambda}$) is equal to zero and therefore can be added to the equation (20), giving the complete set of force equilibrium equations:

$$\mathbf{W}^{*} = \dot{\mathbf{q}}^{*T}\left(\mathbf{M}\ddot{\mathbf{q}} - \mathbf{Q} + \mathbf{\Phi}^{T}_{\mathbf{q}}\boldsymbol{\lambda}\right) = 0 \tag{27}$$

And consequently:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}^{T}_{\mathbf{q}}\boldsymbol{\lambda} = \mathbf{Q} \tag{28}$$

Therefore, the virtual power of the forces acting on a multibody system can be written as:

$$\dot{\mathbf{q}}^{*T}(\mathbf{M}\ddot{\mathbf{q}} - \mathbf{Q}) = 0 \tag{29}$$

$\dot{\mathbf{q}}^{*}$ must satisfy the first derivative of the constraint equations at a stationary time, (19).

It is possible to find the m values of the vector $\boldsymbol{\lambda}$. This vector establishes the magnitude of the constraint forces, so that:

$$\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}^{T}\boldsymbol{\lambda} = \mathbf{Q} \tag{30}$$

where now the vector $\mathbf{Q}$ contains the external forces plus all the velocity dependent inertial terms.

Note: The expressions and definitions for the principles of virtual work and virtual force are adapted from (de Jalón & Bayo, 2009).

Method of the Lagrange multipliers

Equation (30) represents a set of n equations with a total of $(n+m)$ unknown variables: n variables belonging to vector $\ddot{\mathbf{q}}$ and m variables, belonging to the vector $\boldsymbol{\lambda}$. To solve the problem, it is necessary to have the same number of equations and number of unknowns, so there is a need to supply m more equations. To do this, acceleration kinematic equations can be added. These are obtained by differentiating the constraint equations (7) twice with respect to time:

$$\boldsymbol{\Phi}_{\mathbf{q}}\ddot{\mathbf{q}} = -\dot{\boldsymbol{\Phi}}_{t} - \dot{\boldsymbol{\Phi}}_{\mathbf{q}}\dot{\mathbf{q}} = \mathbf{c} \tag{31}$$

The vector $\mathbf{c}$ appears as a form of simplification of the previous equation. Writing equations (30) and (31) together, makes it possible to introduce the following matrix system:

$$\begin{bmatrix} \mathbf{M} & \boldsymbol{\Phi}_{\mathbf{q}}^{T} \\ \boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{Q} \\ \mathbf{c} \end{Bmatrix} \tag{32}$$

which is a system with $(n+m)$ equations with $(n+m)$ unknowns. This system can be used to solve both accelerations and Lagrange multipliers (de Jalón & Bayo, 2009).

Semi recursive formulation

This type of formulation allows the dynamic analysis of a body to be done more efficiently, being used in computational analysis. It requires the system to have a tree structure (see figure 2.3). The main advantages of using recursive definitions are an easier derivation of the equations of motion, easier software coding and debugging, and reduced computational effort.

Using the semi recursive formulation, the equations of motions can be written as follows:

$$\ddot{\mathbf{q}} = f(\mathbf{q}, \dot{\mathbf{q}}, t) \tag{33}$$

where $\ddot{\mathbf{q}}$ has a reduced size because it only contains relative coordinates of one body respectively to an adjacent body of the system. For this type of definitions, only reduced matrices need to be inverted, which requires less computational effort than having to invert large matrices (the integration itself does not take too much computational effort but the inversion of large-sized matrices does). As a result, only small equations of similar type as equation (20) need to be integrated.

Semi recursive formulation uses the relationships between an arbitrary pair of contiguous bodies belonging to the same system. The pairs interact through a joint. Considering that all bodies are connected to each other by any type of joints, a tree structure can be considered, so that the formulation can be applied.

A tree structure is also considered an open loop chain system of bodies. Figure 2.3 exemplifies a tree structure of a multibody system. The body B^1 is considered the base of the system and so on. The bigger the number of bodies between bodyB^i and body B^1, the bigger the number assigned to the body B^i (Slaats, 1991).

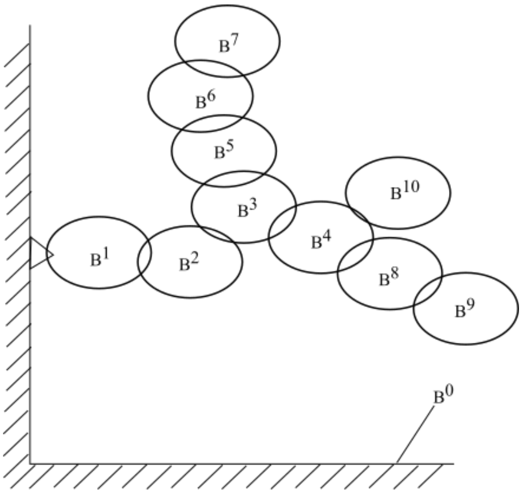


Figure 2.3 Tree structure of a multibody system, adapted from (Slaats, 1991)

Semi recursive formulation starts by describing the kinematics associated with each arbitrary pair. Figure 2.4 exemplifies an arbitrary pair of bodies, B_{j-1} and B_j, connected by an arbitrary joint. X, Y, and Z represent the global coordinate reference axis, while $\bar{x}_{B_{j-1}}$, $\bar{y}_{B_{j-1}}$, and $\bar{z}_{B_{j-1}}$ represent the local coordinate reference axis at B_{j-1}, located in its centre of mass. Points P and Q represent the location of the joint in both bodies. Vector $\mathbf{d_{j-1,j}}$ shows the relative displacement between points P and Q (Jaiswal;Islam;Hannola;Sopanen;& Mikkola, 2018).

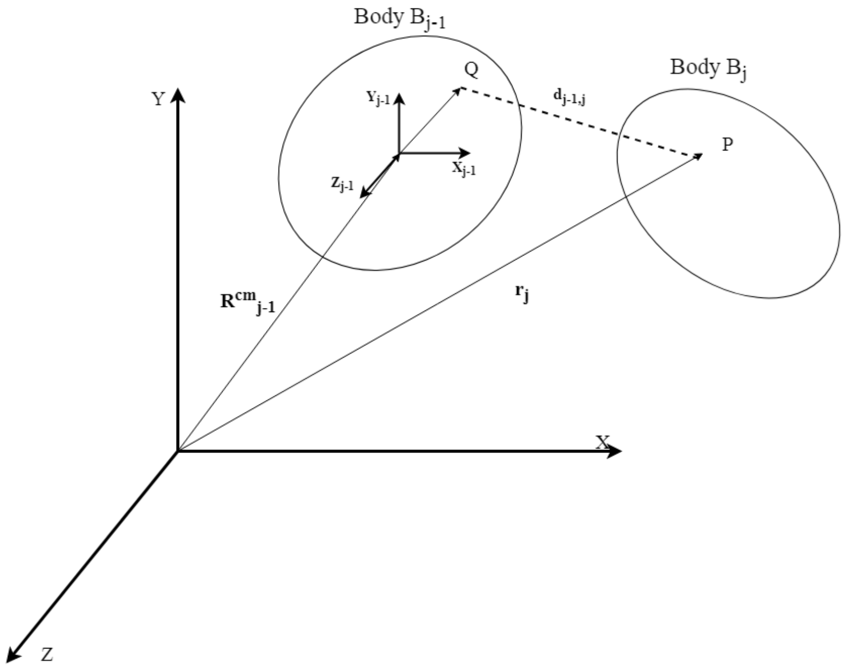


Figure 2.4 Arbitrary pair of bodies B*j-1* and B*j*, adapted from (Jaiswal;Islam;Hannola;Sopanen;& Mikkola, 2018)

The global position vector of body $\mathbf{B}_j$, $\mathbf{r}_j$, global velocity vector, $\dot{\mathbf{r}}_j$, and global acceleration vector, $\ddot{\mathbf{r}}_j$ can be written as:

$$\mathbf{r}_j = \mathbf{R}_{j-1} + \mathbf{A}_{j-1}\bar{\mathbf{u}}_{j-1} + \mathbf{d}_{j-1,j} \tag{34}$$

$$\dot{\mathbf{r}}_j = \dot{\mathbf{R}}_{j-1} + \dot{\theta}_{j-1}\mathbf{A}_{j-1}\bar{\mathbf{u}}_{j-1} + \dot{\mathbf{d}}_{j-1,j} \tag{35}$$

$$\ddot{\mathbf{r}}_j = \ddot{\mathbf{R}}_{j-1} - \dot{\theta}^2_{j-1}\mathbf{A}_{j-1}\bar{\mathbf{u}}_{j-1} + \ddot{\theta}_{j-1}\mathbf{A}_{j-1}\bar{\mathbf{u}}_{j-1} + \ddot{\mathbf{d}}_{j-1,j} \tag{36}$$

The angular velocity of the body B_j, $\boldsymbol{\omega}_j$, and its angular acceleration, $\dot{\boldsymbol{\omega}}_j$, can be written is terms of the global frame of reference, which means, in terms of the global angular velocity, $\boldsymbol{\omega}_{j-1}$, and the global angular acceleration, $\dot{\boldsymbol{\omega}}_{j-1}$, respectively:

$$\boldsymbol{\omega}_j = \boldsymbol{\omega}_{j-1} + \boldsymbol{\omega}_{j-1,j} \tag{37}$$

$$\dot{\boldsymbol{\omega}}_j = \dot{\boldsymbol{\omega}}_{j-1} + \dot{\boldsymbol{\omega}}_{j-1,j} \tag{38}$$

with the index $(j-1, j)$ meaning the velocity and acceleration of the body B_j with respect to the body B_{j-1}.

Using the principle of virtual work in matrix form and the kinematic analysis presented above, the equations of motion for the body B_j can be written:

$$\{\dot{\mathbf{q}}_j^{*T} \quad \boldsymbol{\omega}_j^{*T}\}\left(\begin{bmatrix}\mathbf{M}_j\mathbf{I}_3 & 0 \\ 0 & \mathbf{A}_j\mathbf{J}_0(\mathbf{A}_j^T)\end{bmatrix}\begin{bmatrix}\ddot{\mathbf{R}}_j \\ \dot{\boldsymbol{\omega}}_j\end{bmatrix} + \begin{bmatrix}0 \\ \tilde{\boldsymbol{\omega}}_j\mathbf{A}_j\mathbf{J}_0(\mathbf{A}_j^T)\boldsymbol{\omega}_j\end{bmatrix} - \begin{bmatrix}\mathbf{F}_j \\ \mathbf{T}_j\end{bmatrix}\right) = 0 \tag{39}$$

Where $\dot{\mathbf{q}}_j^*$ and $\boldsymbol{\omega}_j^*$ are, respectively, the virtual translational velocity and the virtual angular velocity, $\mathbf{M}_j$ is the mass matrix of the body B_j, $\mathbf{I}_3$ is the index three identity matrix, $\mathbf{J}_\mathbf{0}$ is the rotation inertia tensor, $\mathbf{A}_j$ is the rotation matrix, $\tilde{\boldsymbol{\omega}}_j$ is the skew-symmetric matrix of $\boldsymbol{\omega}_j$, and $\mathbf{F}_j$ and $\mathbf{T}_j$ are forces and torques applied to the body B_j, respectively.

For the entire system, the equations of motions can be written like equation (28):

$$\dot{\mathbf{q}}^{*\mathrm{T}}\left(\mathbf{M}\ddot{\mathbf{q}} - \mathbf{Q} + \boldsymbol{\Phi}_{\mathbf{q}}^{T}\boldsymbol{\lambda}\right) = 0 <=> \dot{\mathbf{q}}^{*T}(\mathbf{M}\ddot{\mathbf{q}} - \mathbf{Q} + \mathbf{C}) = 0 \tag{40}$$

$\boldsymbol{\Phi}_{\mathbf{q}}^{T}\boldsymbol{\lambda}$ is the quadratic velocity and it can be represented by the letter value $\mathbf{C}$. This system of equations can shorten by using relative joint coordinates. Global velocities and relative joint velocities can relate by using the velocity transformations matrix, **R**:

$$\dot{\mathbf{q}} = \mathbf{R}\dot{\mathbf{z}} \tag{41}$$

where $\dot{\mathbf{z}}$ is the relative joint velocities vector. The accelerations, $\ddot{\mathbf{q}}$ can be obtained by differentiating the previous system in order of time:

$$\ddot{\mathbf{q}} = \mathbf{R}\ddot{\mathbf{z}} + \dot{\mathbf{R}}\dot{\mathbf{z}} \tag{42}$$

where $\ddot{\mathbf{z}}$ is the relative joint acceleration vector and $\dot{\mathbf{R}}$ is the time derivative of **R**.

By substituting the equations (41) and (42) into equation (40), the equations of motion can be expressed as:

$$\dot{\mathbf{z}}^{*T}(\mathbf{R}^T\mathbf{M}\mathbf{R}\ddot{\mathbf{z}} + \mathbf{R}^T\mathbf{M}\dot{\mathbf{R}}\dot{\mathbf{z}} + \mathbf{R}^T\mathbf{C} - \mathbf{R}^T\mathbf{Q}) = 0 \tag{43}$$

Because this equation is valid for any arbitrary vector of independent velocities, $\dot{\mathbf{z}}^*$ can be eliminated from the expression and the equations of motion can be written in a simpler way:

$$\mathbf{R}^{\mathbf{T}}\mathbf{M}\mathbf{R}\ddot{\mathbf{z}} = \mathbf{R}^{\mathrm{T}}(\mathbf{Q} - \mathbf{C}) - \mathbf{R}^{\mathrm{T}}\mathbf{M}\dot{\mathbf{R}}\dot{\mathbf{z}} \tag{44}$$

Note: The explanation presented in this subchapter is adapted from (Jaiswal;Islam;Hannola;Sopanen;& Mikkola, 2018)

2.5 *Numerical Integration*

Dynamic analysis of constrained multibody systems leads to a set of differential algebraic equations (DAE). Because those equations are very difficult to solve analytically, researchers have tried to seek numerical methods to approximate the solutions at discrete times (t_1, t_2, t_3, …). The amount of time passed between discrete times is considered constant (for the method used) and is defined as a time step Δt .To approximate the solution at different discrete times, these equations need to be transformed into second order differential equations (ODE), and after that into first order ODE’s, so that the numerical integration can be fast and accurate.

The numerical integration can be made using various computational tools and it is becoming an important part of computational dynamics due to the easiness to implement as well as faster implementation time.

Second order differentials like the acceleration equation can be expressed as:

$$\ddot{\mathbf{q}} = \mathbf{F}(t, \mathbf{q}, \dot{\mathbf{q}}) \tag{45}$$

This equation's solutions also need to satisfy the initial conditions $\mathbf{q}(t_0) = \mathbf{q_0}$ and $\dot{\mathbf{q}}(t_0) = \dot{\mathbf{q}}_\mathbf{0}$. Most computer software tools try to solve this type of equations by turning them into first order ODE's and then solving them. The transformation can be done by substituting the variable $\dot{\mathbf{q}}$ for a first order one, $\mathbf{s}$, ($\dot{\mathbf{q}} = \mathbf{s}$). That way, a new ODE appears instead of equation (45):

$$\dot{\mathbf{s}} = \mathbf{f}(t, \mathbf{q}, \mathbf{s}) \tag{46}$$

Introducing $\mathbf{y}^\mathbf{T} \equiv \{\mathbf{q}^\mathrm{T}, \mathbf{s}^\mathrm{T}\}$ as a new vector variable that includes $\mathbf{q}$ and $\mathbf{s}$, equation (46) can be rewritten in a simpler way:

$$\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y}) \tag{47}$$

Given t and $\mathbf{y}$, the value of $\dot{\mathbf{y}}$ is calculated and consequently, the accelerations end up being calculated. This process is called *function evaluation* (de Jalón & Bayo, 2009).

Figure 2.5 resumes the process of numerical integration in dynamic analysis. It is important to notice that numerical integration process occurs until the end of the analysis time. The instant where the numerical integration occurs is represented by the letter t. When the numerical integration happens, the function evaluation checks if the instant is higher or equal than the total time. If it is not higher or equal, the next instant is considered by adding an infinitesimal amount of time, dt, to the previous time instant.

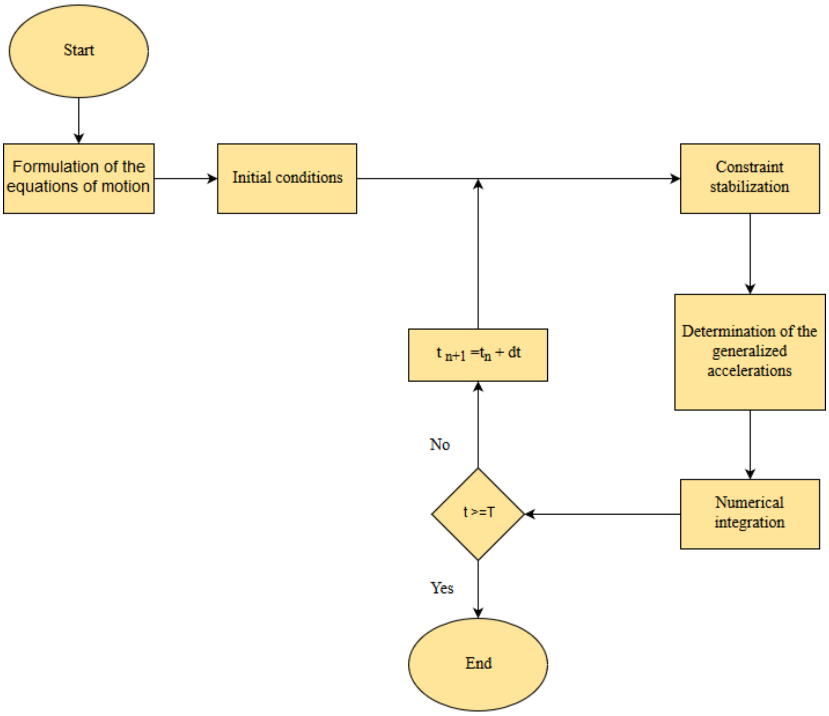


Figure 2.5 Numerical integration in dynamic analysis

Runge-Kutta Method

This method works as a numerical integration method to transform the DAE's into ODE's and solve them. It was created due to the low accuracy levels of Euler's method and the difficulty of obtaining the higher order derivatives $\mathbf{f}(t,\mathbf{y})$ in Taylor's series. Although Euler's method easiness to implement, it requires the time steps used to be very small, making the round-up errors increase and consequently making this method useless (de Jalón & Bayo, 2009).

Runge-Kutta method works by doing function evaluations instead of computing higher order derivatives. By knowing information about t and y it computes the value of $\dot{y}$ for each time step. This reduces the round up errors (Flores, 2015).

Starting with the Euler's method, the equation that provides the solution for each y_i is presented as:

$$y_{i+1} = y_i + \Delta t f(t_i, y_i) + O(h^2) \tag{48}$$

where $O(h^2)$ represents the local truncation error. The method's order and accuracy can be specified by its local truncation order's error. For a truncation error with order $O(h^{p+1})$**,** the method's order will be p. It is easy to understand that the Euler's method is a first order method. The higher the order of the method, the more accurate it becomes.

To increase the order of the Euler's method, a new method was developed, called the *Runge-Kutta second order method* which can also be called the improved Euler's method:

$$y_{i+1} = y_i + \frac{\Delta t}{2}(k_1 + k_2) + O(h^3) \tag{49}$$

$$k_1 = f(t_i, y_i) \tag{50}$$

$$k_2 = f(t_i + \Delta t, y_i + \Delta t k_1) \tag{51}$$

This method requires two function evaluations for each time step, meaning two solutions of the equations of motion to obtain the accelerations at any given time step. It is important to notice that k_1 does not depend on k_2. Figures 2.6 and 2.7 serve as an example for the Euler's method and the Runge-Kutta second order method. From the first look it is possible to understand the difference between both in terms of accuracy. Runge-Kutta methods use more function evaluations for each time step, reducing the round-up errors. For the second order method (figure 2.7), between points t_i and t_{i+1}, the midpoint derivative is used to increase accuracy. In both figures, the time instants are represented by the letter X.

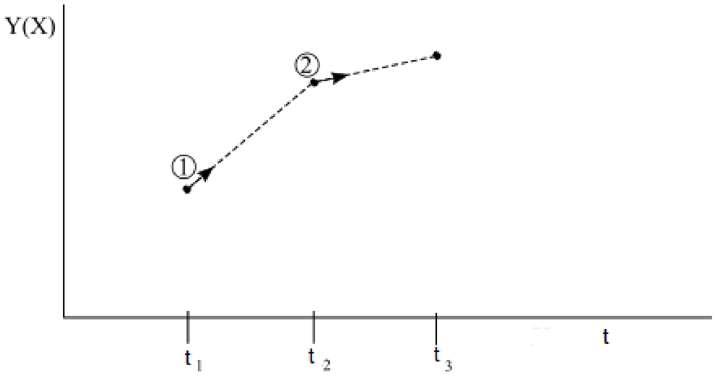


Figure 2.6 Euler's method, adapted from (Press;Teukolsky;Vetterling;& Flannery, 1992)

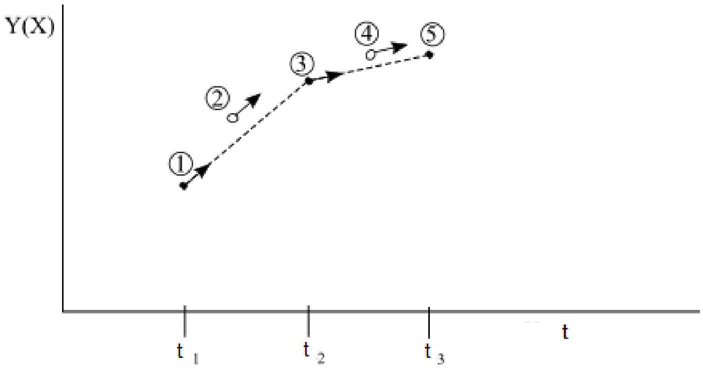


Figure 2.7 Rung-Kutta second order method, adapted from (Press;Teukolsky;Vetterling;& Flannery, 1992)

One of the most used methods used in multibody dynamics analysis is the *Runge-Kutta fourth order method.* As the name says, it is a fourth order method, meaning that it should be more accurate than the previous two methods presented. It allows a bigger timestep to be used (and for a timestep twice as big as the one used in the second order method the accuracy is the same) with the biggest difference being the requirement of four function evaluations for each time step. This method can be expressed as:

$$y_{i+1} = y_i + \frac{\Delta t}{6}(k_1 + k_2 + k_3 + k_4) + O(h^5) \tag{52}$$

$$k_1 = f(t_i, y_i) \tag{53}$$

$$k_2 = f\left(t_i + \frac{\Delta t}{2}, y_i + \frac{\Delta t}{2}k_1\right) \tag{54}$$

$$k_3 = f\left(t_i + \frac{\Delta t}{2}, y_i + \frac{\Delta t}{2}k_2\right) \tag{55}$$

$$k_4 = f(t_i + \Delta t, y_i + \Delta t k_3) \tag{56}$$

Now the values of the different k_{i+1} depend on the previous values. This method does not provide an estimate of the local error, so it is difficult to know if the time step being used is the correct one. Because of the reduced computational cost, it does not a problem to try different time steps. Figure 2.8 shows the four function evaluations between time steps, ordered by numbers. The first evaluation occurs at the initial point of the time step, while to next two ones occur in the midpoints. The final one should occur in the final point of the time step. These evaluations can be considered as tangents to the actual function.

