LAPPEENRANTA UNIVERSITY OF TECHNOLOGY
LUT School of Energy Systems
LUT Mechanical Engineering

Stanislav Ustinov

DEVELOPMENT OF A NOVEL REAL-TIME MODEL OF HUMAN SKELETON IN MEVEA PLATFORM

Examiners: Professor Heikki Handroos
Dr. Sc. Tech. Eng. Hamid Roozbahani

Master's Thesis 2018

ABSTRACT

Lappeenranta University of Technology
LUT School of Energy Systems
LUT Mechanical Engineering

Stanislav Ustinov

Development of a Novel Real-Time Model of Human Skeleton in Mevea Platform

Master's Thesis

2018

84 pages, 43 figures, 10 tables, and 4 appendices

Examiners: Professor Heikki Handroos
Dr. Sc. Tech. Hamid Roozbahani

Keywords: multibody systems, real-time simulation, human skeleton, biomechanics, motion traction, Mevea software.

The aim of the project was to develop a real-time simulation of the human skeleton model for the clinical application and to describe real-time changes of the torque, while the person was moving in front of motion capture camera. The hypothesis of the project is based on the fact that there are few real-time skeleton models, which are used in physiotherapy rehabilitation, and Mevea solution is novel for this purpose.

The project was made using various software such as Mevea simulation software, Matlab Simulink, 3ds Max, and Windows SDK 2.0. Kinect for Windows v2.0 sensor was used as system hardware. Firstly, Mevea multibody model of the human skeleton was created. Then, Simulink external interface for Mevea was made. Secondly, the connection between the Kinect and the Simulink was established. After that, in order to obtain the results of the project, the developed model has been tested. The experimental section was divided into three separate case studies that were a test of the elbow joint, thoracic joint, and full body.

Results of the project are described via graphs which show changes in torque and angular position of the elbow and thoracic joints, according to the input position of joints. Case study 3 provides the results of the full body test, which was made with Kinect for Windows device. The results of this case study are demonstrated via connection of the system and obtaining of torque values from Mevea software.

ACKNOWLEDGEMENTS

I want to express my greatest gratitude to all the people who somehow helped me with writing this work, inspired me and helped with advice.

First of all, thanks to my supervisors Professor Heikki Handroos and Dr. Sc. Tech. Eng. Hamid Roozbahani for their useful advice. Without your advice, it would be challenging to achieve such results. I also want to thank Juha Koivisto for providing all the necessary equipment and helping in setting it up.

Then, I would like to express special thanks for Teemu Priha and Ilya Kurinov for helping with Mevea software. Moreover, last but not least, thanks to my family and friends who inspired me during this challenging research project.

Stanislav Ustinov

Stanislav Ustinov

19.10.2018, Lappeenranta

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

Latin symbols

$\mathbf{0_3^T}$	Null vector
A	Scale coefficient of controller input
a_e	Depth of ellipsoid [m]
a_{i-1}	Link length [m]
a_p	Depth of rectangular prism [m]
$\mathbf{A^i}$	General rotation matrix
$\mathbf{A_x^i}$	Rotation matrix about an *x*-axis
$\mathbf{A_y^i}$	Rotation matrix about a *y*-axis
$\mathbf{A_z^i}$	Rotation matrix about a *z*-axis
$\mathbf{A^T}$	3x3 transformation matrix
B	Offset of controller input
b_e	Height of ellipsoid [m]
b_p	Height of rectangular prism [m]
c_e	Width of ellipsoid [m]
c_p	Width of rectangular prism [m]
$\mathbf{D_1}, \mathbf{D_2}, \mathbf{D_3}$	Rotation matrices of Euler angles
d_i	Link offset [m]
d_n	Damping constant [Nms/rad]
D_z	Dead zone of the input
F	Muscle force [N]
$\mathbf{I}$	Moment of inertia tensor
$\mathbf{i_1}, \mathbf{i_2}, \mathbf{i_3}$	Unit vectors along axes X_1, X_2, X_3 of a global frame of reference $X_1X_2X_3$
I_x, I_y, I_z	Principal mass moments of inertia [kgm^2]
$\mathbf{i_1^i}, \mathbf{i_2^i}, \mathbf{i_3^i}$	Unit vectors along axes X_1^i, X_2^i, X_3^i of a body frame of reference $X_1^iX_2^iX_3^i$
K_n	Spring constant [Nm/rad]
l_c	Length of the cylinder [m]
m	Mass [kg]