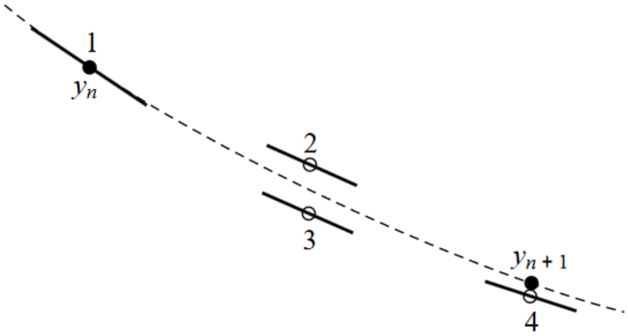


Figure 2.8 Fourth-order Runge-Kutta method, from (Press;Teukolsky;Vetterling;& Flannery, 1992)

Note: the information and equations presented in this sub-chapter are adapted from (Press;Teukolsky;Vetterling;& Flannery, 1992), (de Jalón & Bayo, 2009), and (Flores, 2015).

Mevea software uses a semi-recursive formulation and a Runge-Kutta fourth order method to perform the system’s dynamic analysis.

2.6 *Collision and contact modelling*

When performing a simulation using complex systems, collisions should be taken into consideration. The model and the simulation need to be realistic, prohibiting the interpenetration of bodies. It is necessary to determine when contact happens, how many bodies are colliding and what the collision response should be. To understand this, two major steps are considered: contact detection and collision response. Contact detection is considered the most important step. It relies on the geometry of the bodies to be accurate. The collision geometry needs to be simple, so that the software can return the possibility of when and where the collision is going to happen. Figure 2.9 shows the difference in a body's model geometry and the one used for contact detection. It is possible to notice the differences between the model and the collision model, where the geometry has rectangular shapes. Contact response is what keeps the bodies from entering each other. It analyses the bodies and their properties, like contact forces, geometric properties and relative velocities. (Baharudin;Rouvinen;Korkealaasko;& Mikkola, 2014).

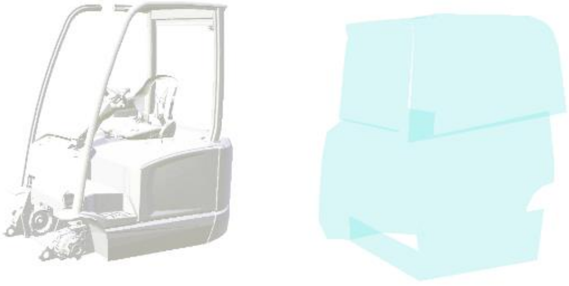


Figure 2.9 Simulation model vs. collision model

One of the existing theories for analyzing collisions in real time simulation is the *Penalty method*. It is also known as the soft contact method because it allows small penetrations between bodies and adds a temporal spring damper. Figure 2.10 resumes how this method works for collision detection. Vectors $\mathbf{d}$ and $\dot{\mathbf{d}}$ are the distance and velocity between the two bodies, while $\mathbf{F}_N$ and $\mathbf{F}_T$ are the normal and tangent forces. When $\mathbf{d}$ and $\dot{\mathbf{d}}$ are equal or less than zero the collision occurs and the collision response method starts to work (Baharudin;Rouvinen;Korkealaasko;& Mikkola, 2014).

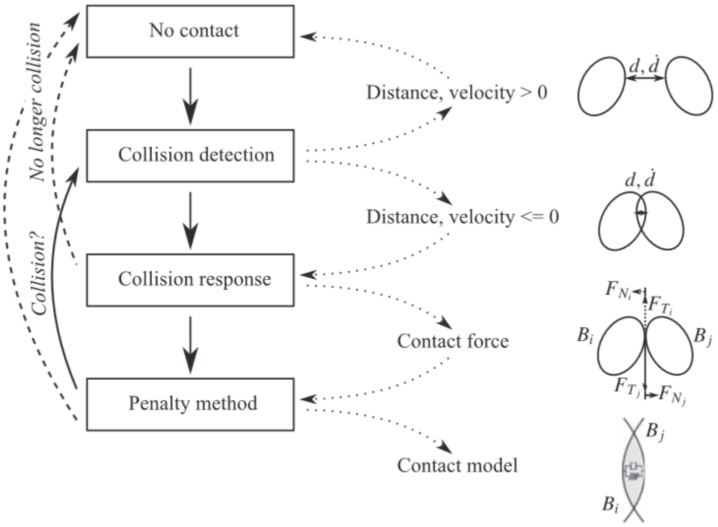


Figure 2.10 Collision detection between two bodies, from (Baharudin;Rouvinen;Korkealaasko;& Mikkola, 2014)

The collision can be described using a global frame of reference and two bodies like it is showed in figure 2.11. The vectors $\mathbf{r}_i$ and $\mathbf{r}_j$ represent the location of the contact point of both bodies with respect to the global frame. The distance between the two bodies is $\mathbf{d}_p$.

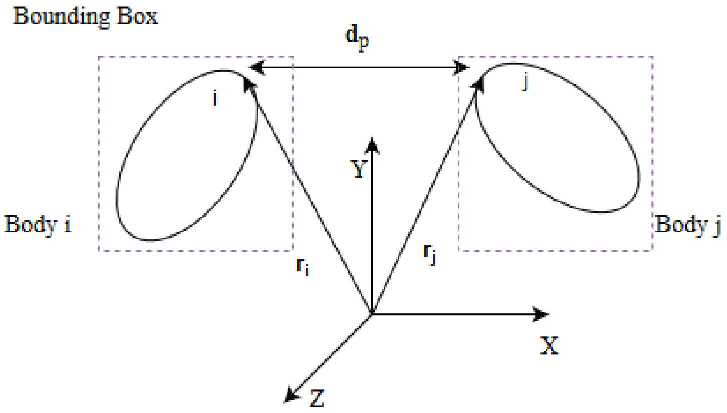


Figure 2.11 Contact between two bodies, adapted from (Baharudin M. , 2016)

$\mathbf{d}_p$ can be calculated as:

$$\mathbf{d}_p = \mathbf{r}_j - \mathbf{r}_i \tag{57}$$

It is possible to calculate the normal vector at the contact point, **n**:

$$\mathbf{n} = \frac{\mathbf{d}_p}{\|\mathbf{d}_p\|} \tag{58}$$

and with the normal vector, the location of the collision, **d,** can be known:

$$\mathbf{d} = \mathbf{n}^{\mathrm{T}}\mathbf{d}_p \tag{59}$$

The relative velocity of the collision can also be calculated:

$$\mathbf{v}_n = \mathbf{n}^{\mathrm{T}}(\dot{\mathbf{r}}_j - \dot{\mathbf{r}}_i) \tag{60}$$

In order to define the contact forces, a spring and a damper are added. That collision force can be expressed as:

$$\mathbf{F} = -K\mathbf{x} - S(\mathbf{n}\mathbf{v}_n) \tag{61}$$

where $K\mathbf{x}$ represent the spring's force and S is the damping coefficient. It is important to choose the coefficients' values right to obtain valid collision results.

Note: The information and expressions here presented are adapted from (Baharudin;Rouvinen;Korkealaasko;& Mikkola, 2014).

2.7 Hydraulic circuit

When using a system with a hydraulic circuit in it, it becomes necessary to perform also an analysis of this sub-system. When modelling and analyzing hydraulic circuits, there are two important properties to take into consideration: viscosity and bulk modulus.

Viscosity can be defined as a measure of the fluid's resistance to deformation by shear or tensile stress, while the bulk modulus can be defined as a measure of the fluid's resistance to compression, or in the other words, it can be defined as the infinitesimal pressure increase related to the infinitesimal volume decrease. The bulk modulus, B, can be expressed by the following expression and has a typical value of 1500 Mpa, for oil:

$$B = -\frac{dp}{\frac{dV}{V}} = -\frac{Vdp}{dV} \tag{62}$$

Figure 2.12 shows a unit sized hydraulic volume, used as an infinitely stiff container to exemplify the bulk modulus. The volume is compressed by a force, causing a $-dV$ change. That change causes the pression to increase dp.

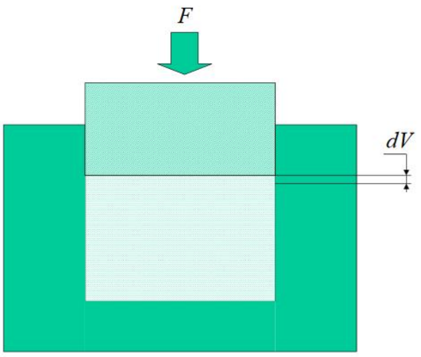


Figure 2.12 Unit sized hydraulic volume, from (Mikkola, Week 9-18moo, 2018)

When using multiple volumes, the system has multiple bulk modulus. This combined effect of the different bulk modulus is called *effective bulk modulus,* B_e (Mikkola, Week 9-18moo, 2018).

When applying a force to a certain hydraulic volume, the oil compresses while the container expands. The starting volume can be expressed as:

$$V_t = V_1 = V_{c1} \tag{63}$$

where V_1 is the initial fluid volume, and V_{c1} is the initial volume of the container. The means it does not matter if the volume chosen is the volume or the container one. When imposing volume changes, dV_t, it can be written:

$$dV_t = -dV_1 + dV_{c1} \tag{64}$$

The oil used when modelling hydraulic circuits is compressible and it behaves like a spring. Different oils have different properties.

For mobile machines, it is possible to use the lump fluid theory, explained in this chapter.

Flow types

The flow type is a very important factor to take into consideration when working with hydraulic circuits. The flow can be divided into two types: laminar and turbulent.

Laminar flow occurs when the fluid moves smoothly between layers. There are no movements in another direction different from the direction of the movement. On the other hand, turbulent flow occurs when the fluid's particles have no specific trajectory and have different velocities Figure 2.13 gives an example of the two types of flow.

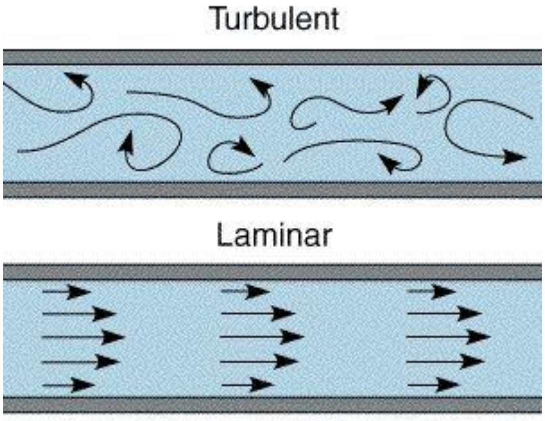


Figure 2.13 Laminar flow and turbulent flow, from (Vagabond, 2012)

The type of flow can be accessed by calculating the *Reynolds number*, Re:

$$Re = \frac{vD_h}{v} \tag{65}$$

where v is the fluid's viscosity, D_h is the hydraulic diameter, and v is the fluid's velocity. Normally, in pipes or hoses, laminar flow occurs for $Re < 2300$ while turbulent flow occurs for $Re > 4000$.

Lumped fluid theory

This theory can be applied to hydraulic systems where the air pressure is considered insignificant. In mobile machines, the hoses are normally short, and the work cycles are relatively slow, making pressure waves insignificant.

To use this theory, it is necessary to divide the circuit into volumes, where the pressure is assumed to be constant in each one. The volumes are considered separated by throttle valves through which the fluid can pass. Each volume has its differential equations and when all are solved, it is possible to obtain the system's pressure. Figure 2.14 gives an example of a simple hydraulic system with two volumes. Each one has a volume, V_i, a pressure, p_i and a bulk modulus, B_i. Q represents the flow passing from one volume to the other (Mikkola, Week 9-18moo, 2018).

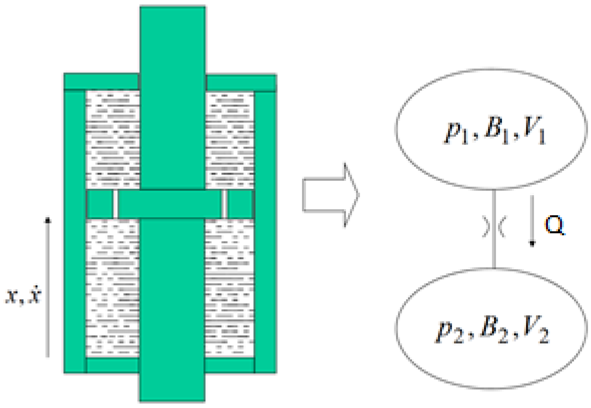


Figure 2.14 Hydraulic system with two volumes, from (Mikkola, Week 9-18moo, 2018)

To calculate the different pressure in the different volumes, it is possible to use the flow continuity equation. An example of how this equation works is presented in the figure shown below. The value ρ_i represents the density of the oil and Q_i is the flow rate.

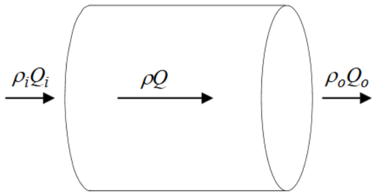


Figure 2.15 Hydraulic volume, from (Mikkola, Week 9-18moo, 2018)

The mass of the volume can be expressed as:

$$M = \rho V \tag{66}$$

If the flows going in and out are not equal, the initial and final masses are going to be different:

$$\frac{d}{dt}M = \dot{m}_i - \dot{m}_o \tag{67}$$

where $\dot{m}_i$ and $\dot{m}_o$ are the mass flow rates going in and out, respectively. Using the equation of mass flow:

$$\dot{m}_i = \rho_i Q_i \tag{68}$$

equation (67) takes the form:

$$\frac{d}{dt}(\rho V) = \rho_i Q_i - \rho_o Q_o <=> \frac{d\rho}{dt}V + \frac{dV}{dt}\rho = \rho_i Q_i - \rho_o Q_o \tag{69}$$

When the going in fluid's density is equal to the one going out, equation (69) can be rewritten:

$$\frac{d\rho}{dt}\frac{V}{\rho} + \frac{dV}{dt} = Q_i - Q_o \tag{70}$$

Using the relation $\frac{dV}{V} = -\frac{d\rho}{\rho}$ in the equation of the bulk modulus (62) leads to:

$$d\rho = \frac{dp\rho}{B_e} \tag{71}$$

If equation (71) is introduced into the equation of flow continuity (70) the resulting expression is:

$$\frac{dp}{dt} = \frac{B_e}{V}\left(Q_i - Q_o - \frac{dV}{dt}\right) \tag{72}$$

The final equation is the first order differential equation for the unknown pressure. It is called *the flow continuum equation* (Mikkola, Week 9-18moo, 2018).

For n hydraulic volumes, the total differential pressure can be calculated by:

$$\frac{dp}{dt} = \frac{B_{e_i}}{V_i} \sum_{i=1}^{n} [(Q_i - Q_o) - \frac{dV_i}{dt}] \tag{73}$$

Hydraulic Components

Hydraulic circuits normally need to be modelled using valves, hoses, cylinders, and pumps, according to the machine's needs.

Valves are used to control the pressure, flow rate, and direction of the fluid. They are controlled using spools. A spool can block the fluid's way, enabling or disabling movement as needed. To model a spool's behaviour, an equation of motion is necessary, taking into consideration the pressure, friction, flow spring and (when applicable) electric forces (Mikkola, Week 10-2018mool, 2018).

Some simpler valves, like throttle valves, do not work using spools. In that case, the only equations needed to describe the functioning are the ones describing the volume flow.

Because equations of motion for valves are complicated to get and can have a low accuracy, three approaches were proposed for modelling of valves: empirical, analytical, and semi-empirical approaches.

On one hand, the empirical approach is an experimental method, using information from different measurements, and not relying on any theoretical information. On the other hand, the analytical approach is based on the equations that describe the different parts in the valve, not relying on any tests. The semi-empirical approach uses the previous two approaches, being based on the laws of physics, but also using measurements to describe the system's parameters (Mikkola, Week 10-2018mool, 2018).

Using the semi-empirical approach, it is necessary to calculate the flow rate passing through a valve:

$$Q_v = C_v X_0 \sqrt{d_p} \tag{74}$$

where Q_v is the flow rate, C_v is the flow rate constant and X_0 is the relative spool position, in meters. X_0 can be calculated:

$$\frac{dX_0}{d_t} = \frac{U_{ref} - X_0}{\tau} \tag{75}$$

where U_{ref} is the reference voltage signal for the reference spool position, given in meters, and τ is the time constant and it can be obtained from the Bode-diagram supplied by the valve's manufacturer.

The hydraulic cylinders are used to convert the hydraulic pressure into mechanical power. Figure 2.16 shows a hydraulic cylinder used as an example.

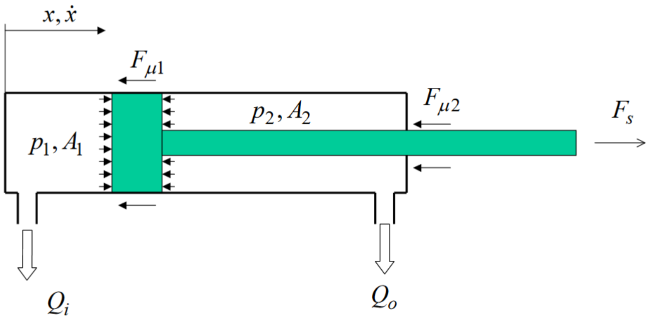


Figure 2.16 Hydraulic cylinder, from (Mikkola, Week11-2018mood, 2018)

The force produced by the cylinder, F_s, is calculated by:

$$F_s = A_1 p_1 - A_2 p_2 - \sum F_\mu \tag{76}$$

where A_1 and $A_2$are the areas in the cylinder side and piston side, respectively, p_1 and p_2 are the hydraulics pressures from the cylinder and the piston sides. The vector F_μ represents the friction forces.

Note: The information and expressions presented are adapted from (Mikkola, Week 10-2018mool, 2018) and (Jaiswal;Islam;Hannola;Sopanen;& Mikkola, 2018).

2.8 Computer-aided simulation

The Webster dictionary defines a model as a simplified representation of something and therefore it is possible to create models for almost everything. Modelling refers to the activity of using a model as an abstract representation of a real-world entity. (Crespo Marquez, 2010) This approach can be considered as a first step to understand what computational aided simulation is.

Computation solutions as CAD or CAM have appeared as tools to help engineers face complex problems and assess complex solutions (Smith, 2012) that would take considerably more effort if done by traditional methods due to round up errors and would need results confirmation. These aspects make computer-aided processes a more efficient way to work, reducing times and costs when working.

CAS systems can be used for describing a single body or for describing the final product as an assembly of those parts. To achieve a successful simulation, machines need such capabilities as computer-aided analysis, design sensitivity or optimization. (Nikravesh, Computer-Aided Analysis of Mechanical Systems, 1988).

Many software tools have been created to perform numerical analysis and are able to do it using different definitions. This is possible because these definitions use systematic techniques, where the user only has to insert the parameters to be evaluated.

According to Haug (Haug, 1993) and Larson & Papalambros (Larson & Papalambros, 2003), multibody dynamic simulation (MBS) analysis, should follow these stages:

- Modelling the multibody system;
- Deriving the equations of motion:
- Simulating the trajectories of the generalized coordinates
- Animating the bodies
- Evaluating the dynamical performance by adequate criteria

Real-time simulation

Simulations that perform the analysis in real time need a special set of requirements. They need to run at the same speed as a real machine would do and also need to accommodate the different input signals and calculate the response in minimum time steps (Cellier & Kofman, 2006).

In real-time simulations, the accuracy of the analysis depends not only on the representation of the system and the formulation chosen but also on the amount of time used to produce results. The amount of time needed must be the same as the physical counterpart would need. Figure 2.17 illustrates how the functions, f(t), used should behave regarding the time of the simulation, sim. clock. The time steps should be accurate when compared to reality. (Bélanger;Venne;& Paquin, 2010). From figure 2.17, it becomes easier to understand the importance of the time steps for the analysis. If the function needs an amount of time to achieve a solution bigger than the time step chosen, the simulation stops being real-time, because it loses its capability to solve the functions and update the behavior in the correct amount of time.

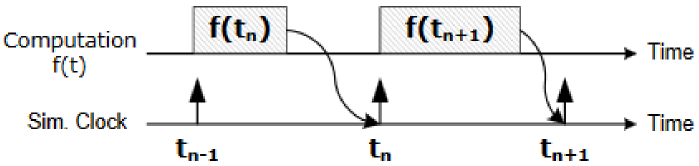


Figure 2.17 Chronological principle of real-time simulation, from (Bélanger;Venne;& Paquin, 2010)

Using the example given by Cellier and Kofman (Cellier & Kofman, 2006), where they say that a flight simulator used for training purposes is useless if it cannot produce a response to an input in real time, because the trainee makes mistakes when learning how to control the system, and those mistakes can be very costly if they were to happen in the real machine, it is easy to understand the importance of the simulation's accuracy.

To implement a real-time simulation, Cellier and Kofman (Cellier & Kofman, 2006) enumerate four steps:

1 – *The real-time clock*, which is responsible for the synchronization of the real-time and the simulation time. The simulation time reacts to the clock and tries to maintain the same speed;

2 – *The analogue to digital converters*, which are responsible for reading the inputs at the beginning of each integration step, to make sure they are updated;

3 – *The digital to analogue converters*, set at the end of each integration step, and are responsible for the outputs update;

4 – *External Events* that are generated outside the analysis and react to the outputs given. They can be changes in values of some parameters, change of colours, or other visual displayed data.

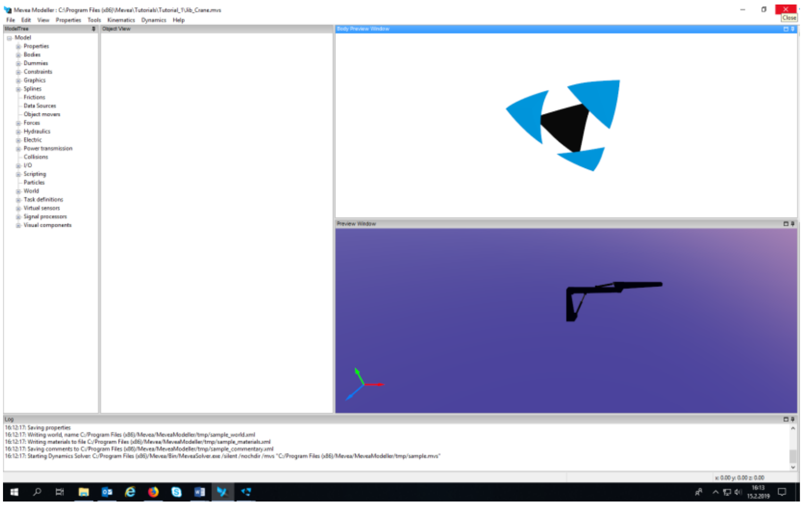
Mevea Software

The work in this thesis will be developed using one simulation software called Mevea. This software allows the simulation of bodies for testing and dynamic analysis purposes, as well as for creating trustful simulations. The software enables the creation of the model and its simulation due to the two existing interfaces: the modeller and the solver.

The modeller allows the creation of the different bodies and dummies, as well as the use of the physics system to impose constraints or describe where the loads should be applied. The collision graphics can also be added because the collision model is already implemented in the physics system. Physical and mechanical properties of each body are also added using the modeller. After describing the centre of mass, position, mechanical properties of the materials, and other important details for each part, it is possible to create the different inputs that should guide the assembly, like the motion it should have, the hydraulic behaviour of the different sub-systems. Finally, it is also in the modeller that the data sources are created. These files are responsible for storing important data such as the position of the different bodies, velocities, or the input signals.

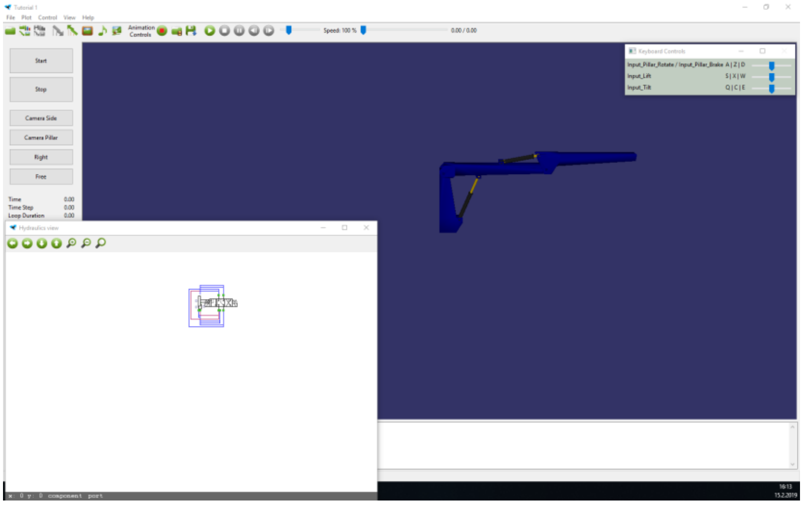
The solver, on the other hand, is where the simulation is shown, with the machine's behaviour responding to the different inputs created. From the data files created, the solver can display accurate graphs for the information regarding the necessary information asked. The evolution of these graphs occurs as the same time as the simulation, and consequently the dynamic analysis occurs.

Figures 2.18 and 2.19 show both of Mevea's interfaces. On figure 2.18, the modeller is presented, showing a crane and its menu. On figure 2.19, the solver exemplifies the crane with one of its hydraulic circuits and the keyboard input controls. The simulation allows seeing the functioning of the hydraulic circuit while the crane moves.



```
16:12:17: Saving properties
16:12:17: Writing world, name C:/Program Files (x86)/Mevea/MeveaModeller/tmp/sample_world.xml
16:12:17: Writing materials to file C:/Program Files (x86)/Mevea/MeveaModeller/tmp/sample_materials.xml
16:12:17: Saving comments to C:/Program Files (x86)/Mevea/MeveaModeller/tmp/sample_commentary.xml
16:12:17: Starting Dynamics Solver: C:/Program Files (x86)/Mevea/Bin/MeveaSolver.exe /silent /nochdir /mvs "C:/Program Files (x86)/Mevea/MeveaModeller/tmp/sample.mvs"
```

Figure 2.18 Mevea modeller



Time	0.00	
Time Step	0.00	
Loop Duration	0.00	

Input_Pillar_Rotate / Input_Pillar_Brake	A | Z | D
Input_Lift	S | X | W
Input_Tilt	Q | C | E

Figure 2.19 Mevea solver

3 *Gamification*

Although the gamification is still part of the methodology, it is in a different chapter due to its complexity and the approach chosen. In this chapter, all of the gamification part is addressed, like game elements, the software used to develop these elements, and how the user experience is important. At the end of this chapter, recent studies developed about game-like simulation and real-time dynamic analysis, and the applications of real-time simulation are presented.

Gamification has garnered a lot of attention in the last few years, with numerous applications, one of them being productivity (Remi-Omosowon, Cant, & Langensiepen, 2016). Many studies about using game-like environments have been developed in the last few years. Articles like the one Mark J Nelson wrote (Nelson, 2012) where it is discussed the use of game elements to improve workers' productivity and two different approaches, the American "Fun at Work" and the Soviet "Socialist Competition" can be the basis to discuss the introduction of gamified simulations into the mechanical simulation field.

First, it is important to describe what gamification means. It can be viewed as the introduction of game technology and elements to environments outside the game industry (Deterding;Dixon;Khaled;& Nacke, 2011). The introduction of game elements outside the gaming world can be used to influence people's behaviours and motivation (Van de Boer, 2013).

Specifically, in the mechanical engineering field, gamification can be used to optimize simulations for testing products by adding motivational elements, which aren't essential to the good functioning of the simulation but can improve the quality of the simulation. Performing tests in a virtual prototype becomes more attractive and the possibility of using a group of users to test it and collect feedback about the product creates the chance to have more accurate data, making it easier to improve the product. Suraj Jaiswal and others believe that users when testing any product, want to have the best experience. This can be achieved by motivating users to achieve goals, so that they believe they're contributing to the product's development. Users' goals are to have fun, explore and use the product. That way, manufacturers can develop a more customer-oriented product (Jaiswal;Islam;Hannola;Sopanen;& Mikkola, 2018).

It is possible to use game elements in simulation for other purposes, one of them being a training aid system for inexperienced users, making training safer and producing satisfactory results (Remi-Omosowon, Cant, & Langensiepen, 2016).

3.1 Game Elements

The concept of "game elements" can be abstract for some. Deterding and others propose a simple definition for what these are. They treat the game elements as a set of features shared by the game, but they are not necessary conditions for a game to exist (Deterding, Dixon, Khaled, & Nacke, 2011).

To understand what type of game-elements should be added, it is necessary to define the different types of users. Knutas and others propose a definition for the different types of players and the different game features each one relates to, summarized in table 3.1.

Table 3.1 Types of users, adapted from (Knutas, et al., 2018)

Player type	**Description**	**Provided tasks (features)**
Philanthropist	Motivated by purpose. They do not expect any reward	Tasks to help others, in need at the moment
Socialiser	Motivated by relatedness. They want to interact	Tasks that channel the socialization to upkeep the collaborative spirit of the system
Free Spirit	Motivated by autonomy. They like to create and explore	Tasks that channel exploration and reward the joys of discovery
Achiever	Motivated by competence. They seek to progress by performing	Tasks that are competitive
Player	Motivated by extrinsic rewards. They believe rewards are more important than tasks	Tasks that encourage work with others and building a community
Disruptor	Motivated by triggering chance. They tend to disrupt the system	Tasks to be involved in the developing of the system

After understanding the different type of users that exist, the next step is to comprehend how to build a good game/gamified environment. Adding game-elements for itself can sometimes have the opposite effect of what was expected. Starting with Knutas article, where it is said that the motivation of participants, goals and gameful design needs to be taken into consideration, otherwise, the features added will have no impact (Knutas, et al., 2018), it is possible to define an efficient way to create a successful gamified simulation, directed for the targeted users.

Mark Rosewater, a successful game-designer proposes a list of features that any game needs:

1 - The first item is a *goal*, or a set of goals, so that the game has a point, a purpose. It should be clear and simple to follow;

2 – There should be a set of *rules,* so players know what they can and cannot do. Rules also should make the goals harder to accomplish;

3 – *Interaction*. The game should have some aspects that interact with the user and vice-versa and (when it is the case) interact with other users;

4 – A *catch-up feature*. This feature means that it should be a way for a user who got left behind to still accomplish the goals and having a chance to win. It helps the user to keep motivated;

5 – *Inertia*. The game should have something to moves it towards completion. The game shouldn't be endless, because that way the user gets bored;

6- Elements that can *surprise the user*. There should be elements or moments that the user is not expecting. A surprise element helps keeping the user playing;

7 – *Element of strategy*. A player should get better with time. Users keep playing the same game because they expect to learn and improve from previous experiences;

8 – The game should be *fun*. The user needs to enjoy himself while playing. A game exists for entertainment purposes.

9 – *Flavour*. A game shouldn't be bland. Any game gets better if it is associated with a story or a theme instead of just being playable or efficient

10 – The last element is a *hook*, something that makes people want to try the game.

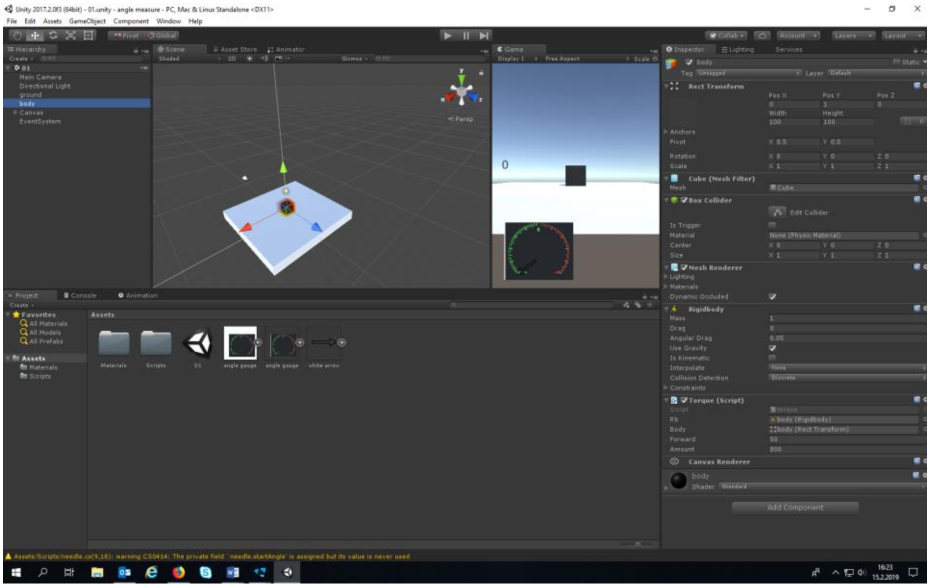
(Rosewater, 2011)

Manufacturers can choose what elements they want to introduce, according to what type of interaction they pretend. For example, when simulating an excavator, the game elements chosen can go from time constraints to goals and rankings, according to the performance of users (Jaiswal, Islam, Hannola, Sopanen, & Mikkola, 2018). Different game elements can motivate different users, making possible to receive a wider type of feedback of the product itself.

One last important point is that mechanical gamified simulations have a slightly different purpose when compared to "traditional" games. They might not need to fulfill all of the features presented, nor do need to aim at all types of users.

3.2 Unity software

For this thesis, to add game elements, Unity software can be used. Being a game engine, it can be used to work on two dimensional (2D) or three dimensional (3D) platforms. Although being a game engine, Unity can be used together with other platforms to create better simulations. In this specific case, the game engine software will be used for graphics and game-elements purposes, working together with Mevea. Figure 3.1 shows the software's interface. There are two main canvas, the scene one and the game one. The scene canvas is where the objects are created and added while the game scene shows what the user will see when playing the game. On the left side it is possible to see the list of objects created, ordered by hierarchy. On the right of the figure, there is the inspector box. The inspector allows to change, add, or remove properties from the objects created. These properties go from rigid bodies details to colours and textures. Finally, in the bottom, all the folders, files, scripts, images, and parts are shown. This way, the user can easily access to any file needed to work.



body Static
Tag Untagged Layer Default

Rect Transform
Pos X 0 Pos Y 1 Pos Z 0
Width 100 Height 100
Anchors
Pivot X 0.5 Y 0.5
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Cube (Mesh Filter)
Mesh Cube

Box Collider
Edit Collider
Is Trigger ☐
Material None (Physic Material)
Center X 0 Y 0 Z 0
Size X 1 Y 1 Z 1

Mesh Renderer
Lighting
Materials
Dynamic Occluded ☑

Rigidbody
Mass 1
Drag 0
Angular Drag 0.05
Use Gravity ☑
Is Kinematic ☐
Interpolate None
Collision Detection Discrete
Constraints

Torque (Script)
Script torque
Rb body (Rigidbody)
Body body (Rect Transform)
Forward 50
Amount 800

Canvas Renderer

body
Shader Standard

```
Assets/Scripts/needle.cs(9,18): warning CS0414: The private field `needle.startAngle' is assigned but its value is never used
```

Figure 3.1 Unity interface

Improving the graphics and adding game elements using Unity will allow to create a better experience for the user, producing better results for the manufacturer.

3.3 Mevea and Unity interface

The main software used to create the simulation model, has it has already been said, is Mevea. This means all physics principles, as well as collision detection and response, are derived from Mevea. Unity is only used for graphic reasons, and to create a better, more immersive simulation. The input controls are also given by Mevea.

Although Unity's ability to use its own physics system, for this type of simulation it does not work that well, being preferable to use a third-party software to mimic the machine's behaviour. As mentioned, Unity is a game-engine, not a dynamic analyser.

Mevea and Unity work together by using coding files that work as an interface between them (API). These files make Mevea act as a server and Unity to act as the host. These are files written in C# language linking both programs and allowing the simulation happening in Mevea to be reproduced by Unity. It also allows Unity to use the data stored in Mevea, like the real-time speed of the information about the different parts, if the user wants to use them to any game elements related to that.

3.4 User experience

User experience as a concepts can have many definitions, according to many different areas, but in a general way, user experience (UX) can be defined as how people use a product, the sensations associated to it, the understanding of the product, the feelings experienced while experimenting the product, and how the product serves its purpose (Alben, 1996)

The human-product experience englobes various disciplines, not only efficiency. Areas like aesthetics, sustainability, and functionality should be taken into consideration when orientating any product to a customer. Users search for pleasant experiences when trying any product, from all points of view. For example, one product can be very efficient mechanically but if it is not functional or pleasant for the customer, he/she might not be interested in it (Desmet & Hekkert, 2007).

Figure 3.2 gives an example of how the process should be user-centred, meaning that one product should be developed for the user, making sure all types of users are taken into consideration, from inexperienced to the more experienced, regarding every possible problem one could have.

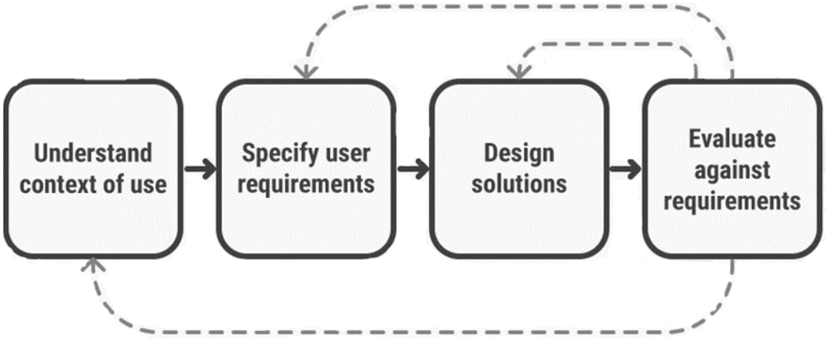

Understand context of use
Specify user requirements
Design solutions
Evaluate against requirements

Figure 3.2 UX user-centred, from (Interaction Design Foundation, 2002)

According to Desmet and Hekkert (Desmet & Hekkert, 2007), three things can be identified as relevant parameters for evaluating user-product interactions: the aesthetic experience, the experience of meaning, and emotional experience.

The aesthetics of a product is its capacity to pleasure the user's senses, pleasant to look at, pleasant sound, and/or pleasant to touch. The experience of meaning is related to the utility of the product, and the ability for the user to associate a product to a meaningful experience. Finally, the emotional experience can be defined as what emotions the product triggers in the user.

One way to evaluate if user needs are met is through the use of surveys. If using a consistent set of metrics, this method allows understanding if the expected impacts are achieved, or if the users feel that their needs are not being fulfilled. Surveys can explore the accessibility, usability, or desirability as parameters to explore the feedback given. The possible answers to the surveys should have a scale, where the lowest value stands for maximum disagreement, and the highest stand for maximum agreement (Ellwanger;Pierre da Silva;& Antunes da Rocha, 2018).

3.5 *Related work*

Like it was mentioned before, many studies about multibody dynamics, simulation, and gamification applied to mechanical simulation have been developed in the last few years. Next, some articles from the last 20 years are presented.

Starting in 2001, Larsson presented his thesis with the title "Multibody dynamic simulation in product development". It addresses the efficiency issue in product development in a more demanding environment, with customer's expectations getting higher along with the need to produce faster and with limited budgets. The author characterizes the simulation processes in order to raise its efficiency while introducing where the multibody dynamic analysis should occur. (Larsson, 2001). Figure 3.3, presented below, exemplifies this last part. The analysis should be performed after the detail design phase and before testing the product. By performing the analysis with the help of simulations, the time needed is relatively short and the computational cost is low, making the process faster and cheaper.

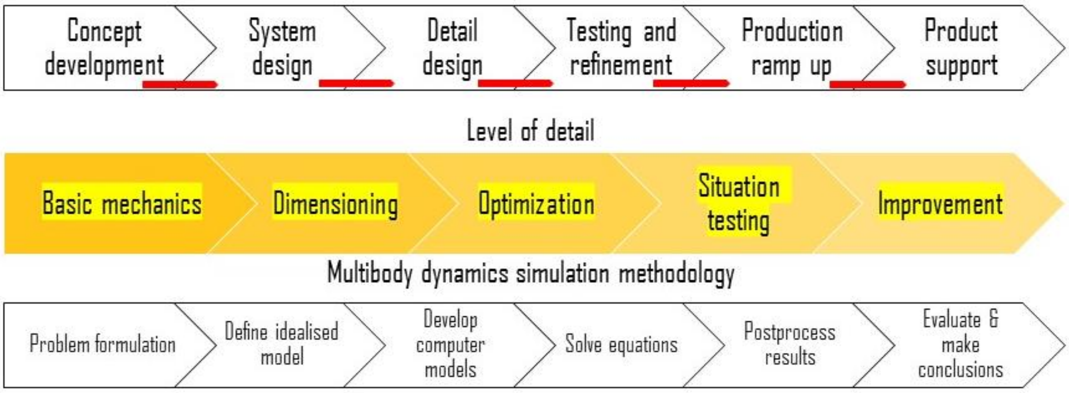


Figure 3.3 Multibody dynamics simulation in product development, adapted from (Larsson, 2001)

In 2016 an article, this time with the title "Applying Gamification Principles to a Container Loading System in a Warehouse Environment", written by Remi-Omosowon, Cant and Langensiepen, addressed the gamification process to solve real-world container loading problems. With the introduction of game-elements, user's confidence and engagement increases without the need to understand the complicated loading algorithms running in the background of the simulation, leading to better feedback and data sources. The simulation enables the users to better understand the process instead of reading the loading proceeds manual in plain text, which can be difficult to read and understand.

The simulation environment needed to recreate a visual representation of the warehouse with the placement of the pallets and the final goal, which was to re-arrange them in their correct place. The environment needed to be built to scale, so if the loads fit the simulation, they would fit in the real warehouse, making the process accurate. (Remi-Omosowon, Cant, & Langensiepen, 2016) . Figure 3.4 shows an example of the output expected, which was the arrangement of the pallets. It could be done in three different ways to fit all the pallets. The figure shows the different solutions from above. Two different types of pallets are considered, each one having a different colour associated with. From the simulation, the workers from the warehouse understand how to arrange the pallets faster, not needing to work in the real warehouse until they find the best solutions for the problem.

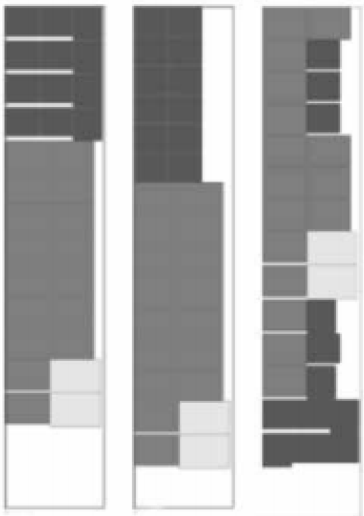


Figure 3.4 Three possible outputs expected for the arrangement of loads, from (Remi-Omosowon, Cant, & Langensiepen, 2016)

In 2010, Li, Yao, Wang, and Lou wrote an article named "Dynamic Multibody Simulation of a 6-DOF Robotic Arm" where they studied a robotic arm and compared the numerical results with the physical results, to understand if there would be any differences between the values obtained from both methods. Using the topological projection algorithm and Adams simulation software they were able to prove that the results expected from both approaches were similar, confirming the utility of simulation software tools in future works in product development (Li, Yao, Wang, & Lou, 2010). Figure 3.5 shows the model created for the robotic arm. It has a base, rotational parts, snatch parts, joints and driving parts.

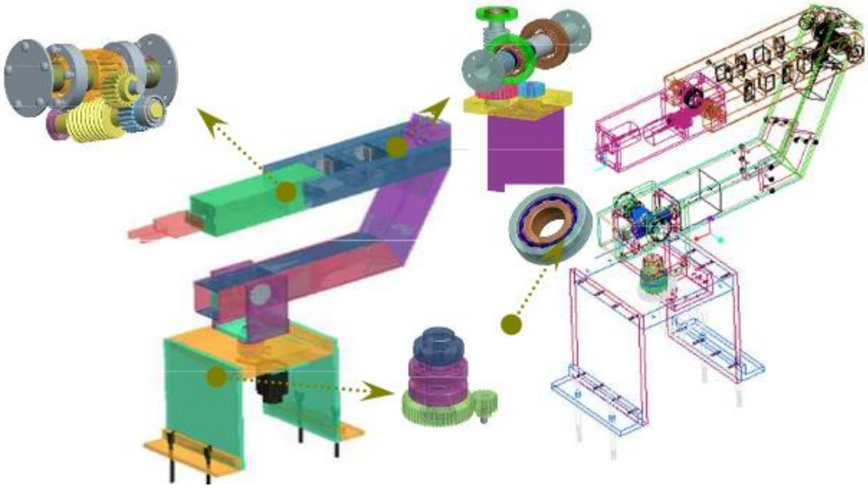


Figure 3.5 Model for the robotic arm, from (Li, Yao, Wang, & Lou, 2010)

In 2011, an article was written by Deterding, Dixon, Khaled, and Nacke, named “From Game Elements to Gamefullness: Defining “Gamification”” approached the gamification topic. It addressed the introduction of game elements in non-game simulations, taking inspiration from video games, and making it easier for mass-market consumers to be interested in this kind of technology. The main goal was to motivate and increase user activity and it grew rapidly to the design market. One of the main ideas was that these applications are suitable to larger audiences potentially creating new sources of data for a more customer-oriented industry (Deterding, Dixon, Khaled, & Nacke, 2011). Figure 3.6 illustrates the benefits of using game elements in a simulation. They can be separated into two groups belonging to the simulation part: the simulation interactivities and the simulation feedback. The interactivities can be considered as a motivational element while the Feedback group can be considered as the reward element, where users can see if their effort paid off. Both of these groups have the same importance.

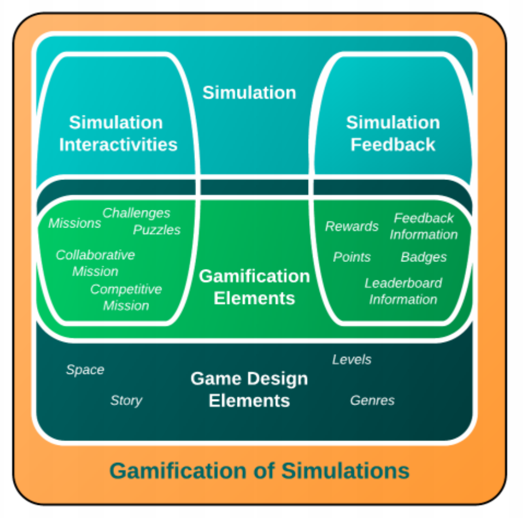


Figure 3.6 Gamification of simulations, from (Pirker & Guti, 2015)

In 2014 an article for the Institution of Mechanical Engineers was written, with the title “Real-time multibody application for tree harvester truck simulator” by academics Baharudin, Rouvinen, Korkealaakso, and Mikkola. Using a recursive formulation for performing a dynamic analysis and the lumped fluid theory for the hydraulics system, a real-time simulator was developed to improve vehicle and cutter operation as well as tree management (Baharudin, Rouvinen, Korkealaakso, & Mikkola, 2014). Figure 3.7 illustrates the example given as the simplification for the model considered. The harvester truck model has six bodies, each one with its local coordinate frame system. The expected movements are also represented. Bodies are simplified to make numerical integration easier, maintaining its accuracy.