m_b	Mass of bone [kg]
n_b	Number of bodies
n_c	Number of constraint equations
P_{xy}, P_{xz}	Mass products of inertia (x) [kgm^2]
$P_{yx},\ P_{yz}$	Mass products of inertia (y) [kgm^2]
$P_{zx},\ P_{zy}$	Mass products of inertia (z) [kgm^2]
r_c	Radius of the cylinder [m]
r_F	Moment arm [m]
r_s	Radius of the sphere [m]
$\mathbf{R^i}$	Global displacement vector of body reference
$\mathbf{r^i}$	Position vector of the point $\mathbf{P^i}$ of body i with respect to a global frame of reference
$\mathbf{r_4^i}$	4x1 position vector of the point $\mathbf{P^i}$ of body i with respect to a global frame of reference
T	Torque [Nm]
T_a	Total torque produced by rotational spring-damper actuator [Nm]
T_{fn}	Friction torque [Nm]
T_n	Constant torque applied to the actuator [Nm]
$\mathbf{T_1}, \mathbf{T_2}, \mathbf{T_3}, \mathbf{T_4}$	Transformation matrices of Denavit-Hartenberg parameters
$\mathbf{T_4^i}$	4x4 homogeneous transformation matrix
$\mathbf{T^{i,i-1}}$	4x4 Denavit-Hartenberg transformation matrix
$\mathbf{u^i}$	Vector from a global frame of reference to point $P^i$of the body i
$\bar{\mathbf{u}}^\mathbf{i}$	Position vector of the point $\mathbf{P^i}$ of the body i with respect to the body reference
$\mathbf{u_1^i}, \mathbf{u_2^i}, \mathbf{u_3^i}$	Components of the vector $\mathbf{u^i}$ in a global frame of reference
$\bar{\mathbf{u}}_\mathbf{1}^\mathbf{i}, \bar{\mathbf{u}}_\mathbf{2}^\mathbf{i}, \bar{\mathbf{u}}_\mathbf{3}^\mathbf{i}$	Components of the vector $\mathbf{u^i}$ in the local frame of reference of body i
$\bar{\mathbf{u}}_\mathbf{4}^\mathbf{i}$	4x1 position vector of the point $\mathbf{P^i}$ of the body i with respect to the body reference
V_b	Volume of bone [m^3]
x	Input value of the controller
y	Output value of the controller signal

Greek symbols

α	Yaw angle [rad]
α_{i-1}	Link twist [rad]
β	Pitch angle [rad]
γ	Roll angle [rad]
θ	Euler angle of first rotation [rad]
θ_{0n}	Initial angular position [rad]
θ_i	Joint angle [rad]
θ_n	Actual angular position [rad]
$\dot{\theta}_n$	Angular velocity [rad/s]
ρ_b	Density of bone [kg/m^3]
φ	Euler angle of second rotation [rad]
ψ	Euler angle of third rotation [rad]

Abbreviations

3D	Three-dimensional
BMI	Body Mass Index
CAD	Computer Aided Design
DOF	Degrees of Freedom
IMRAD	Introduction, Methods, Results and Discussion
IR	Infrared
MBS	Multibody System
PEC	Parallel Elastic Component
RGB	Red Green Blue
SDK	Software Development Kit
SEC	Series Elastic Component
TCP/IP	Transmission Control Protocol/Internet Protocol
VR	Virtual Reality

1 INTRODUCTION

Computer real-time simulation is a tool which can be used in different areas of life and industry such as games, logistics, robotics, electrical and mechanical engineering. Real-time simulation approach allows simulating various systems via user feedback. Such systems are vehicles, heavy machinery, planes, trains, humans, and others. In the mechanical engineering industry, real-time simulation is used to solve problems of industrial machines, vehicles or robots reliably and safely by using a simulation model instead of a real machine or vehicle prototype.