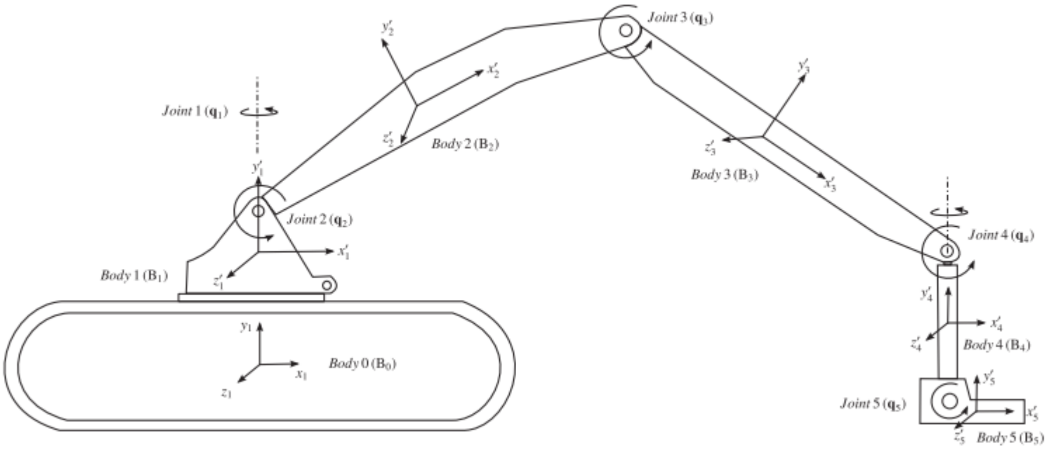


Figure 3.7 Tree truck harvester, from (Baharudin, Rouvinen, Korkealaasko, & Mikkola, 2014)

Finally, in 2018, another article was written by Jaiswal, Islam, Hannola, Sopanen and Mikkola with the title “Gamification Procedure based on Real-time Multibody Simulation”. It addressed the introduction of game elements in real-time multibody simulation. The machine the article worked on was an excavator and the objective was to see which combination of bucket size and hydraulic cylinder was better in terms of fuel consumption and time taken to perform the task. The study used a semi-recursive formulation to perform the dynamic analysis, using a relative coordinate system from a global coordinate system (Jaiswal, Islam, Hannola, Sopanen, & Mikkola, 2018). Once again, the topics of gamification and introduction of game elements are addressed. The goal of game-elements’ introduction was to increase the involvement of users (Jaiswal, Islam, Hannola, Sopanen, & Mikkola, 2018), turning a simple simulation into an immersive experience for the user, improving the feedback and ultimately, helping product development for a more customer-oriented product.

The excavator model is presented in figure 3.8. It has nine bodies, 10 joints and 11 degrees of freedom. The driver can control six hydraulic cylinders to control the arm’s movement and crane’s rotation. The ground is modelled using a granular particle system (Jaiswal, Islam, Hannola, Sopanen, & Mikkola, 2018)

The game-elements are introduced to perform tasks and to improve the user-machine relationship. Elements such as time constraints, speedometer, fuel consumption gauge are introduced and can produce different outcomes, depending on the combination of the bucket and hydraulic cylinders chosen. Figure 3.9 shown below, shows the interface pretended. The main goal is to fill the industrial hopper.

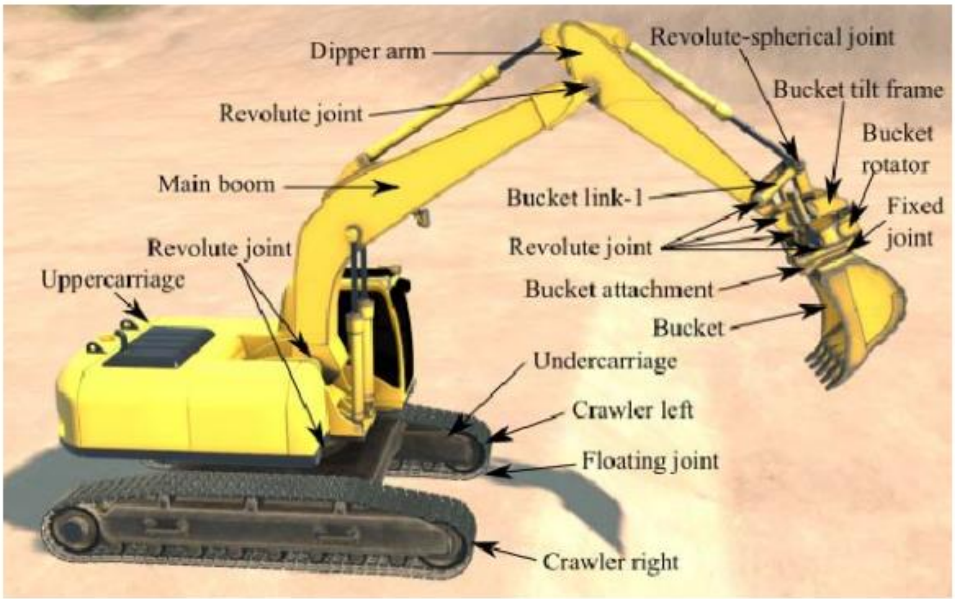


Figure 3.8 Excavator model, from (Jaiswal, Islam, Hannola, Sopanen, & Mikkola, 2018)

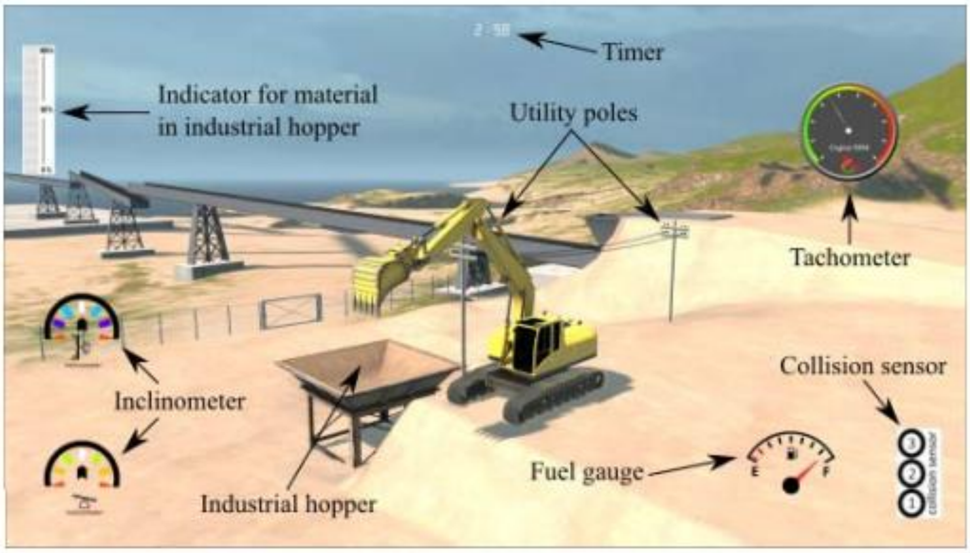


Figure 3.9 Simulation interface, from (Jaiswal, Islam, Hannola, Sopanen, & Mikkola, 2018)

4 *Case Study*

In this chapter, the case study is introduced, with its parts and their purpose being described. The simulation model will be presented with the different tasks proposed. Finally, the gamification topic will be addressed, as well as the game elements created.

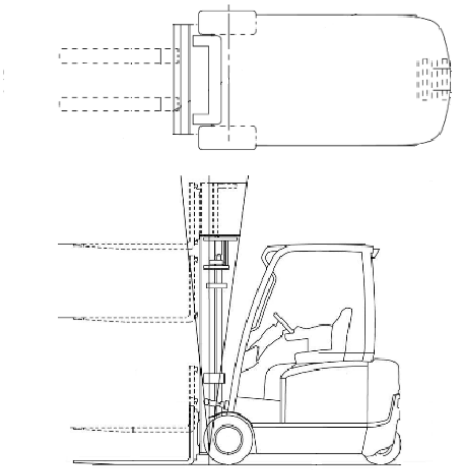
4.1 Forklift

The study case for the multibody dynamic analysis and simulation is a forklift from Mitsubishi, the EVOLT48 3 W 2.0-ton forklift. The main features are:

- drive operation: automatic (electric)
- motor type: AC
- lifting capacity: 2000 kg
- lifting height: 4.0 m
- mast type: duplex

The forklift is equipped with some technology that makes the driving easier and safer. The driver's panel has a set of buttons with images, to control the movements, instead of the traditional levers, making it easier for the driver to understand the forklift's behaviour. The machine is also equipped with a "sensitive drive system", a system capable of helping the driver while he/she drives, increasing the machine's stability, for example.

Figure 4.1 shows the forklift draw as presented in the catalogue, with the load carrier positioned in its different possible positions, along with some of the forklift's characteristics. It is also possible to see how much the masts can tilt.



Some of the forklift's measurements are:

- total length: 3119 mm
- height (with mast up): 2125 mm (4335 mm)
- load carrier (fork): 1150x100x35 mm
- chassis width: 1140 mm
- tilt (front/back) degrees: 6/10

Figure 4.1 Forklift EVOLT48 3 W 2.0presented in the catalogue from (Mitsubishi, 2019)

Figure 4.2 shows the forklift's concept. The most important parts are indicated. The forklift has a main body where the driver is seated and where the two motors are placed. The first and second stage masts are responsible for securing the load carrier and its movements. Finally, the front and backwards movement is guaranteed by its two front wheels. They are connected to the main electric motor. The rear tyres are connected to the rear axle which is connected to the steering wheel. The turning of the machine is given only by its back wheels, which can turn 90 degrees both sides.

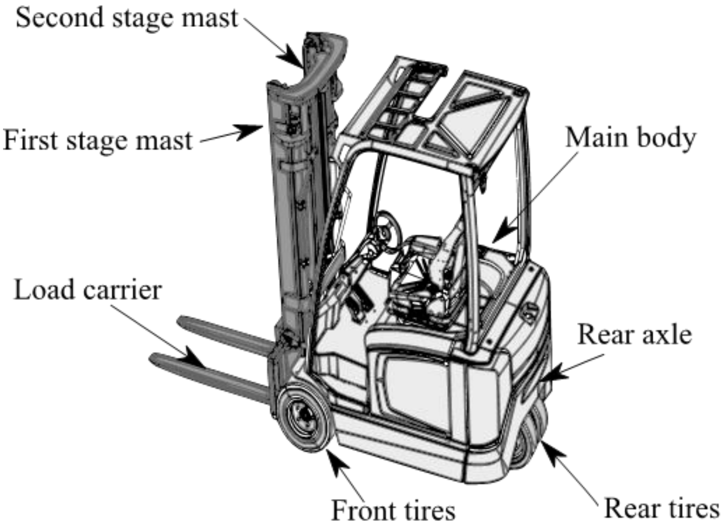


Figure 4.2 Forklift model

The main schematics used to represent the model are shown in figure 4.3. As seen in this figure, two types of joints are used for the entire model: translational and revolute ones. There is one revolute joint connecting the main body to the rear axle which is responsible for the movement of the rear wheels. To allow the tilt of the first and the second masts, the main body is also connected to the first stage mast by a revolute joint. The load carrier needs to be able to go up and down without rotation along the second stage mast, and this one needs to do the same movement relatively to the first stage mast. Because of that, there are two translational joints connecting the three parts, giving the ability to the load carrier to go up and down accordingly to the position of the load.

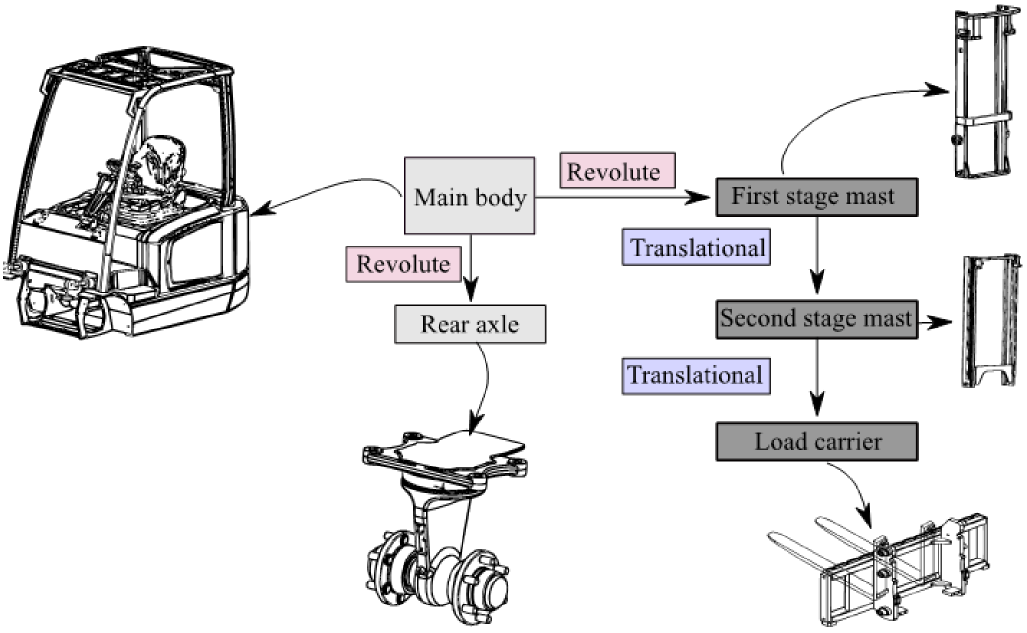


Figure 4.3 Joint schematics

To understand how the movement happens, figures 4.4 and 4.5 show a simplified schematic for the front and rear wheels, respectively. Because the motor is supposed to work at high speeds, as well as at slow ones, there is a need to use a gearbox to increase the torque transmitted to the front axle and consequently reduce the speed induced into the wheels. If there wasn't a gearbox installed, to work at such small velocities, the motor would need to be bigger, being more expensive and less efficient. The front axle has two planetary gears, each one connected to one end of the axle. One of the benefits of using planetary gears for each wheel is the fact that the axle can transmit more torque and be able to keep the machine moving when carrying heavier loads. It also allows the machine to work more silently.

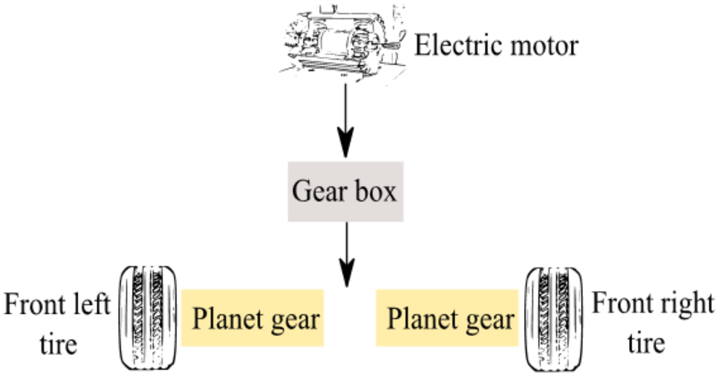


Figure 4.4 Front wheels schematics

The back wheels, represented in figure 4.5, are only responsible for turning the machine, thus making this sub-system simpler. A different motor is responsible for the wheels' turning movement. It can be considered as being directly connected to the rear axle (for simplification purposes). Because these wheels are not responsible for the machine's translational movement, there's no need to use planetary gears at each end of the axle.

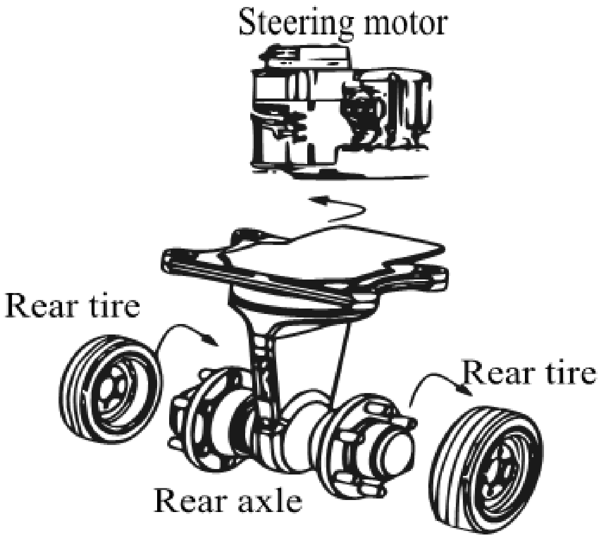


Figure 4.5 Back wheels schematics

4.2 Simulation model

As mentioned in the previous chapters, Mevea and Unity are used to develop the forklift's simulation.

The model is a simple representation of the real system. The graphics are imported according to the manufacturer's specifications and in a suitable file format. The simulation model's goal is to be a truthful representation of the real model. To do that, the model must respect the design parameters imposed by the manufacturer. This means that the different parts need to be in their correct place with the correct scale, with their local coordinate systems placed in the right places, and with their physical characteristics being accurate.

Figure 4.6 shows the forklift's model used to perform the analysis and the simulation. The model looks similar to the real forklift because of the graphics imported. It is important to make sure that the virtual model has the same properties as the physical one.

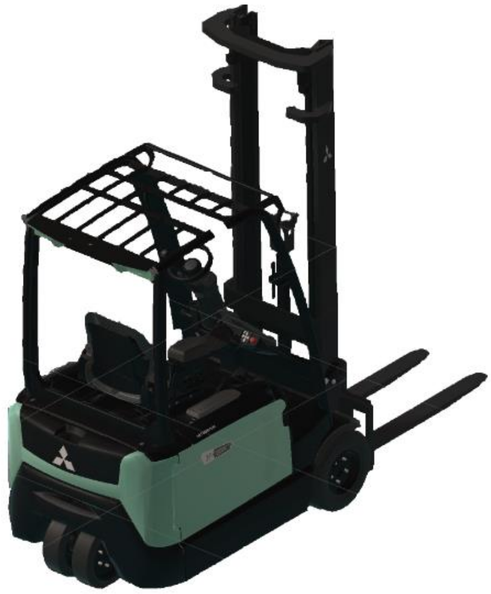


Figure 4.6 Forklift model

There are two types of parts in the model. The so-called bodies and the dummies. Bodies have physical properties like moment of inertia and centre of mass and are considered the most important parts of the system. The bodies considered are the main body, the first stage mast, the second stage mast, the rear axle, the load carrier, and the pulley holder. All of them are considered as being rigid bodies, not presenting an elastic behaviour. The relevant characteristics of these bodies are available in appendix 1. Some of these bodies have more than one component, but they are considered as one, in order to simplify the model. The main body, for example, which includes the driver's cabin, the mainframe, and a counterweight placed below the driver, is considered as being only one single part, combining all the small parts mentioned.

The dummies, on the other hand, although being considered as rigid bodies, have no dynamical properties. They increase the mass and inertia properties of the overall system but have no impact on the analysis outcome. This happens because Mevea considers these objects as not having any impact on the overall system when external loads are being applied. The individual bodies behaviour does not change when the analysis occurs. They are rigidly attached to the bodies that have loads applied on them without affecting the results. They behave like the bodies they are attached to. All the tyres, the hydraulic components, and the chains are considered dummies and there is no need to know their characteristics for the sake of dynamic analysis. The tyres present an elastic behaviour, that allows the machine to wobble and vibrate.

The graphics are created using AutoDesk 3d studio and were already built. Mevea was used to attach the right graphics to the right bodies created and to create the final assembly of the system. Figures 4.7 to 4.10 show the bodies considered for the dynamic analysis. These parts do not have any colour attached to them when treated as individuals. It will only be added when the final assembly is put together and when using Unity. The second figure presented (figure 4.8) is the chain pulley system used for the lifting part, with the two chains in the correct place with the pulley and pulley holder. This figure is important for a better understanding where the different parts assemble together.

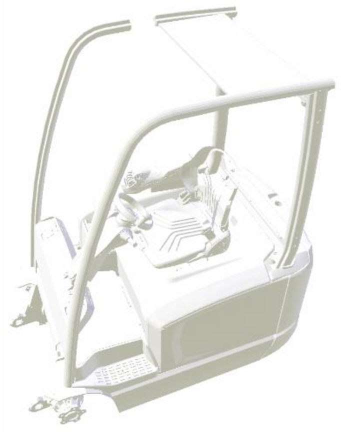


Figure 4.7 Main Body

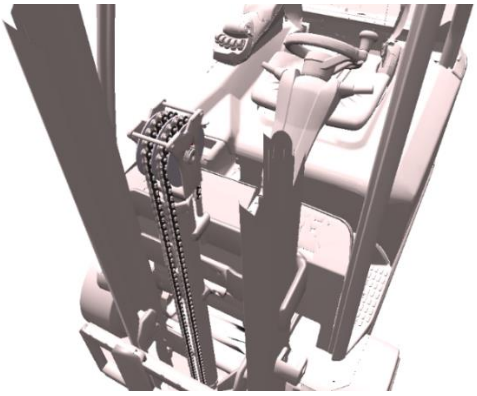


Figure 4.8 Chain pulley system

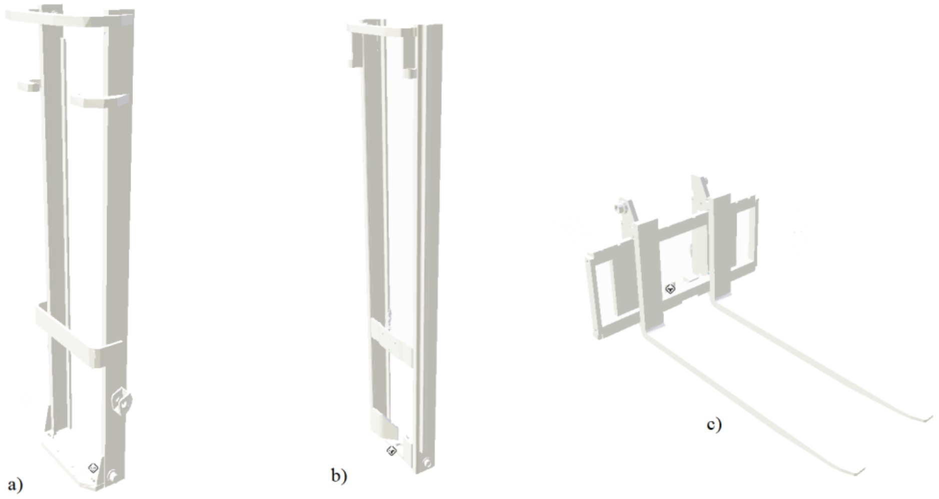


Figure 4.9 a) First stage mast b) Second stage mast c) Load carrier

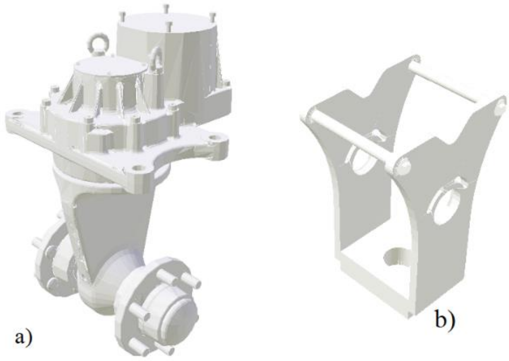


Figure 4.10 a) Rear axle b) pulley holder

Dummies are presented in figures 4.11 and 4.12, starting with the tyres and the pulley that helps the chain to move (figure 4.11). The back and front tyres are equal in size and shape, with the only difference being the weight. The front wheels, because they´re responsible for the front and back movement weight about 35 kg while the back wheel have a weight of 15 kg.

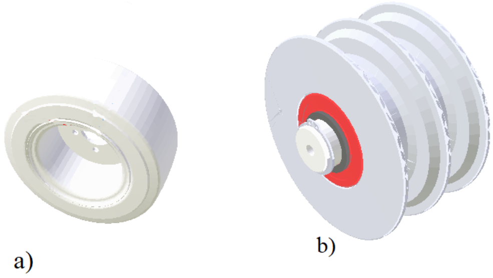


Figure 4.11 a) tyre b) pulley

When the pulley system (figure 4.8) is used together with the lifting hydraulic circuit (figure 4.12), it allows the load carrier (figure 4.9 c)) to lift a maximum weight of two tons without compromising the structure. The lifting cylinders (figure 4.12 a)) have two supports to guarantee that their position does not change and the lifting occurs evenly.

To allow the masts to tilt, two cylinders, like the ones presented in figure 4.12 c), one on the left side and the other on the right side, work together, making the tilt hydraulic system. The tilt cylinders are thinner than the lifting ones and do not need the supports.

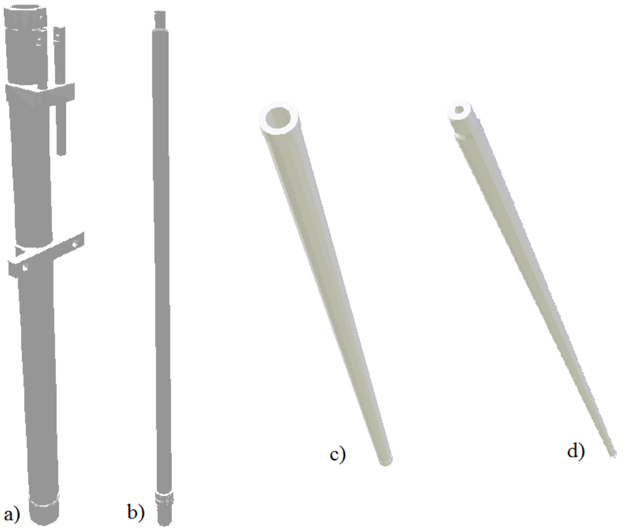


Figure 4.12 a) Lift cylinder b) Lift piston c) Tilt cylinder d) Tilt piston

System analysis

For the dynamic analysis, there is a total number of seven bodies considered: the main body, the first stage mast, the second stage mast, the load carrier, the rear axle, the pulley holder, and the lifting load. The total number of dummies is 16 and these dummies include the four tires, one battery, 10 hydraulic components, and a load to be carried. Some of the objects considered as dummies cannot be considered as merged to the main bodies they are connected to because, for the software used, it is necessary to indicate which dummies are supposed to apply loads, and for that reason they need to be assigned as separate entities. The mass of the system, considering and not considering the weight of the dummies is shown in table 4.1. The mass of the load is not considered.

Table 4.1 Weight of the system

System	**Without dummies**	**With dummies**
Weight (Kg)	3071.015	4294.015

Because the simulation should occur in a 3D environment, each body will have six possible DOF, along the three main axes, X, Y, and Z, and rotation along those axes, θX, θY, and θZ, giving a total of 42 generalized coordinates for the seven bodies considered.

Seven joints are used in the system: three translational joints, two revolute joints, and two floating joints. The floating joints are imposed by Mevea to keep the bodies from floating in the air, making them behave accordingly to the physics principles. The joint's use is:

- Ground – Main Body: Floating
- Main Body – Rear Axle: Revolute
- Main Body – First Stage: Revolute
- First Stage – Second Stage: Translational
- Second Stage – Load Carrier: Translational
- Second Stage – Pulley Holder: Translational
- Ground – Lifting Load: Floating

The joints DOF's are explained in table 4.2. Floating joints are created by Mevea to allow the physic system work, and for that, they need to be considered. From analysing the table, it is possible to infer that there is a total number of 25 constraint equations, resulting in a total of 17 remaining DOF for the system.

Table 4.2 Joint specifications

Joint	**Translational**	**Floating**	**Revolute**
DOF	1	5	2
Constraint Eq	5	1	4

Regarding the Jacobian matrix for the system, it will be a matrix composed by 25 rows, the number of constraint equations, and 43 columns, the number of generalized coordinates.

Working in a 3D environment means the mass matrix for each body will be an index six square matrix, with three rows and columns associated to translational mass, and the other three rows and columns associated to inertial momentums. As mentioned before, the mass matrix, for each body, can be written as:

$$\mathbf{M^i} = \begin{bmatrix} m^i & 0 & 0 & 0 & 0 & 0 \\ 0 & m^i & 0 & 0 & 0 & 0 \\ 0 & 0 & m^i & 0 & 0 & 0 \\ 0 & 0 & 0 & \bar{G}^{iT} i \bar{G}^i_{11} & \bar{G}^{iT} i \bar{G}^i_{12} & \bar{G}^{iT} i \bar{G}^i_{13} \\ 0 & 0 & 0 & \bar{G}^{iT} i \bar{G}^i_{21} & \bar{G}^{iT} i \bar{G}^i_{22} & \bar{G}^{iT} i \bar{G}^i_{23} \\ 0 & 0 & 0 & \bar{G}^{iT} i \bar{G}^i_{31} & \bar{G}^{iT} i \bar{G}^i_{32} & \bar{G}^{iT} i \bar{G}^i_{33} \end{bmatrix} \tag{77}$$

The mass matrix for the whole system will be an index 42 square matrix, considering the seven bodies, which one with an index six square matrix. The first 21 rows and columns are associated with the translational mass of the system, while the remaining rows and columns are associated to the inertial moments of the system.

As mentioned in chapter 2, to perform the analysis, the relation between global velocities and relative joint velocities is needed, and that relation can be expressed through the velocity's transformation matrix, **R**. For this system, this matrix will have 42 rows, the number of generalized coordinates, and 17 columns, the number of free DOF.

Hydraulic circuit

For this dynamic analysis and simulation, the system's hydraulic circuit can be simplified. The real hydraulic circuit can be very complex, and, for simulation purposes, it is not necessary to represent all its components. It is only necessary to represent the main components, the ones who have a real influence on the machine's behaviour.

The simplified hydraulic circuit used for the model is presented in figure 4.13. The model works with five main cylinders. Cylinder one is responsible for lifting the load carrier in the first phase, from the initial position to the end of the second stage, and it will be called pulley cylinder, while cylinders two and three, called the main lift cylinders, work together at the same time and are responsible for lifting the second stage mast, and consequently perform the second phase lift for the load carrier. Cylinders four and five are responsible for the first stage mast's tilt and work in the same way as the cylinders responsible for lifting the second stage mast. Numbers eight and nine represent directional valves (DV) and are responsible for controlling the movements. Number eight is a 4/3 valve responsible for the lifting movements while number seven, which is also a 4/3 valve, is responsible for the tilt movements. Numbers six and seven work together in the lifting movement and can be considered as a safety measure. Number seven is a counter-balance valve that allows the fluid to go up to the lifting cylinders but does not allow it to go the opposite way, so when the load carrier is working with heavier loads, it does not allow the cylinders to fail. Number six's purpose is to compensate for the counter-balance valve. When the input makes the cylinders go to their initial position, this 2/2 directional valve opens, allowing the fluid that can't flow through the counter-balance valve to flow through that path. Finally, numbers 10 and 11 are, respectively, a volume pump and a volume tank. The first is responsible for distributing the fluid across the system while the second works as a reservoir for hydraulic fluid.

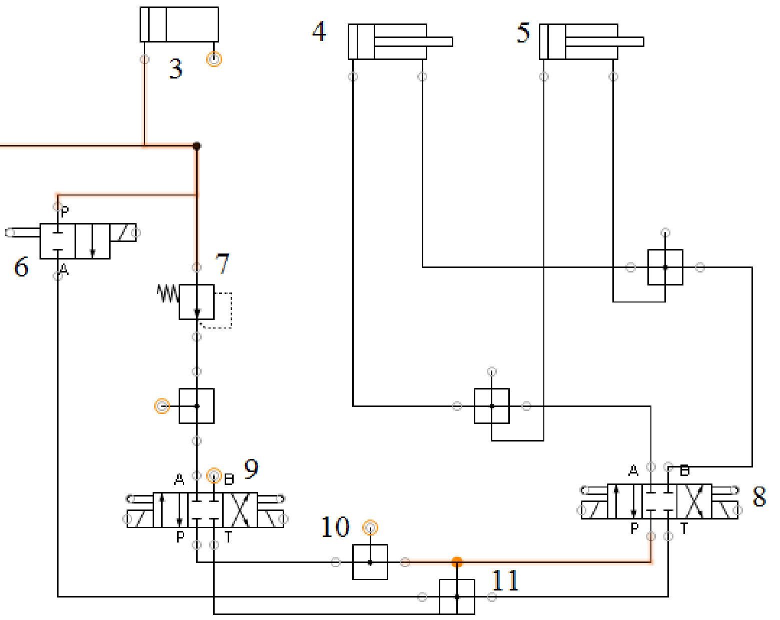


Figure 4.13 Simplified hydraulic circuit. 1- Pulley lift; 2- First stage mast right lift; 3- First stage mast left lift; 4- Mast right tilt; 5- Mast left tilt; 6- DV22 Lift; 7- Counter-balance lift valve; 8- DV43 tilt valve; 9- DV43 lift valve; 10 – Volume pump; 11 – Volume tank

Reference data for the hydraulic components is presented in table 4.3. The first part of the table presents the different heights and diameters of the cylinders, as well as their weight a surface area. The tilt and lift cylinders have identical properties. This table also presents the reference values for the hydraulic components, like the existing pressure to perform the movement, the flow rate expected, and the force the cylinder produces, considering the movements without a load and with a 2000 kg load. The change of the pressure available from one situation to the other, and consequently the change of the available force from the cylinder, is possible due to the hydraulic pump, making sure the system can manage the heavier loads.

Table 4.3 Hydraulic components data

Component		**Lift Cylinders**	**Pulley Cylinder**	**Tilt Cylinders**
Measurements	Cylinder length (m)	0.07	0.063	0.07
	Cylinder diameter (mm)	70	63	70
	Weight (kg)	15	15	15
	Piston surface area (mm^2)	3848	3117	3848
Reference values for no load	P (MPa)	101.9	20.6	101.9
	Q (l/min)	184.7	35.5	184.7
	F (N)	3777	3237	3777
Reference values for 2000 Kg	P (MPa)	128.5	152.2	128.5
	Q (l/min)	184.7	35.5	184.7
	F (N)	23397	42477	23397

4.3 Environment

Mevea allows creating environments for the simulation. The environment works as a background and can interact with the model being simulated. It has a dome shape with sky and terrain graphics. It can also have weather conditions, although for this case study they have no real impact. All physics principles are considered, making all simulation objects behave like real objects, without the possibility of floating and simulating real collisions.

For the forklift simulation, the environment is a warehouse with its surroundings, more specifically with containers, streetlights and some obstacles. Figure 4.15 presents this environment, with the warehouse being presented in blue. The containers presented the first part of the simulation interact with the forklift in a way that is possible to collide with them, just like a real environment. All the other existing obstacles try to simulate the different objects that can exist in an environment like this, creating the necessity for the forklift to adapt.

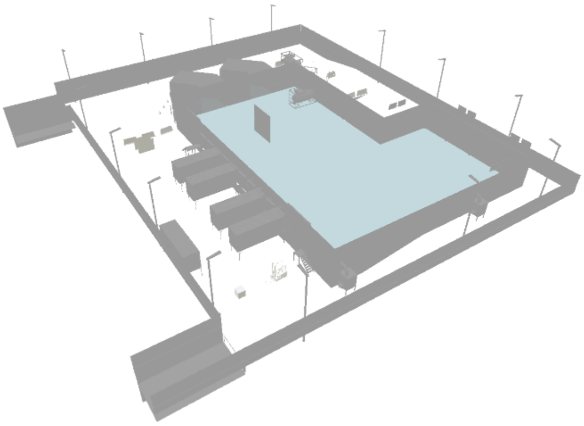


Figure 4.14 Simulation environment using Mevea

It is not possible for the forklift to go out of this environment nor going through walls or fences. All of those are considered solid bodies.

Using Unity, it is possible to make the simulation’s environment more real, with the possibility of adding lights, colours, and textures. Figure 4.15 shows the same environment as in figure 4.14 but using Unity. It is possible to compare both figures and understanding the benefits of using Unity for representation reasons. All the physic principles used for Mevea are passed to the Unity environment.

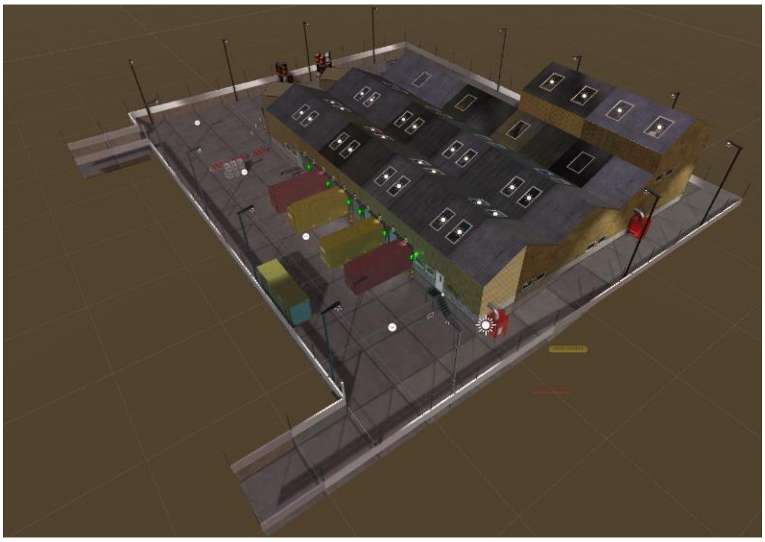


Figure 4.15 Simulation environment using Unity

4.4 Gamification

Gamification's purpose is to make the forklift's simulation more attractive with fun elements to encourage the user to test the model. It involves the different game-elements created, the different tasks proposed and how the user can control the simulation, with the different input signals possible to be given.

For research purposes, it is preferable to have more users testing the simulation with different levels of knowledge. To achieve this, the simulation should be appealing for everyone, or at least, for the biggest group of users possible. The more users try and test the simulation, the more information is available, and researchers are able to understand the existing problems better and make the best, most efficient product for the client. Figure 4.16 shows where the gamification process should occur when performing a simulation. It is necessary to define what tasks should be performed, in order to add the right game-elements.

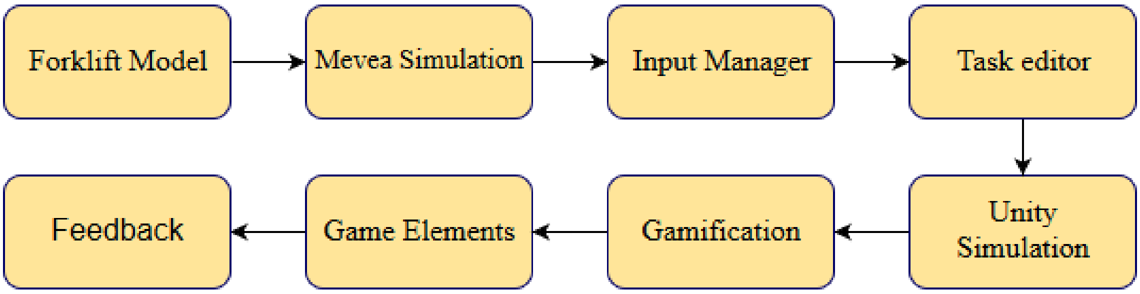


Figure 4.16 Gamification the simulation process

Users will be encouraged to test the forklift model, giving feedback about the experience after it is finished. One of the aspects for making the simulation more attractive is the possibility to try a first-person view. VR simulation, using a third platform is also possible, to create a more immersive environment, with visual, auditive, and touch stimulus. The simulation should occur in a simulator, prepared for this type of simulations, that reacts to the forklift's movement.

User experience

It is necessary to define what the simulation should aim for, its main goal. The main goal for this simulation is to collect information from users, concerning two aspects. The first should be information about the forklift, regarding its performance, easiness to control, and the experience of driving the machine. The second one should be information about the simulation, regarding the performance, game-elements added, and the environment.

To collect this data, and for positive feedback, the simulation should aim at meeting the user's needs. The needs go from simplicity to efficiency. It is not enough to provide only a functional simulation. It should be an immersive environment, that can accurately represent the real conditions. The usability of the machine should also be accurate, in such a way that the differences between the simulation and the real machine are minimal. It is not enough to fill the parameters given but instead, there should be a concern about other factors, such as realness, motivation, or the experience of the simulation. The user should be satisfied after the experience.

Five user needs were identified. Simplicity, efficiency, functionality, accuracy and fun are considered the most important aspects to fulfil, in order to have a successful simulation. These five parameters were then deconstructed to smaller needs that help to fulfil the main aspects. The necessity of having controls, the machine behaviour, the immersiveness, game goals and rewards, as well as stimulus were also identified as important user needs. A representation of the user needs identified is shown in figure 4.17. Many of these smaller needs are interconnected.

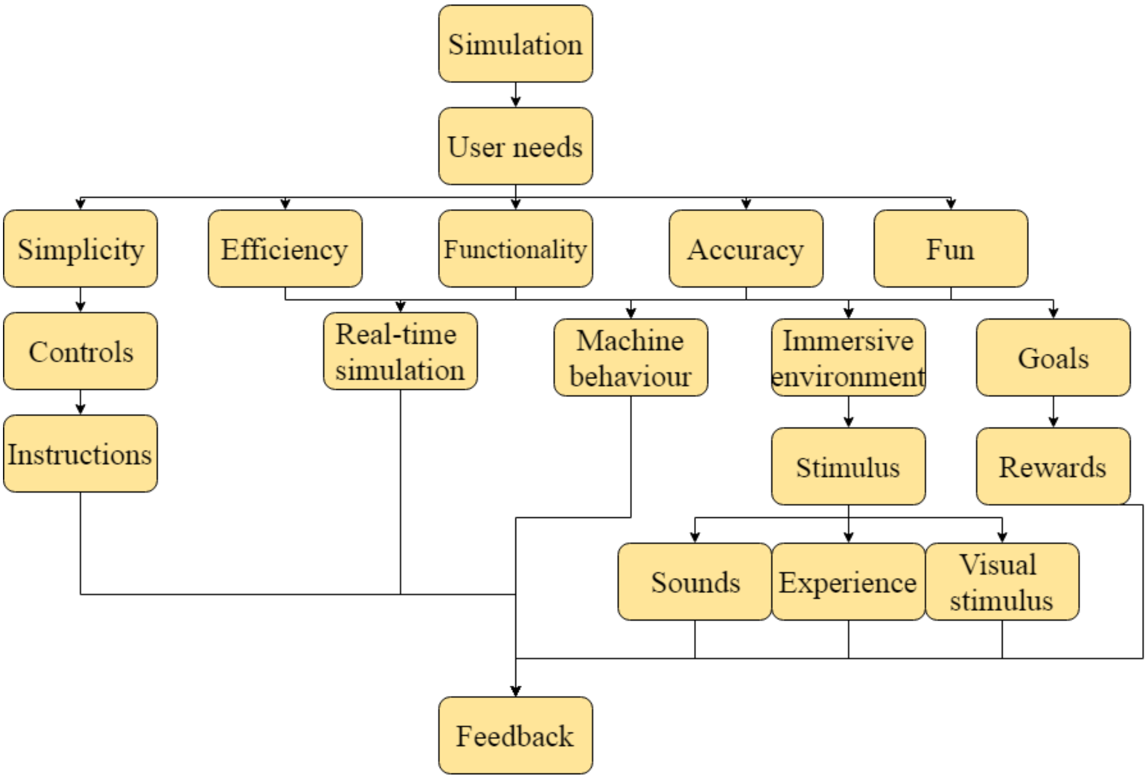


Figure 4.17 User needs

To try to meet all these parameters, the simulation must follow different directions, from the elements displayed on the screen to the interaction the user has with the simulation itself.

Starting with the simplicity aspect, users require the simulation to be simple, although the necessity to drive a forklift. For this, the controls (inputs) need to be simple and the instructions should be clear and easy to understand. The instructions can go from simple text, explaining how to perform, to hovering text displayed in the environment, to guide the user. Game-elements are important to help achieve this. If this need is met, the simulation should be accessible to everyone, from the more experienced to the totally inexperienced user.

Efficiency, functionality, accuracy, and fun are all connected and need to be analysed as a whole. The whole point of this simulation is to mimic a real forklift, so the model needs to behave like the physical model. The physic system and the dynamic analysis need to be performed well. Actions like crashing, carrying loads, or just turning the forklift need to be accurate. The user should feel the wheels trembling with the machine's vibration, should feel the difficulty of turning the forklift when carrying different loads, and feel the sudden break. Still regarding the functionality of the simulation and machine behaviour, users should be able to feel comfortable when performing, without dizziness or confusion.

The environment where the simulation occurs must be immersive. To achieve this a VR set is used, which guarantees the existence of visual and hearing stimulus, with the possibility of having a first-person view from the cabin and hearing the sounds the forklift does when performing, as well as being able to see his own hands. This way, users can have a better and more accurate experience, being more likely to have fun while trying the simulation. Game elements are, again, important to help visual stimulus and to keep the user interested in performing. On-screen elements that react to movement and inputs help to improve the experience.

Finally, the element of fun is met with the introduction of goals, constraints and rewards. As a gamified simulation, game elements were added to increase the gaming aspect and to try to get the biggest number of people interested. For example, a competition element helps increasing the motivation, that leads to an increasement of the fun.

To evaluate if the user requirements are met, an analysis regarding the simulation and the gamification will be made.

Game elements

Figure 4.18 shows the simulation without any element added. The simulation looks bland and unattractive although colour is added. If the simulation stayed like this, users wouldn't feel motivated in trying the model.