There is a number of different software, which allows implementing a simulation approach for industrial and biomechanical solutions. Examples of such software are ADAMS, OpenSim and Mevea simulation software. Every mentioned software has its distinctive features and user interface. However, every simulation software package is based on the principles of multibody dynamics.

Mevea is a company that suggests various simulation solutions. The main application of Mevea software is creating of simulation multibody models of industrial machinery based on its real technical documentation. Mevea simulation software is able to simulate mechanics, hydraulics, power transmission and surrounding environment for the created model of vehicle or industrial machine. (Mevea Ltd. 2018a) Moreover, Mevea develops the hardware for real-time simulation. Developments of the company include VR (Virtual reality) glasses, armchair solutions, and Mevea cabin. VR glasses are applied to simulation and can be implemented as a user interface that allows a user to interact with the simulation with own hands. Armchair and Mevea cabin give the opportunity to simulate virtual machines based on Mevea real-time simulation as real machines. (Mevea Ltd. 2018b)

In addition to the simulation of industrial machinery, the subject of simulation of the human body is also an important issue. Creation of a human body simulation model opens up tremendous opportunities for various studies and researches in the fields of biomechanics and VR. One of the most important issues in biomechanics is the creation of a real-time

model of the human musculoskeletal system to recognize supporting forces and torques acting in muscles and joints, respectively.

Nowadays, computer musculoskeletal models and the motion capture technology are widely used in physiotherapy. Computer models are generated via motion capture, and then models are used for the determination of forces acting in people's muscles during physiotherapeutic rehabilitation sessions, shown in figure 1. This data is important for further observation of the rehabilitation process. The data could also be used to predict possible improvements in physical health after further physiotherapeutic sessions.

Figure 1. Motion capture technology in physiotherapeutic rehabilitation session (Cision 2015.).

Prepared real-time skeleton model is an alternative way to observe and collect the data, which is obtained from the physiotherapy session. The prepared model can be produced in Mevea software. The reason for using Mevea simulation software is the closest to reality physical engine. The software is able to show the forces and torques, which act in human muscles and articulations. Moreover, it is possible to connect the motion capture camera to the simulation model in order to control it.

The goal of the present research paper is to describe the modeling process of a novel real-time model of the human skeleton that is based on Mevea platform. The model has to be made for clinical purposes, mainly for physiotherapeutic sessions. The novelty of the research is based on the fact that similar type of work was never done in Mevea software and it is a challenge at the same time due to software vehicle and machinery oriented real-time simulation.

1.1 Similar researches

It was mentioned that a human skeleton has never been modeled in Mevea software as a real-time simulation model. On the other hand, several types of research describe the modeling of a human musculoskeletal system in different simulation software for different purposes. Examples of these researches are described in this chapter.

The full human body model close to the real human body was made by L. P. Nedel and D. Thalman in 2000. The goal of the research was to create a model that is based on a real anatomical structure of the human. The realism of the body was an important issue in this research. The model was represented by the musculoskeletal system of the human and skin. The simplified skeleton of the model had 31 joints, and the full system had 62 degrees of freedom (DOF). Skeletal muscles of the body were created via making action lines between the attachment points at the bones. In addition to the action lines, the shape and deformation model of the muscle were also developed. The whole model was created in Body Builder Plus, which is an “interactive human body modeling system” (Nedel & Thalman 2000, p. 316.). The results of the study were represented by the human body model, which movements were similar to real human movements. (Nedel & Thalman 2000, pp. 306-321)

The research made by E. Y. S. Chao in 2003 is also associated with the human body modeling. The research was implemented in VIMS (Virtual Interactive Musculoskeletal System) software. The model contains the full body skeleton and muscles, which provide the full body motions. The main purpose of the work was to introduce the modeling software, however, several biomechanical tests were done. The examples of such tests are kinematic analysis, distribution of muscle forces, joint constraints analysis, bone stress analysis, and other biomechanical tests. The software showed acceptable results, which are rather close to reality. (Chao 2003, pp. 201-212)

Another full body musculoskeletal model was created by R. Al Nazer et al. in 2011. The purpose of the work was to analyze bone strains during human locomotion. The research was implemented via utilization of flexible multibody approach. Motion data were obtained by dint of motion capture camera. Tibia was modeled as a flexible body in order to obtain its strain values during the locomotion. The model was created in BRG.LifeMODE software based on ADAMS commercial software. (Al Nazer et al. 2011.)