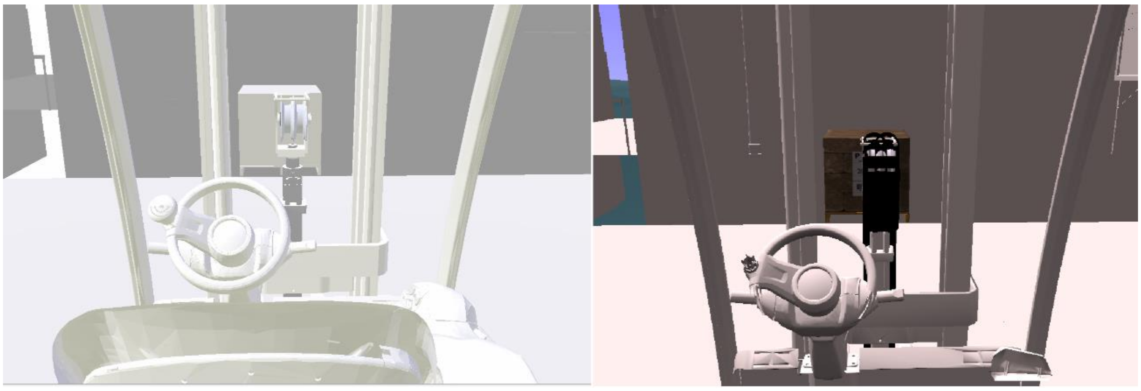


Figure 4.18 Mevea simulation display

The first step to introduce game elements is to understand how to make the users feel enthusiastic about this simulation. The elements proposed can be divided into two main groups, presented in the figure 4.19 for a better understanding. The first group englobes the information, while the second one is about the motivation/competition elements. The two groups are explained next.

The information group englobes every game element that can provide any type of information. It can be subdivided into four smaller groups: general information, goals, data from inputs, and information in the environment. The first one contains information about how to control the forklift's movement, from the forward and back movement to the lift's height and tilt. This information can be displayed on the environment, on the user's view, or can be provided beforehand. The goals appear at the beginning of the simulation and tell the user the tasks he should perform. This information is shown on the screen for a fixed amount of time, disappearing after. The data from the inputs is important for the user to understand what is happening. This type of information includes the gear which is on or information about the stage's height and tilt. It is displayed on the screen so that the user can always check it. Finally, the environment can also have information for the user, like indicators showing the correct way, the place where the user must be, etc. Like the name says, this type of information is displayed on the environment.

The motivation/competition group can also be subdivided into smaller groups: the first one being the constraint elements and the second one being the motivation elements.

The constraint elements, as the name says, are used to constrain the user. A time constraint, as well as collision consequences, can be used to motivate the user while performing correctly. On the other hand, motivation elements tend to make the user work harder around the simulation, giving it a more real feeling.

Being a gamified simulation, it is important to have a scoring system, that rewards good actions and punishes bad ones (like collisions or not performing the right tasks). It is also important to try the mimic the car features. For that, car elements can be added, like a speedometer and a steering wheel angle displayer.

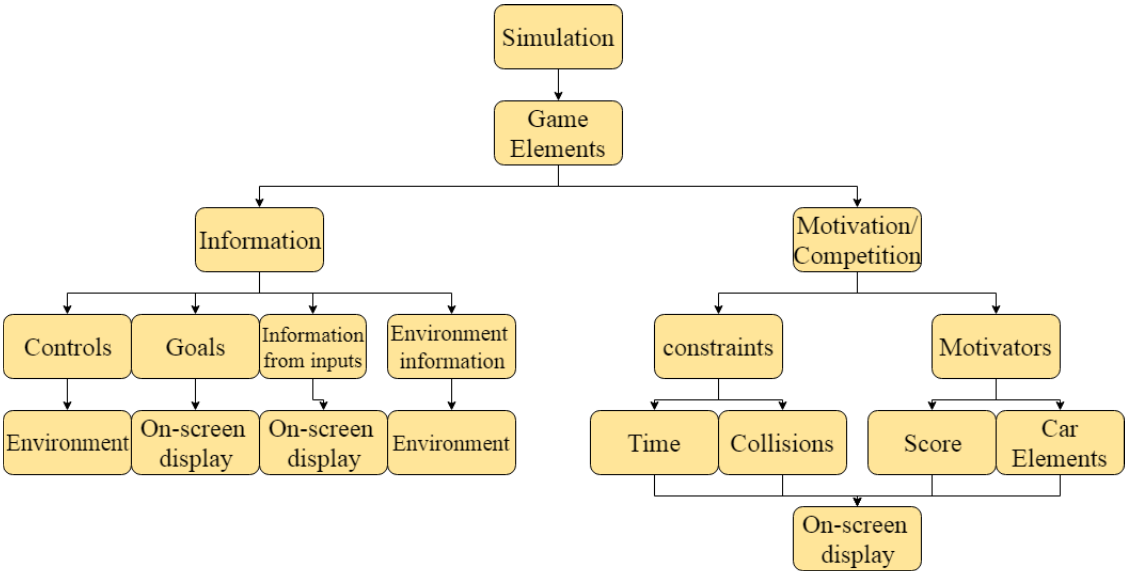
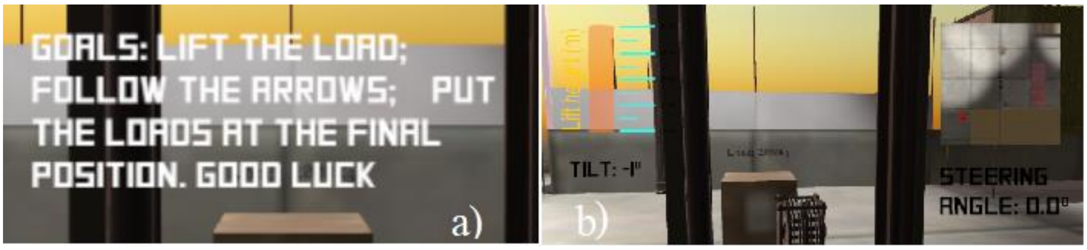


Figure 4.19 Game-elements

The information elements are particularly important. They should explain to the most inexperienced user how to control the forklift, how the scoring system works, and what he should do to perform well. The on-screen information, as well and the one existing in the background/environment needs to be visible, so the user can read it clearly. For this purpose, it was decided to create some instruction elements.

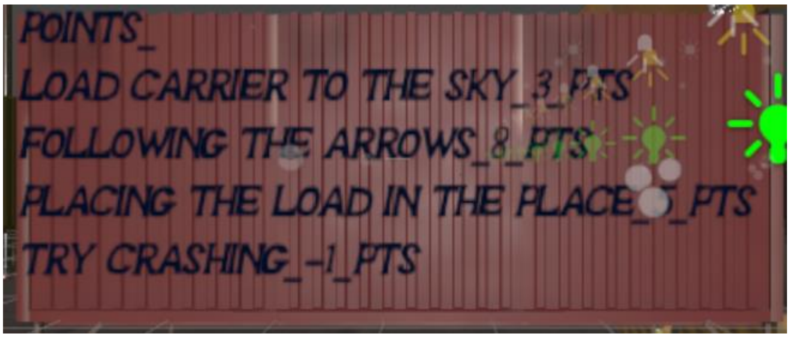
Figure 4.20 shows the on-screen information. Figure 4.20 a) shows the goals, visible in the centre of the screen. This will be the first thing the user will see and read, so his attention should focus immediately on these instructions. After a fixed amount of time, this panel disappears, giving place to next information presented in 4.20 b). Figure 4.20 b) shows the information from the inputs. The height and tilt are displayed in the top-left corner of the screen. The information is displayed in a way that the user can easily understand it without feeling confused or overwhelmed. An inexperienced user might feel lost during the simulation. Because of that, a mini-map is displayed in the top-right corner of the screen.



GOALS: LIFT THE LOAD; FOLLOW THE ARROWS; PUT THE LOADS AT THE FINAL POSITION. GOOD LUCK

Figure 4.20 On-screen information a) Goals b) Input information

Figures 4.21 and 4.22 show the environment information. Figure 4.21 shows how the scoring system works. Positive tasks are rewarded, and bad ones are punished. These instructions are placed at the beginning of the simulation, next to the forklift, on one of the containers. These instructions are not displayed in the on-screen information because the user would get lost with the amount of information. When the user starts working the simulation around, he is able to see these instructions.



POINTS_
LOAD CARRIER TO THE SKY_3_PTS
FOLLOWING THE ARROWS_8_PTS
PLACING THE LOAD IN THE PLACE_5_PTS
TRY CRASHING_-1_PTS

Figure 4.21 Score system information

Figure 4.22 shows a different type of information. These are instructions to help the user along the course. This way the user always has a suggestion that helps him performing well. For this simulation, the most important in-game instructions are a way to indicate the correct course (Figure 4.22 a)) and the place where the load should be placed (figure 4.22 b)). To achieve the desired outcome, arrow-like elements were added to the simulation.

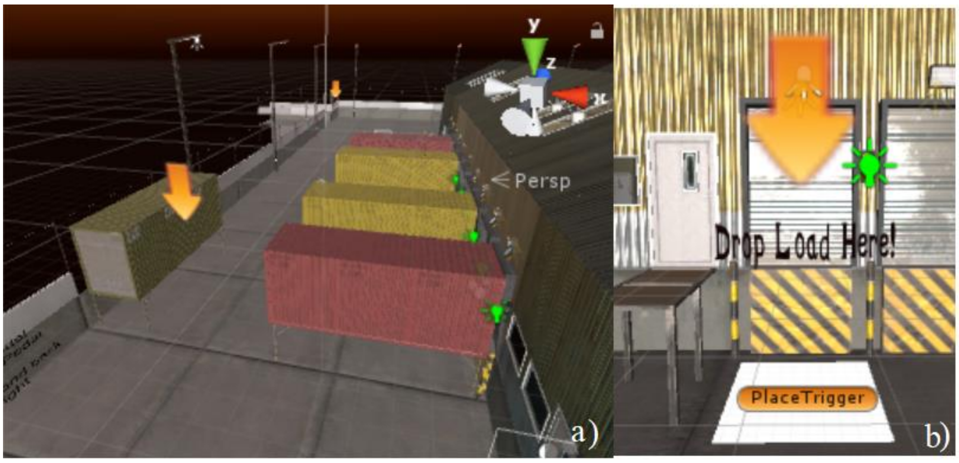


Figure 4.22 On-game information a) Arrows showing the correct path b) Load dropping area

The experience of driving any mobile machine does not feel complete if the user cannot experience some features. The motivation elements play an important role in terms of making the experience much better. Being able to experience a first-person view from the driver's cabin helps to make the simulation better but, for itself, does not add much value in terms of experience. Knowing the forklift's speed is key to help improving the simulation. Due to its configuration, with the rear wheels playing an important role when controlling the direction, users should also be able to know how much they're turning the steering wheel and consequently the wheels, so an element that enables that is needed. Because the user uses a VR set and cannot see what gear is on, a third element should be displayed, a gear viewer. This element indicates what gear in on. It will return a green arrow if the gear is forward, and a red one if the gear in backwards. Figure 4.23 shows the user's screen with the different car elements displayed, so it is possible to understand the visualization idea. There is one last element added, but not visible. It is a pedal sensor, that can measure how much the pedals are being pressed. It is placed under the speedometer and is represented by a horizontal bar. they are only activated when any of the pedals is pressed.

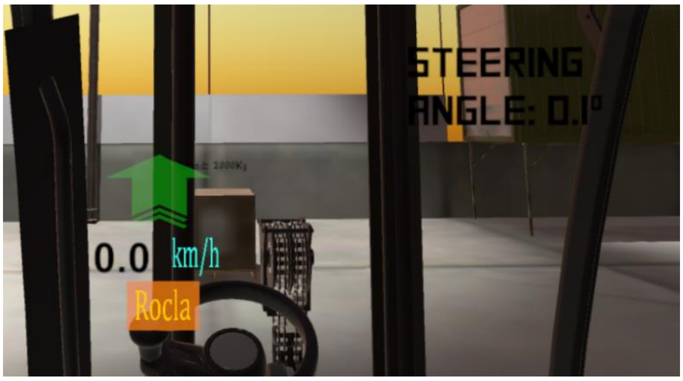


Figure 4.23 Car elements

Regarding the constraints, two main elements are added: a timer and a collision system. The timer only registers the time spent in the simulation, being a normal timer, instead of a countdown timer. It is placed on the bottom right corner of the screen-view and it blends on the simulation, not distracting the user. The collision system is an “invisible” element, created with triggers placed in the same places as the containers and other obstacles. Although the triggers are invisible, the obstacles are well visible, so the user needs to avoid them, like in the real world. Figure 4.24 shows both constraint elements added. Figure 4.24 a) shows the timer, while figure 4.24 b) shows a set of containers with triggers placed in them. Every object that exists in the simulation and is supposed to be avoided has a trigger placed on it.

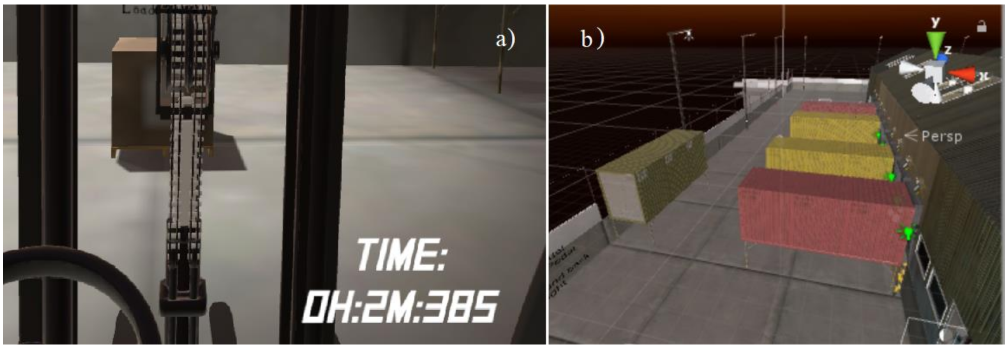


Figure 4.24 Constraint elements a) Timer b) Containers for the collision system

The last element, the scoring system, as it was already mentioned can both reward the user for performing well or punish him for making mistakes. This means that most possible actions in the simulation have a score associated with them. This element, although being associated with the motivational elements, is connected to all the other elements, since it is affected by them. A way to collect point is by going through the correct path and going around the circuit, placing the load in its correct place, or trying to lift the load carrier to its maximum height. On the other hand, collisions are penalized with the loss of points. The scoring system is organized in the following way:

- eight points for each arrow gate the user goes through;
- five points for placing the load in its correct place;
- three points for lifting the load carrier correctly;
- minus one point for each collision.

It is possible to organize the game in a flow chart, for the optimum experience and scoring system. The game process is shown in figure 4.25 with the different points associated to the different actions. Although the way presented in the figure is not the exclusive way to test the simulation, it is the way that enables the best experience.

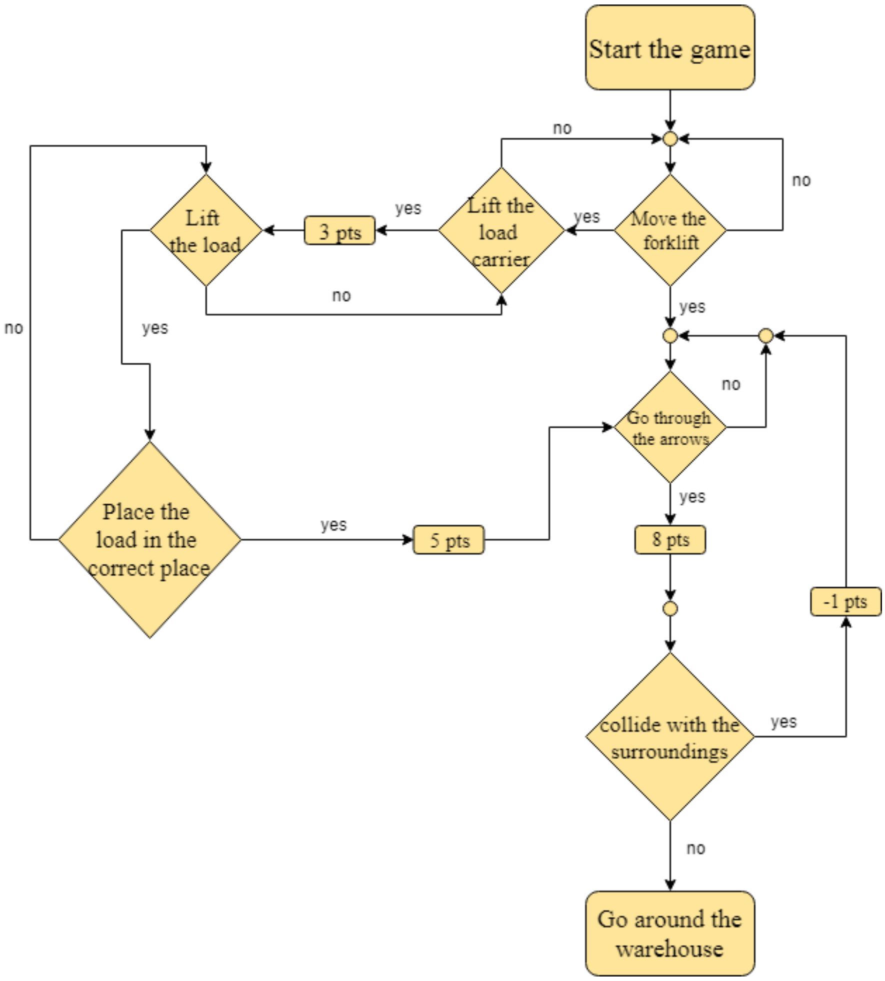


Figure 4.25 Game chart

Input control

The inputs that make the simulation work can be divided into two groups. The first one regards the forklift movements, controlled by Mevea and its physics system. This group englobes all the movements the forklift can do, the collision detection and response, and the stage's movement: height and tilt. Unity only reads these when rendering the simulation. The other group englobes the actions that have an impact on the scoring system and those inputs are controlled by Unity. Figure 4.26 shows how the inputs are organized.

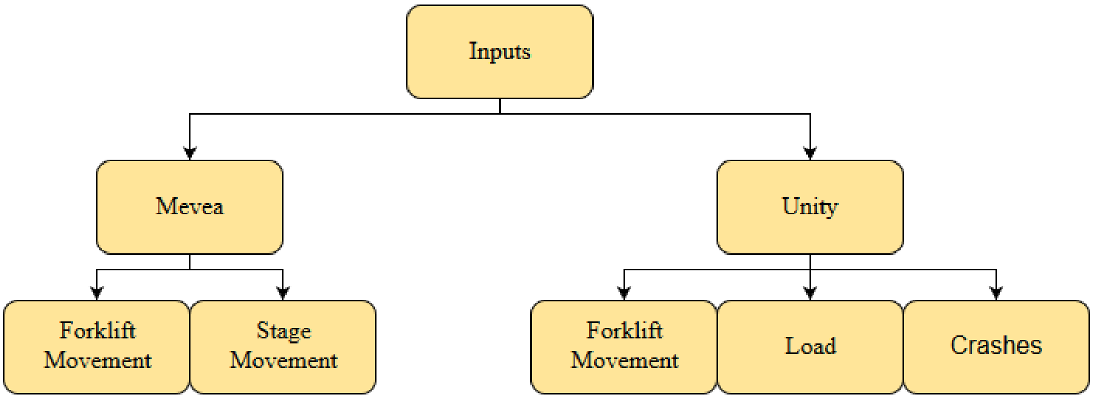


Figure 4.26 Inputs

Mevea allows different inputs to be created, using keyboard keys or some third-party hardware. In this case, the inputs created are given by a steering wheel, pedals, and a joystick, so the control can be similar to a real forklift. The steering wheel and pedals set is a Logitech G29 racing wheel with three standard pedals, while the joystick is a Logitech 3D pro. The first set is responsible for controlling the forklift movement and mimics a real model, with only two gears, front and backwards (due to the electric motor) using only two pedals. The buttons on the steering wheel allow the user to control if the movement is front or backwards (the gear that is on). The joystick's purpose is to control the load carrier and the stage's movement. Movement along the Y axis controls the load carrier's height, while movement along the X axis controls the stage's tilt.

Unity, as a game-engine, allows creating a different set of inputs that derivate from the ones given through Mevea. Unity can read the inputs given through Mevea and reacts, transmitting other different inputs. Starting with the forklift's movement, Unity reads the speed, height and tilt of the forklift and displays them for the user, being updated in real time. When unity reads the input signal from the front/back/neutral movement it enables one of the referred lights to appear on the screen. Regarding the scoring system, Unity reads the position of the forklift in the environment and adds points to the scoring system when some milestones are achieved. The opposite thing happens when the machine crashes with some of the surrounding objects, taking points from the scoring system. Carrying the load is also controlled by Mevea but Unity is responsible for adding points to the scoring system if the activity succeeds and the load is placed in its correct place. Although Unity uses some information from Mevea to control the forklift and to know its position, the scoring system is built using exclusively Unity.

Summing up, Mevea is responsible for all inputs regarding functionality and the physics of the system, while Unity is responsible for all the inputs related to game-elements.

5 *Results*

In this chapter, the results of the work developed are presented. The chapter is divided into two sub-chapters, the first one regarding the results for the dynamic analysis, and the second one regarding the results for the simulation and the introduction of game-elements.

5.1 *Dynamic analysis*

The dynamic analysis was performed by testing the model created with Mevea. Some of the interesting results are the evolution of the load carrier’s position when performing the lifting operation, with and without a load, the velocity of the forklift when not carrying any load and then when carrying a load, and the forklift’s acceleration in the same conditions, that allow to test the efficiency of the analysis and consequently, the accuracy of the model. The hydraulic analysis, as well as any important values for the understanding of the accuracy of the general analysis, are also presented. The stage’s tilt wobbling in different loading situations can also be analysed and compared. For the two limit situations considered (without a load and with a 2000 Kg load), two tests were performed, one for each situation and all the figures presented next regard those two tests.

The first analysis performed was the evolution of the load carrier’s height when not lifting any load and comparing it with the evolution when lifting the 2000 Kg load. The load carrier’s movement is controlled by the joystick and consequently, the joystick movement is considered the same in both situations. Figure 5.1 shows the evolution of the position of the three components responsible for the lifting movement, considering no load. The scales used for the position (vertical axis) vary because of the local coordinate reference used for each body. The time unit chosen (horizontal axis) was seconds. The first stage does not move, with its height changing 0.01 metres appearing as a consequence of the forces applied during the lifting process. The load carrier’s movement starts at the time step 2.39 s and lasts 3.73 s. The second stage mast only starts to move when the load carrier reaches its end (time = 4.58 s). The load carrier’s total change of position is equal to 4.0 meters according to the analysis. It is also possible to notice the velocity change on the load carrier’s position evolution when different hydraulic cylinders are working.

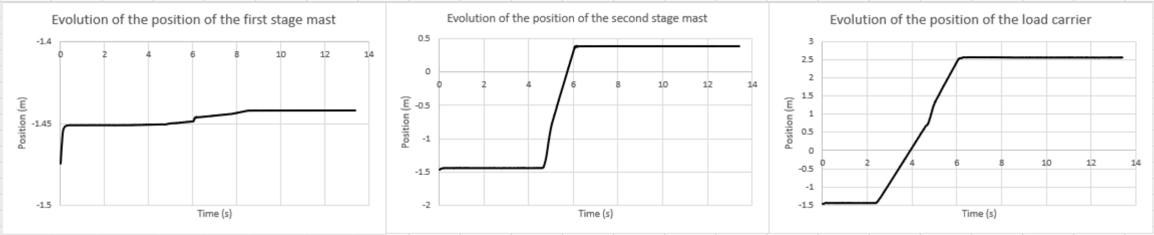


Figure 5.1 Lifting system position analysis considering no load

Figure 5.2 shows the velocity analysis, considering the initial situation. Because the position of the first mast shouldn't change, its velocity should also remain zero. The change of the first stage mast velocity is coincident with the beginning of the lifting process, which is related to the existence of vibration. The second stage mast reaches a maximum velocity of 2.37 m/s when it begins its movement, but the velocity stabilizes at 1.13 m/s. The load carrier´s movement has two different velocities, as suggested in figure 5.1 In its first lifting phase the average velocity is 0.97 m/s, while in the second phase it moves at the same velocity of the second stage mast. The analysis also shows that when the movement stops, the velocity goes to zero, with some oscilation associated.

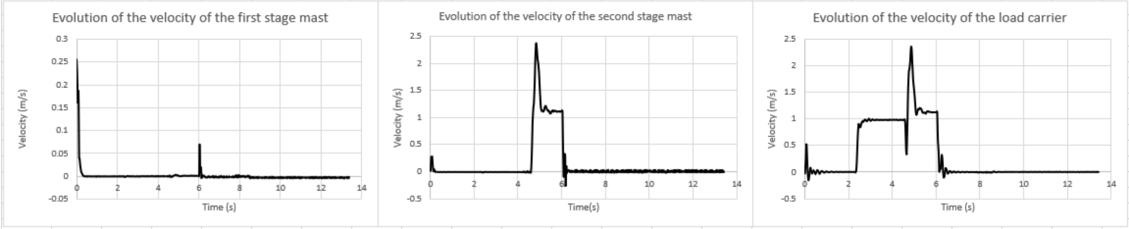


Figure 5.2 Lifting system velocity analysis considering no load

Next, the analysis for the second situation considered is performed. Figure 5.3 shows, one more time, the evolution of the position of the components involved in the lifting process, this time while lifting a 2000 Kg load. Once again, the different scales used are related to the local coordinate reference of each body. The lifting process starts at the time step 11.96 s. The first stage mast's height decreases 0.01 m at the beginning of the process due to the load's weight, but it comes to its reference value, with the same vibration at the end of the movement, as for the previous situation. The load carrier's movement takes a total of 3.94 s, almost the same time as the previous situation. The second stage starts moving two seconds after the lifting movement starts. The maximum height reached is 4 m but, after the load carrier's position stabilizes, this height decreases to 3.988 m.

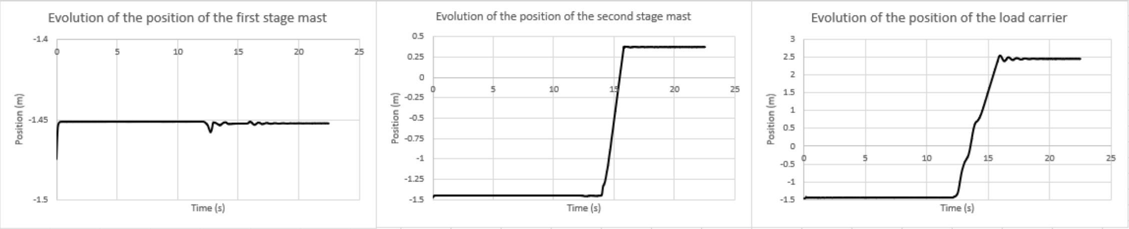


Figure 5.3 Lifting system position analysis for a load = 2000Kg

Figure 5.4 shows the new velocity analysis, for the second situation considered. Because the first stage mast's height changes when the load carrier gets in contact with the load, its velocity analysis also shows a change initially, being this the first visible oscilation. The velocity remains zero after only with a few oscilations. The second stage's velocity evolution is similar to the one from the first situation, without the maximum peak shown in the first analysis. Its average speed is around 1.13 m/s, while the maximum velocity achieved was of 1.3 m/s. Finally, the load carrier's velocity for this situation has a different evolution when comparing it with the previous situation. The first phase of the lifting process does not occur at a constant speed, instead, the velocity has two peaks, respectively, 2.08 m/s and 2.02 m/s. On the second phase lifting, the velocity shows a constant behaviour with a value of 1,12 m/s. At the end of the movement, the velocity tends to zero, with vibration associated, until it stops completely.

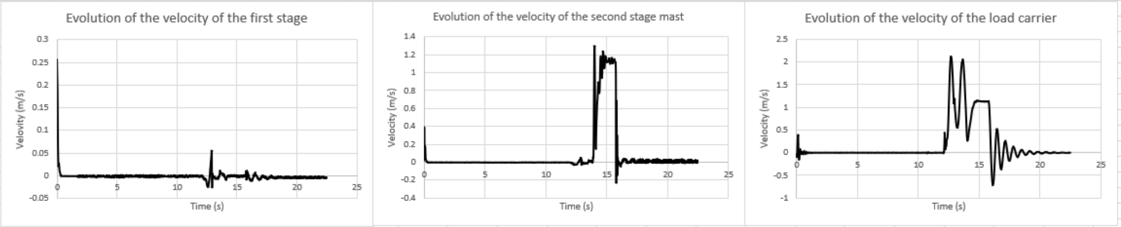


Figure 5.4 Lifting system velocity analysis for a load = 2000Kg

When analysing the velocity achieved while performing the dynamic analysis when carrying and when not carrying the load, it should correspond to the real linear velocity of the forklift in the same conditions. Figure 5.5 shows the model's velocity when not carrying the load, on the left, and when carrying a 2000 Kg load, on the right. The velocity is measured in kilometres per hour (vertical axis), while the time step considered is seconds (horizontal axis). The velocity described was obtained when the accelerator pedal was fully pressed. The reference values are also presented, provided by the maufacturer. The values belonging to the simulation are in a dashed line while the reference values are in a continuous line.

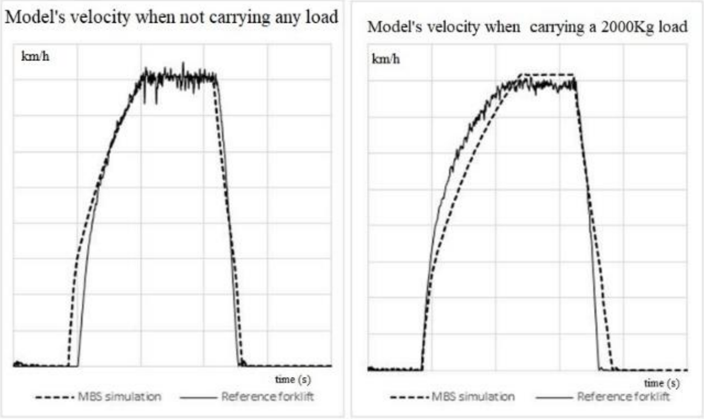


Figure 5.5 Velocity analysis

The same type of analysis can be done for the acceleration. Once again, the model's acceleration should be compared with the real forklift's acceleration in the same situations: when carrying and when not carrying the maximum load. The acceleration analysis is presented in figure 5.6, with the values for when not carrying any load being presented on the left and for when carrying the 2000 Kg load on the right. The acceleration is presented in kilometres per hour square, in the vertical axis and time step chosen is seconds, corresponding to the horizontal axis. As happened for the velocity analysis, the accelerations were obtained when the accelerator pedal was fully pressed. The reference values are also displayed, in a continuous line, while the model's values appear in a dashed line.

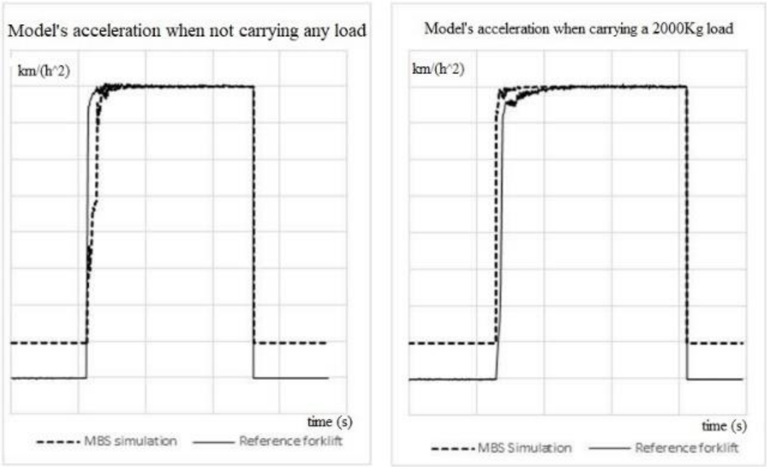


Figure 5.6 Acceleration analysis

Hydraulic Analysis

It is possible to perform a hydraulics analysis by comparing the values from hydraulic components with the parts they have an influence on.

Figure 5.7 shows the behaviour of the two groups of cylinders responsible for the lifting operation. Like the position analysis for the load carrier, this first hydraulic analysis is for the situation where no load is considered. As already mentioned, the pulley cylinder is responsible for the first phase of the lifting, where the load carrier goes from its initial position, on the ground, to the end of the second stage, while the main lift cylinders are responsible for lifting the second stage mast and, consequently, performing the second phase of the load carrier's lifting. The cylinders' stroke is measured in millimetres, while the load carrier position is shown in metres. The maximum stroke of the pulley cylinder is equal to 1087.1 mm, with its initial position being 62 mm and the final 1106.7 mm. By looking to the pulley cylinder's stroke graph, it is possible to see that after its movement is completed (time step 4.58) the stroke has a value of 1076.18 mm, less than the final value. The final change in the stroke value is related to the force transmitted to the second stage mast, which pulls the piston of the pulley cylinder a few millimetres.

The maximum stroke for the main lift cylinders is 1834.5 mm. As expected, the main lift cylinders only start to move after the pulley cylinder reach its final position. The sum of the strokes is equal to 2921.6 mm, with the final height of the load carrier being given by the sum of this value to the second stage's length.

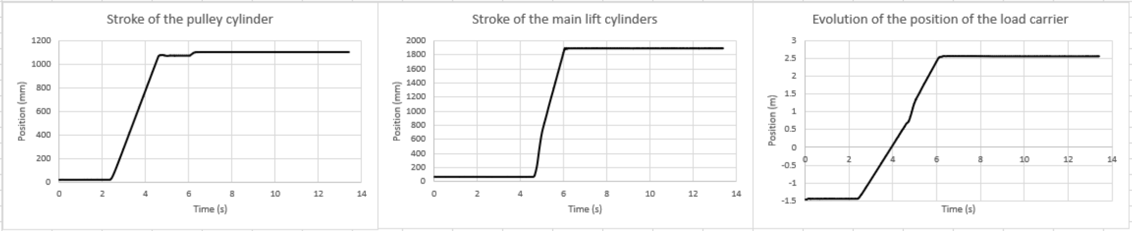


Figure 5.7 Hydraulic lifting components position analysis considering no load

Still considering the same situation (no load), it is possible to analyse and compare the velocity of the cylinders' movements with the load carrier's speed. Figure 5.8 shows the velocity analysis for the hydraulic components. Like the position, the cylinders' velocities are presented in millimetres per second, while the load carrier's velocity is presented in meters per second. The average speed achieved by the pulley cylinder is 486 mm/s, with the maximum peak being almost 600 mm/s at the end of its movement. When the pulley cylinder reaches its final position, the velocity tends to a negative value. The average velocity achieved by the main lift cylinders is around 1120 mm/s with a peak value of 2370 mm/s. From this analysis, it is possible to see that, even though the velocity of the main lift cylinders and the velocity of the load carrier is the same, the same does not happen with the pulley cylinder. The load carrier goes up faster than the cylinder because of the pulley system.

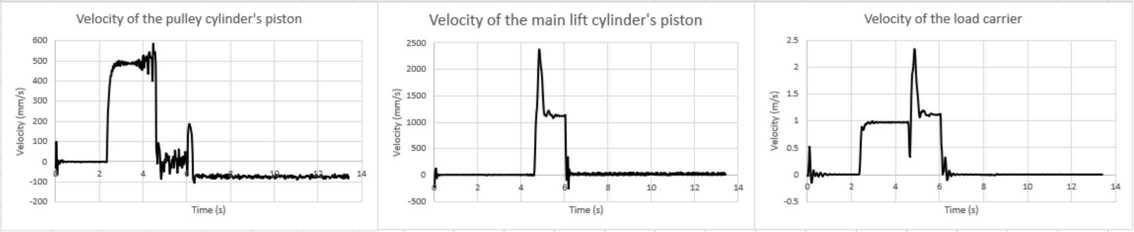


Figure 5.8 Hydraulic lifting components velocity analysis considering no load

The last analysis performed is the one considering the lifting cylinders when lifting the 2000 Kg load. For this second case, the position and velocity analysis are presented.

Figure 5.9 shows the hydraulic cylinders' stroke and the load carrier's position in time, for the lifting process. The cylinder's positions are presented in millimetres, while the time step chosen is seconds. The pulley cylinder's maximum stroke is 1078,7 mm, with its initial position being 20 mm and the final 1098.7 mm.

Once again, by the time the pulley cylinder reaches its final position, the piston's position is lower than the final one, for this case being 1091 mm. The reason for this difference is the same as the one for the first situation. The main lift cylinders have a maximum stroke of 1831 mm, a little less than the stroke for the first situation. The total stroke is equal to 2909.7 mm, which is consistent with the analysis performed to the load carrier and the stages.

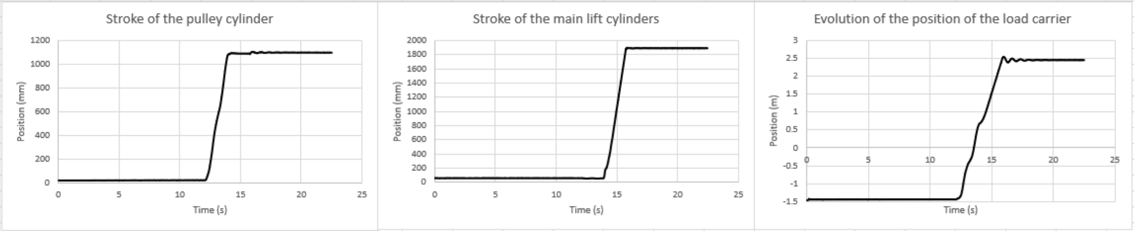


Figure 5.9 Hydraulic lifting components position analysis for a load = 2000 Kg

Figure 5.10 shows the last hydraulic analysis for the second situation considered. Velocities for the cylinders are once again in millimetres per second, and the time step is seconds. The pulley cylinder has a different behaviour this time, not showing a constant velocity evolution, but instead having two peaks and vibration at the end of its movement. These two peaks represent a velocity of 842.2 mm/s. Regarding the main lift cylinders, their behaviour is also different, although still having a peak and a constant velocity zone. The maximum velocity achieved is 1.3 m/s and the average velocity having a value of 1.13 m/s. Putting the cylinders' velocity graphs together it gives a load carrier's velocity graph, with again, the difference of values between the pulley cylinder and the load carrier's ones because of the pulley system.

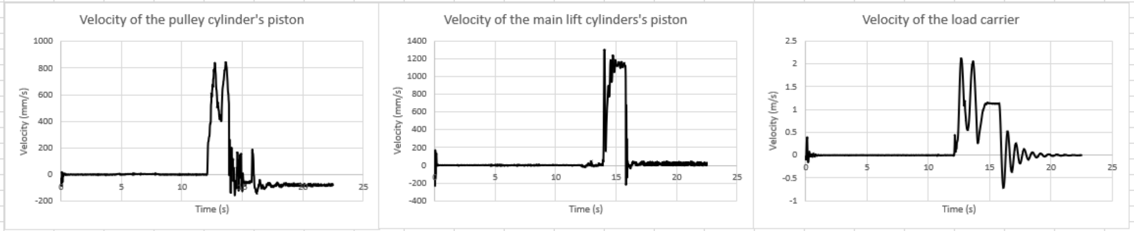


Figure 5.10 Hydraulic lifting components velocity analysis for a load = 2000 Kg

5.2 *Simulation and gamification*

The results obtained for the simulation and the introduction of the game-elements can be structured in the following way: first, the simulation itself is evaluated, with its functionality, accuracy, and efficiency being discussed. After the game elements introduced are also evaluated, with their capability to make the simulation easier, more fun, motivating, or more difficult being analysed. The reason why there are two separate evaluations is that the two topics can be separated, and that way, a more objective discussion can happen.

Simulation

The simulation should occur in real-time. This means that the loop time of the simulation needs to be analysed and compared with the time-step used. For an ideal simulation, the loop time considered should always be lower or equal as the time step considered. If in some very small part of the simulation the loop time happens to become greater than the time step, the simulation is still considered as a good real-time simulation. It is important to avoid the loop time to be bigger than the time-step for long periods of time. This might happen due to any parameter that can affect the numerical integration, such as the meshes used in the objects present in the simulation. Having a loop time greater than the time step means the simulation and the analysis aren’t occurring at real-time and the simulation itself can have experience some problems with the most extreme being crashing.

As mentioned before, the loop time of the simulation is defined as the time needed for the simulation to perform all the numerical integration for the dynamic analysis. For this simulation, the time step used is 1.2 milliseconds. Figure 5.11 shows a graph where is possible to compare the loop time with the time step used for this case. The loop time and the time step are represented by the colour’s orange and back, respectively. It is possible to see that the time step remains constant while the loop time varies between 0.6 and 0.8 milliseconds. The total simulation time considered was about 65 seconds, equivalent to one trip around the warehouse. The loop time varies because it is affected by possible collisions, the position of the load carrier, and all the other elements present in the simulation that, although not being directly influencing the numerical integration, need to be rendered.

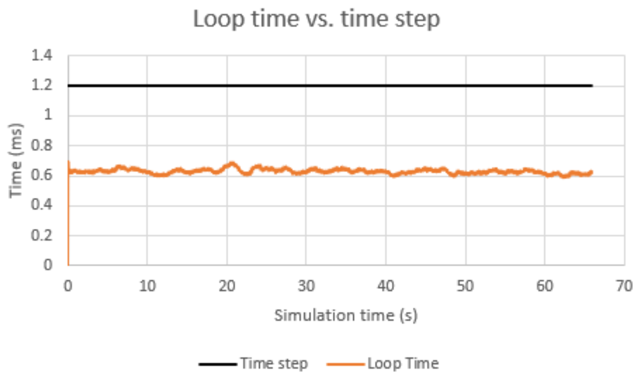


Figure 5.11 Loop time and step time comparison

Gamification

First, the screen view from the user's point of view is presented. Figure 5.12 shows what the user sees when using a first person view. The camera used tries to simulate the cabin view, with all the on-screen elements visible. It is also possible to see a bit of the user hand on the bottom right corner. The scoring system mentioned before is presented on the left of this camera. In front of the forklift, the load with its weight hovering on top is also visible. One of the goals is to take this load to its correct place. All the elements mentioned in the previous chapter are shown together in this view. Displaying the game elements like this helps the user maintaining his focus on what is in front of him, although still being easy enough to read the different values.

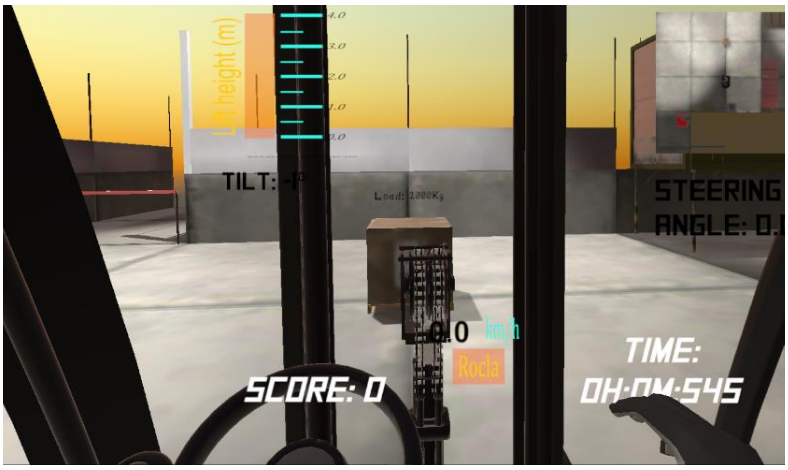


Figure 5.12 Simulation view

The gamification is considered as an important part of the simulation. To evaluate how users are expected to react to the different game-elements, an analysis according to Rosewater principles can be performed.

6 *Discussion*

In this chapter, the results presented in chapter five will be discussed, and the questions presented at the beginning of this works should be answered, so a conclusion can be made. It is important to discuss and validate the results. This chapter is divided into three subchapters: the first two address the dynamic analysis, with the bodies and hydraulic component analysis being discussed, and the last one regarding the simulation, with the real-time accuracy being discussed, as well as the gamification part.

The three questions presented at the beginning of this work are:

- What are the benefits of real-time dynamic analysis?
- What game-elements can be added so that the simulation is effective and motivates users to complete tasks?
- How manufacturers can use the information obtained through game-like simulation?

After this chapter, it should be possible to conclude about the work done.

6.1 *Dynamic analysis*

The dynamic analysis performed can be discussed, with the results being compared to the reference values and/or comparing the two limit situations studied. First, both the velocity and the acceleration of the main body are compared with the reference values. After this first comparison, the position and the velocity analysis for the bodies and hydraulic cylinders is discussed for the two situations considered.

Figure 5.5 shows the velocity analysis, for the two situations considered. Starting with the velocity analysis considering no load, the initial difference refers to the fact that the virtual model starts its movement before the physical prototype. The velocity evolution for the model and the physical prototype is similar, with both achieving the same top speed for the same amount of time. Once the accelerator pedal is released, both velocities drop in an almost linear way and for the same amount of time. For this situation, the model truly represents the physical model. Considering the second case, for a load equal to 2000 kilograms, the physical prototype reaches its top speed faster than the simulated model, but both reach the same maximum velocity. The model shows a more constant speed, considering the small peaks of the reference model’s maximum velocity.

Once again, when the accelerator pedal is released, both velocities drop in a linear way, with the reference model's velocity dropping a little faster than the simulated model's one. For the second case, it can be concluded that the model can represent the physical prototype.

The same discussion can be held for the acceleration analysis. Figure 5.6 shows the acceleration analysis for the same two situations. When the load considered is zero kg, the acceleration for both the model and the reference go up instantly after pressing the accelerator pedal, reaching the same maximum value. When the pedal is released, both accelerations drop at the same rate. The only difference is the starting acceleration values. That difference happens due to the different scaling factors used, in order to use the steering wheel in the simulated model. When considering the second case, there's a small difference between the real acceleration and the model's one. The model's acceleration evolves instantly, while the reference one has a gradient. The model's acceleration also reaches its top acceleration faster than the reference one. This happens due to the heavy load considered. Still, the model can be considered as a trustful representation of the real model in terms of the acceleration.

Because there are no reference values for the lifting mechanisms, the comparison and discussion are done around the two studied situations, without lifting any load (load = 0 kg) and when lifting the 2000 kg load. To validate the model, it is necessary to compare and discuss the position and velocity analysis made for the two situations, comparing the bodies' behaviours with their hydraulic components.

First, the position analysis is discussed. Figures 5.1 and 5.3 show the different components' position evolution for each situation. It is important to notice that the lifting process starts later in the second analysis, since it was necessary to approach the load to lift it. By looking to both analyses, along with the information available, it is possible to notice that the lifting time for the two situations is different. The total time needed for the first was 3.73 s, while the one needed for the second was 3.94 s. The first phase lift occurs faster for the heavier load, but the second phase lift is faster for the first situation. It is also possible to see different behaviours in the three components analysed. The first stage mast, when lifting zero Kg changes its position 0.01 m due to the cylinders' forces, while when lifting the 2000 Kg load, it changes it is the position only in the beginning of the lifting process when the contact with the load happens, remaining normal during the rest of the process. The other main difference happens with the load carrier: in the first situation, it shows a more linear behaviour, with a difference in the graph's gradients being noticeable from the first to the second lifting phases, while in the second situation this component shows less difference between different phases with more oscillation at the end of the movement, affecting the lift's final height. The final height for the first situation is four meters, and 3.988 meters for the second (after the oscillation stops).