The next model was created by T. Rantalainen and A. Klodowski. It was the model of the lower part of the human skeleton. The main goal of this research was to estimate lower limb skeletal loading in order to improve the solutions for bone fracture problems like osteoporosis or accidental bone fractures. This model was developed in LifeMODE package based on ADAMS software. ANSYS software was also used for finite element model analysis. (Rantalainen & Klodowski 2011.)

The next research related to human body modeling was done in 2016 by P. Pathirana et al. The purpose of the research was to create a real-time model of the human for telerehabilitation via using several Kinect sensors. The first stage of the research was the mathematical modeling of the whole multi-Kinect system. The goal was to make a model of translation and rotation of the Kinect sensor and also to provide the mathematical model for model-based state estimation using Kalman filter. Then, in order to obtain results, computer simulation and hardware tests were done. According to the results, the model was admitted as successful and robust. The end use of the model was stated as telerehabilitation and physiotherapy. (Pathirana et al. 2016.)

Another research, which was done by A. Bauer et al. in 2017, describes the real-time human body model in augmented reality. Kinect for Windows v2.0 was used to implement the research. The anatomical model was displayed automatically during Kinect usage. The system calibrated the user anatomical data such bones length and sent the data in real time to the software which created a user-specified model based on the data. (Bauer et al. 2017.)

1.2 Research Objectives

The research has two main objectives. The first goal of the research is to develop a real-time model of human skeleton in Mevea environment for clinical purposes. The main challenge of the work is to create a multibody model of a skeleton in software that is used mainly for heavy machinery real-time simulation. The skeleton has to be developed according to the principles of multibody dynamics and biomechanical research methods.

Then, the second objective of the research is to obtain and analyze motion data of human who will stand in front of a motion capture camera and move. The camera, which is used in this project, is Kinect for Windows v2.0, shown in figure 2. The goal is to obtain joints positions in space and joints torques from Mevea, if possible.

Figure 2. Kinect for Windows v2.0 (Kinect for Windows Product Blog 2014.).

1.3 Research questions

The research is based on the following research questions:

1. How can joints torque change in real-time when the skeleton model moves?
2. How to develop a skeletal multibody system in Mevea?
3. Is it possible to build a real-time model of the human skeleton in a simulation environment which is oriented to heavy machinery?
4. What challenges are possible during modeling?

1.4 Thesis structure

The structure of the thesis is based on the IMRAD academic paper structure and consists of six chapters that are Introduction, Theoretical background, Research methods, Results and Analysis, Discussion and Conclusion. Introduction chapter explains the motivation and main objectives of the research.

Theoretical background chapter is used for explanation of literature review based on approaches and knowledge that are used during the research implementation. It contains information about spatial multibody dynamics, as the basis of Mevea simulation software. The chapter also provides information about biomechanical research methods and the human musculoskeletal system in terms of anatomy.

Research methods chapter describes all software and hardware used in the project. It also describes the process of modeling and connection between software and hardware in the project. Results and analysis chapter explains the results of three case studies from the simulation model: the motion of elbow joint, the motion of thoracic joint and full skeleton real-time simulation with Kinect for Windows 2.0.

Discussion chapter describes possible applications of the real-time model, possible improvements and further researches based on the results of the model. Conclusion chapter summarizes all the aspects of research and answers to research questions.

2 THEORETICAL BACKGROUND

This chapter introduces all the supporting literature and knowledge that were used during the research carrying out. The theoretical review provides information that was involved in the modeling process. Firstly, this chapter presents the theory of spatial multibody dynamics as a basis of Mevea simulation software. Secondly, the chapter provides information about the anatomy of the human skeleton and biomechanical research methods that were used in mathematical modeling.

2.1 Spatial multibody dynamics

A multibody system (MBS) is such a system that is an assembly of subsystems called bodies. The main purpose of multibody dynamics is to explain ways of connection and determination of various rigid or deformable bodies in space. (Shabana 1998, p. 1.) Figure 3 shows an abstract multibody system, which includes five bodies with various joints and connections between them. In this subchapter, concentration is focused on the kinematics of multibody systems with rigid bodies.