With the position analysis done, it is possible to discuss the velocity analysis. The result of this analysis should support the data obtained from the position analysis. Figures 5.2 and 5.4 show the velocity analysis for the same two cases. For both situations, the first stage mast shows the same change, a vibration at the beginning of the lifting process. The second stage mast behaves almost the same way for both situations with a smaller difference. The maximum velocity achieved (the peak) is higher for the first situation, which is correct, for having to lift no load. The biggest difference occurs for the load carrier. For the first situation, the load carrier velocity analysis shows two different, but constant velocities, corresponding to the two different lifting phases and the difference is clear, while in the second situation, for the first phase lift, the speed is not constant, but it presents two peaks. This graphic supports the position analysis for the same situation, with a less linear pattern visible in the load carrier's position. Finally, the load carrier shows a bigger vibration at the end of the second process, due to the instability caused by the heavier load.

6.2 *Hydraulic analysis*

As done for the three responsible bodies for the lifting process, a position and velocity analysis were conducted for the hydraulic components involved in this process. This analysis should be in accordance with the one previously discussed, to prove the correct functioning and interaction of this subsystem.

Starting by the position analysis when lifting zero and 2000 kg, shown in figures 5.7 and 5.9. For both situations, all the cylinders show a similar behaviour. The main difference happens when lifting the heavy load, where the pulley cylinder shows a less linear stroke, which is shown in the load carrier's position. Once again, this is probable to happen due to instability. The difference between the total stroke for both situations (total stroke for the first situation = 2921.6 mm; total stroke for the second situation = 2909.7 mm) explains the difference between the different heights achieved by the load carrier, although being inexpressive (a difference of 12 mm for a load of two tons).

The final dynamic analysis in discussion is the velocity analysis of the hydraulic components. This analysis, for the two situations, is displayed by figures 5.8 and 5.10. As mentioned for the position analysis, the cylinders show similar behaviour. All of them, for both situations, show an almost instant gain of velocity, due to the hydraulic pressure increasing. For both situations, the pulley cylinder shows the same behaviour at the end of its movement, with the velocity tending to a negative value. This happens as a reaction to the force imposed by the main lift cylinders, to stop the pulley cylinder's movement.

The main lift cylinders have similar behaviour as the second stage mast, as expected. It is possible to see by analysing the graphs, that the load carrier's velocity for both situations is the sum of the cylinders' velocities graphs. The fact that, for both situations, the pulley cylinder's average velocity is half of the total velocity of the load carrier's average velocity shows that the pulley system is responsible for this half of the lift, with the pulley cylinder being only responsible for the stabilization at the end of the movement.

By studying all the results for both analyses, for the bodies and the hydraulic components, it is possible to see that all the bodies act according to the hydraulic components, proving the good functioning of this subsystem.

Before starting to answer the questions presented at the beginning of this work, some notes about the model created and the simulation should be taken into consideration. Although the model works well and accurately represents the real model, it is a simplified representation. The electrical part is not modelled and is not taken into consideration, only the power transmitted by the motors is included. Most of the real parts and components of the real forklift are merged in the model's representation, in order to simplify the numerical integration. Finally, the real hydraulic system is much more complex, but once again, it is simplified in the model for simulation purposes. The model's behaviour respects all the parameters given by the manufacturer, but many other parameters were not made available, making the model simpler to simplify.

Now it is possible to answer the first question. Real-time dynamic analysis allows to test products in different conditions, with the possibility of changing the parameters in analysis, and the system reacting to these changes. For the forklift studied, the analysis performed considered two different situations. The results proved to be equal to the reference values, with the convenience of not needing to use a real forklift to perform the analysis. So, the major benefits of using real-time dynamic analysis involve the possibility of testing different situations with different loads, the possibility of testing in virtual prototypes, and the possibility to study multiple complex systems, with less costs involved, more accuracy and efficiency, and less time needed.

6.3 Simulation and Gamification

The results about the simulation's capability to react to inputs in real time, and the accuracy of the model in terms of representing the real forklift, as well as the gamified simulation experience should be evaluated, having in account the different types of users and the features information presented by Rosewater.

Simulation

The first discussion should be about the loop time and the real time of the simulation. The loop time presented in figure 5.11 in orange shows its values floating between the values of 0.6 and 0.8 milliseconds. This means the time necessary to perform any numerical integration changes during the simulation, possibly affecting the results. The correct way to maintain the loop time below the time step is not to choose a bigger time step (that would bring various problems in terms of the dynamic analysis and the definitions chosen), but to try to work the simulation around to keep the loop time low. Because the loop time is affected by the meshes used, proximity between objects, collisions, and simulation rendering, these aspects need to be taken in special consideration. The meshes used in the model, especially in the areas where collisions can happen, can't be simplified, but the ones belonging to objects in the environment, that have no impact in the course of the simulation (the ones used for the visual effect only) can have their meshes simplified, to make the numerical integration faster. Occlusion features in the simulation can also be used so that the parts that are not visible in every moment do not need to be rendered at the same time as the ones visible are. This way it is possible to achieve a loop time lower than the time step and make sure the model's reaction happen without any delay when comparing its actions to the real prototype. With the simulation's loop time being lower than the time step, the simulation can be efficient and accurate.

Regarding the model and its ability to represent the real prototype, there are two main factors that affect this parameter: the graphics used and the way the model reacts to collisions. The graphics are easy to import from the manufacturer, so the model should look like the real prototype. In terms of collision reaction, this factor is affected by the collision graphics used and the complexity of the meshes used. The model uses simple collision graphics, so that the numerical integration can occur without any problems, but the meshes used are complex enough to create a simplified, but accurate collision model. The model behaves in a simpler way when colliding, but accurate enough when comparing it to the real prototype. Its functionality can be validated this way, proving the be an overall good simulation.

For this thesis, user testing was not considered due to calendar problems. A group of users testing the simulation could give a better evaluation about the simulation and its positive and negative aspects. The only tests performed, for this thesis purpose, were made by the author. In a later stage, a representation group, with both experienced and unexperienced users tested the simulation.

Gamification

The gamified simulation can be evaluated by analysing the type of users that can feel motivated to try it, according to the table 3.1 (types of users) and after, by comparing the gamified simulation features with the features Rosewater presents

On a first note, the gamified simulation lacks a first interview to help understand what needs the users feel more important to be met and what type of users the simulation should aim for. A first phase gamified simulation should have been presented and compared to the final one. Nevertheless, the introduction of game elements to turn the simulation into a more immersive environment tries to meet most of the more general user needs, with that being visible in the results obtained.

The first part of this discussion, as mentioned, is to see what type of users can feel motivated to try the simulation. The types of users were philanthropist, socializer, free spirit, achiever, player, and disruptor. The simulation has a purpose and some tasks to be fulfilled, being appealing for a philanthropist user. Being a “single player” simulation, it might not motivate the socialiser user or the player one, since it lacks a socialization element. In other words, it lacks a community. The free spirit can find this simulation appealing, since it allows some kind of exploration of the environment and free movement. The existence of motivation elements as well as constraints can motivate the user defined as the achiever. The last type of user, the disruptor, also might not feel motivated to perform, since the simulation does not allow the chance to change the environment or the possibility of developing it.

It is possible to conclude that from the six different types of users, only three might feel motivated to test the simulation. This happens because the simulation has a specific purpose, instead of being a “normal” game. The simulation is targeted for a specific type of users.

Regarding the comparison of the features introduced by Rosewater and the ones the simulation has, it is good to remember what features were presented in the chapter 3, according to Rosewater. Table 6.1 resumes this information.

Table 6.1 Rosewater´s features

Feature	Description
Goal	The game needs a purpose, a point. Needs to be clear
Rules	Set of instructions the user needs to follow
Interaction	Interaction between the user and the simulation
Catch-up	Possibility for a user that gets lefts behind to catch-up in score
Inertia	The game should have an end
Surprise	Moments that the user is not expecting
Element of strategy	Possibility to learn and improve with time
Fun	The user needs to enjoy himself
Flavour	The game should be associated to a story or a theme, it should not be bland
Hook	Something that makes people want to try the game

This simulation has a main goal, which is to drive the forklift around the warehouse while putting the load in its correct place. It also has a set of rules, like limits where the forklift can go and specific ways to get points. The interaction between the simulation and the user can be achieved with the scoring system, that acknowledges the good action and punishes the bad ones. The scoring system also allows any user that starts poorly to catch up with the other users.

The existence of a timer is another element that allows the user to end up with a better ranking. These two are the catch-up elements. For most users, the forklift´s behaviour itself can be considered as a surprise element. Its behaviour can be unpredictable so the user should be careful not to be too surprised. The more times the user tries the simulation, the more he should learn about the machine´s behaviour, making it possible to find a way to maximize the score obtained, while minimizing the time needed. The possibility of using a VR set or the presence of the numerous game elements already mentioned can help the user to have fun, and to learn about the forklift's details. The final hook to make users wanting to try the simulation is the use of a simulator that tries to mimic the real controls, as well as the chance of having a VR experience. The two negative features are the lack of an end and this simulation not having any story associated to it. Regarding the first negative feature, the simulation can be endless. Although the scoring system has a pre-defined number of ways to get or to loosen points, the timer can go for as long as the user wants, so the inertia feature does not exist. Because this simulation has the goal of being used to perform tests and/or help training drivers, it restricts the ability of having a story or theme associated with.

By comparing the existing features with the Rosewater’s ones, in ten features a game should have, this gamified simulation has eight, proving to be a good simulation in terms of the gamification part.

Finally, it is possible to answer the second question presented regarding the game elements, effectiveness, and motivation. The elements chosen for this gamified simulation proved to be the most adequate ones. Although this simulation might not appeal to every type of user and not having all the features a game should have, for the purposes already mentioned, with the game elements added as well as the tasks created, not many more game elements need to be there. The most important ones are those which provide information to the driver. For this gamified simulation to be appealing to every type of user a story or a theme should be associated to the simulation and the simulation would need to have an end at some point. Regarding the game elements, more could be added, like fuel consumption, hydraulic behaviour, or a more complex scoring system.

6.4 *Other considerations*

Regarding the final question, the easiness to obtain good information is making manufactures and engineers turn to this type of testing. The quickness of creating a model that accurately represents the physical model with the possibility of creating immersive environments, all of this with fewer costs involved when compared to creating several physical prototypes, makes real-time simulation a good tool to improve and/or create different products. By using the data from the game-like simulation about the model, user opinions, and in some cases user feelings while experiencing products, manufacturers can understand what type of features make users happier or more thrilled, uninterested or less likely to enjoy products, making it easier to adapt products to costumer’s concerns. A user/costumer that relates to a product is more likely to buy it, making profits greater and improving products faster.

The work developed for this thesis was made according to some conditions and restrictions because the simulation was to be used for external purposes. Some simplifications were imposed, and some game elements were ruled out because they would not add nay value to the simulation. The simulation file produced along with this work is not available for demonstration purposes because of the conditions imposed.

For a future work with this model, some considerations should be taken into consideration. Mevea allows to build a model, but the data regarding the body’s characteristics need to be known first, as well as the graphics to use. The software does not enable the creation of new parts without graphics or without knowing the mechanical and physical properties.

To develop this work, two directions can be taken: the improvement of the model, or the improvement of the gamified simulation. For the first case, a more detailed model can be created, in order to be studied with more detail. For the second case, a better environment can be created, with more tasks and more freedom to “play” the game. For this second case, special attention is needed, to keep the simulation running as a real-time one, without any delay.

7 Conclusion

In this chapter, the final considerations are made, A summary of the work developed is also presented.

The main objective of this work was to perform a real-time dynamic analysis to a forklift with a simulation build consequently. The model created was expected to mimic the real forklift's behaviour velocity-like, acceleration-like, and with the simplified hydraulic circuit working like the reference model. For the simulation, game elements were added to give a more realistic feeling to the users, for an efficient data collection. The feedback received allows to make some changes to the model, making it better and with the possibility to improve the real forklift. This way, the time needed to test the forklift decreases, costing less money for the manufacturer, and reducing the need to create multiple physical prototypes.

Three main questions were proposed, focusing on the benefits of real-time dynamic analysis, the game elements that can be used in a gamified simulation to make it more effective and fun to use, improving user's motivation, and finally how the information obtained can be used. To answer all these questions, the model needed to perform well, and the simulation needed to accurately represent an usual case scenario for a forklift. The controls and inputs of the forklift tried to be closest possible to the real ones and the simulation tried to cover most of the tasks a forklift should do.

The model's pedal's sensitivity, the fork's behaviour, steering system, and other parameters react the same way as the real machine, proving to be an accurate tool to represent a physical prototype. This was possible to achieve by following the specifications given by the manufacturer.

The game elements added proved to be stimulating enough for the users, helping them to easily understand the functioning of the forklift. After trying the simulation a few times, users start to understand how game elements can help them understanding the forklift's behaviour and improve their performance.

It is possible to conclude that the system works and can mimic the real forklift's behaviour, proving to be an accurate representation for the pretended purpose. The introduction of the game elements proved to be a differentiation element.

Using software tools like Mevea and Unity to build virtual simulation models of machines can be explored in the close future as tools to improve the design and test phases in engineering projects.

Bibliographic References

Alben, L. (1996). Defining the Criteria for Effective Interaction Design. In L. Alben, *Interactions* (pp. 11-15).

AutoDesk. (2016). *Help*. Retrieved from AUTODESK : http://help.autodesk.com/view/SCDSE/2016/ENU/?guid=GUID-637724EB-02CF-4195-AF55-5F9CDD08F332

Baharudin, M. (2016). *Real Time Simulation of Multibody Systems with Applications for Working Mobile Vehicles Based on Multibody Modeling*. Lappeenranta: Acta Universitatis Lappeenrantaesis 690 (ISBN 978-952-265-932-3).

Baharudin, M., Rouvinen, A., Korkealaasko, P., & Mikkola, A. (2014). Real-time Multibody Application for Tree Harvester Truck Simulator. *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, pp. 182-198.

Bélanger, J., Venne, P., & Paquin, J. -N. (2010). The What, Where and Why of Real-Time Simulation. *IEEE Power & Energy Society*, pp. 37-49.

Cellier, F. E., & Kofman, E. (2006). *Continuous System Simulation*. New York: Springer Science & Business Media.

Crespo Marquez, A. (2010). *Dynamic Modelling for Supply Chain Management*. Seville: Springer London.

de Jalón, J. G., & Bayo, E. (2009). *Kinematic and Dynamic Simulation of Multibody Systems, The Real-Time Challenge*. San Sebastian: Donostia.

Desmet, P., & Hekkert, P. (2007). Framework of Product Experience. *International Journal of Design*, pp. 57-66.

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From Game Design Elements to Gamefulness: Defining Gamification. *Proceedings of the 15th International Academic MindTrek Conference* (pp. 9-15). Tampere: MindTrek.

Ellwanger, C., Pierre da Silva, R., & Antunes da Rocha, R. (2018). Systemic Modelling and Simulation: Management Strategy in the User Experience Design Process (UXD). *Gestão & Produção*, pp. 659-670.

Flores, P. (2015). *Concepts and Formulations for Spatial Multibody Dynamics*. Guimarães: Springer International Publishing. doi:10.1007/978-3-319-16190-7

Haug, E. J. (1993). *Concurrent Engineering: Tools and Technologies for Mechanical System Design.* Verlag: Springer-Verlag Berlin and Heidelberg GmbH & Co. KG.

Interaction Design Foundation. (2002). *User Experience (UX) Design*. Retrieved from Interaction Design Foundation: https://www.interaction-design.org/literature/topics/ux-design

Jaiswal, S., Islam, M. I., Hannola, L., Sopanen, J., & Mikkola, A. (2018). Gamification Procedure Based on Real-time Multibody Simulation. *International Review on Modelling and Simulations*, pp. 259-266.

Knutas, A., van Roy, R., Hynninen, Granato, M., Kasurinen, J., & Ikonen, J. (2018). A Process for Designing Algorithm-based Personalized Gamification. *Multimedia Tools and Applications*, pp. 1-20.

Kruger, W. R., Vaculin, O., & Kortum, W. (2002). Multi-disciplinary Simulation of Vehicle System Dynamics. *Reduction of Military Vehicle Acquisition Time and Cost through Advanced Modelling and Virtual Simulation* (pp. 16.1-16.16). Wessling: DLR - German Aerospace Center.

Larson, T. C., & Papalambros, P. (2003). Linking Analytical Target Cascading to Engineering Information Systems for Simulation-based Optimal Vehicle Design. *Proceedings of the International Conference on Engineering Design, ICED.* Stockholm: Luleå University of Technology.

Larsson, T. (2001, February). *Multibody dinamic Simulation in Product Developement*.

Li, B., Yao, X., Wang, C., & Lou, H. (2010). Dynamic Multobody Simulation of a 6-DOF Robotic Arm. *Advances Materials Research Vols. 139-141*, pp. 1001-1004.

Mikkola, A. (2018). Lecture1_m. *BK70A0001_03.09.2018 Simulation of a Mechatronic Machine* (pp. 44-53). Lappeenranta: Lappeenranta University of Technology.

Mikkola, A. (2018). Week 10-2018mool. *BK70A0001_03.09.2018 Simulation of a Mechatronic Machine* (pp. 1-41). Lappeenranta: Lappeenranta University of Technology.

Mikkola, A. (2018). Week 10-2018mool. *BK70A0001_03.09.2018 Simulation of a Mechatronic Machine* (pp. 34-36). Lappeenranta: Lappeenranta University of Technology.

Mikkola, A. (2018). Week 8-18mo. *BK70A0001_03.09.2018 Simulation of a Mechatronic Machine* (p. 2). Lappeenranta: Lappeenranta University of Technology.

Mikkola, A. (2018). Week 9-18moo. *BK70A0001_03.09.2018 Simulation of a Mechatronic Machine* (pp. 1-32). Lappeenranta: Lappeenranta University of Technology.

Mikkola, A. (2018). Week11-2018mood. *BK70A0001_03.09.2018 Simulation of a Mechatronic Machine* (p. 11). Lappeenranta: Lappeenranta University of Technology.

Mikkola, A. (2018). week2-18mm. *BK70A0001_03.09.2018 Simulation of a Mechatronic Machine* (p. 26). Lappeenranta: Lappeenranta University of Technology.

Mitsubishi. (2019). *Mitsubishi FB14-20A(C)NT Series*. Retrieved from Mitsubishi Forklift Trucks: https://www.mitforklift.com/en/eu/edia-em-fb14-20a(c)nt-series-electric-3-wheel-forklifts

Multi Body Dynamics (MBD). (2013). Retrieved from Simgrosys: http://www.simgrosys.com/multi-body-dynamics.html

Nelson, M. J. (2012). Soviet and American Precursors to the Gamification of Work. *Proceedings of the 16th International Academic MindTrek Conference 2012: "Envisioning Future Media Environments"* (pp. 23-26). Tampere: MindTrek.

Nikravesh, P. E. (1988). *Computer-Aided Analysis of Mechanical Systems.* Englewood Cliffs, New Jersey: Prentice Hall inc.

Nikravesh, P. E. (2008). *Planar Multibody Dynamics: Formulation, Programming and Applications.* Boca Raton: CRC Press.

Pirker, J., & Guti, C. (2015). Educational Gamified Science Simulations. *Gamification in Education and Business*, pp. 253-275.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (1992). *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge: Cambridge University Press.

Remi-Omosowon, A., Cant, R., & Langensiepen, C. (2016). Applying Gamification Principles to a Container Loading System in a Warehouse Environment. *Proceedings - 2016 UKSim-AMSS 18th International Conference on Computer Modelling and Simulation, UKSim 2016* (pp. 79-84). Cambridge: IEEE.

Rosewater, M. (2011). *Ten Things Every Game Needs, Part1 & Part2*. Retrieved from Magic the Gathering: https://magic.wizards.com/en/articles/archive/making-magic/ten-things-every-game-needs-part-1-part-2-2011-12-19

Roupa, I., Gonçalves, S. B., & Tavares da Silva, M. (2018). Dynamic Analysis of Planar Multibody Systems with Fully Cartesian Coordinates. *The 5th Joint International Conference on Multibody System Dynamics.* Lisbon: Instituto Superior Técnico.

Schiehlen, W. (2006). Computational Dynamics: Theory and Applications of Multibody Systems. *European Journal of Mechanics*, pp. 566-594.

Shabana, A. A. (2001). *Computational Dynamics*. Chicago: John Wiley & Sons, Inc.

Slaats, P. M. (1991). *Recursive Formulations in Multibody Dynamics.* Eindhoven: Technische Universiteit.

Smith, I. F. (2012). Integrating the Science of Computing Undergraduate Engineering Curricula. *ASCE Structures Conference 2012*, (pp. 1-5). Chicago.

Sol, E. J. (1983). *Kinematics and Dynamics of Multibody Systems : A Systematic Approach to Systems With Arbitrary Connections.* Eindhoven: Technische Hogeschool Eindhoven. doi:10.6100/IR82221

US Army. (2013). *Systems Engineering Fundamentals.* Fort Belvior, Virginia: CreateSpace Independent Publishing Platform.

Vagabond, J. (2012). *Turbulent Flow and Viscosity*. Retrieved from Physics and Chemistry for IG and A Level: https://esfsciencenew.wordpress.com/2012/10/26/turbulent-flow-and-viscosity/

Van de Boer, P. (2013). *Introduction to Gamification*.

Appendix

Appendix 1: Bodies positions, masses, centre of mass and moment of inertia

Table 0.1 Main body

Mass (Kg)	**1859.85**		
Position (x,y,z) (m)	-5	-1.35	0
Center of mass (x,y,z) (m)	0.99	0.30996	-0.01
Moment of Inertia	410.65	-86.32	-4.02
		778.89	0.77
			806.52

Table 0.2 First Mast

Mass (Kg)	**245**		
Position (x,y,z) (m)	-0.125	-0.125	0
Center of mass (x,y,z) (m)	-0.01232	1.22	0
Moment of Inertia	142.54	0	0
		14.23	0
			133.06

Table 0.3 Second mast

Mass (Kg)	**190**		
Position (x,y,z) (m)	0	0	0
Center of mass (x,y,z) (m)	-0.05	1.18	-0.01
Moment of Inertia	108.21	0	0
		6.22	0
			103.92

Table 0.4 Load carrier

Mass (Kg)	**165**		
Position (x,y,z) (m)	-0.0625		
Center of mass (x,y,z) (m)	-0.3586	0.09	0
Moment of Inertia	16.62	0	0
		37.63	0
			37.63

Table 0.5 Pulley holder

Mass (Kg)	**20**		
Position (x,y,z) (m)	-0.056	1.36	0.0025
Center of mass (x,y,z) (m)	0	-0.07	0
Moment of Inertia	1	0	0
		2	0
			5

Table 0.6 Rear axle

Mass (Kg)	**89**		
Position (x,y,z) (m)	1.422	-0.01	0
Center of mass (x,y,z) (m)	-0.01	0.08	0
Moment of Inertia	1.73	0	0
		1.42	0
			1.55

Table 0.7 Lifting load

Mass (Kg)	**2000**		
Position (x,y,z) (m)	-10	-1.525	0.5
Center of mass (x,y,z) (m)	0	0.5	-0.5
Moment of Inertia	334.6	0	0
		335.3	0
			334.6