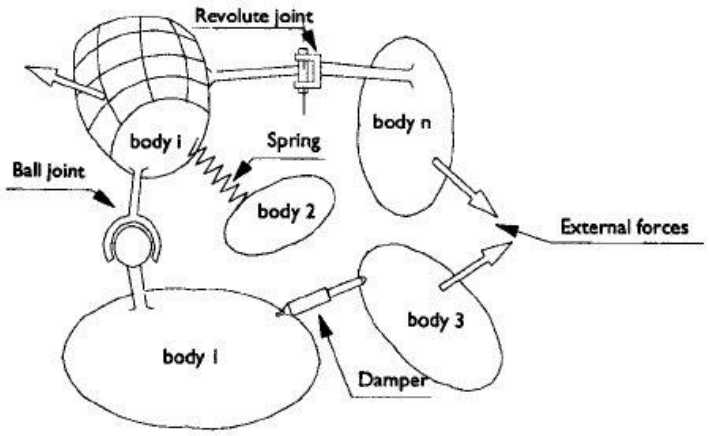


Figure 3. Multibody system (Neto & Ambrosio 2003, p. 83).

2.1.1 Frames of reference

Generally, two basic types of coordinate systems are needed in multibody systems. The first type is a coordinate system that is fixed in time and defines a standard for all bodies in the multibody system. This coordinate system is called a global coordinate system or inertial frame of reference. Each body in the multibody system has own coordinate system that

rotates and translates with the body with respect to the global coordinate system. Orientation and location of these frames of reference have the ability to change over time, and these coordinate systems are called local coordinate systems or body reference. Figure 4 depicts rigid body i in three-dimensional (3D) space, where the $X_1X_2X_3$ coordinate system appears as a global frame of reference, and $X_1^iX_2^iX_3^i$ is a local coordinate system of body i. It is assumed that vectors $\mathbf{i_1}, \mathbf{i_2}, \mathbf{i_3}$ and $\mathbf{i_1^i}, \mathbf{i_2^i}, \mathbf{i_3^i}$ are unit vectors along axes X_1, X_2, X_3 andX_1^i, X_2^i, X_3^i, respectively. (Shabana 1998, p. 4.) Then, vector $\mathbf{u^i}$ that determined in the local coordinate system of body i can be defined as (Shabana 1998, p. 4.):

$$\mathbf{u^i} = \bar{\mathbf{u}}_\mathbf{1}^\mathbf{i}\mathbf{i}_\mathbf{1}^\mathbf{i} + \bar{\mathbf{u}}_\mathbf{2}^\mathbf{i}\mathbf{i}_\mathbf{2}^\mathbf{i} + \bar{\mathbf{u}}_\mathbf{3}^\mathbf{i}\mathbf{i}_\mathbf{3}^\mathbf{i} \quad (1)$$

where $\bar{\mathbf{u}}_\mathbf{1}^\mathbf{i}, \bar{\mathbf{u}}_\mathbf{2}^\mathbf{i}$, and $\bar{\mathbf{u}}_\mathbf{3}^\mathbf{i}$ are components of the vector $\mathbf{u^i}$ in the local frame of reference of body i. Vector $\mathbf{u^i}$ can also be expressed from a global frame of reference point of view. (Shabana 1998, pp. 4-5.) It can be defined as (Shabana 1998, p. 5.):

$$\mathbf{u^i} = \mathbf{u}_\mathbf{1}^\mathbf{i}\mathbf{i}_\mathbf{1} + \mathbf{u}_\mathbf{2}^\mathbf{i}\mathbf{i}_\mathbf{2} + \mathbf{u}_\mathbf{3}^\mathbf{i}\mathbf{i}_\mathbf{3} \quad (2)$$

where $\mathbf{u}_\mathbf{1}^\mathbf{i}, \mathbf{u}_\mathbf{2}^\mathbf{i}$, and $\mathbf{u}_\mathbf{3}^\mathbf{i}$ are components of the vector $\mathbf{u^i}$ with respect to a global frame of reference. Thus, the same vector can be expressed with respect to the global frame of reference as well as the local body frame of reference. (Shabana 1998, p. 5)

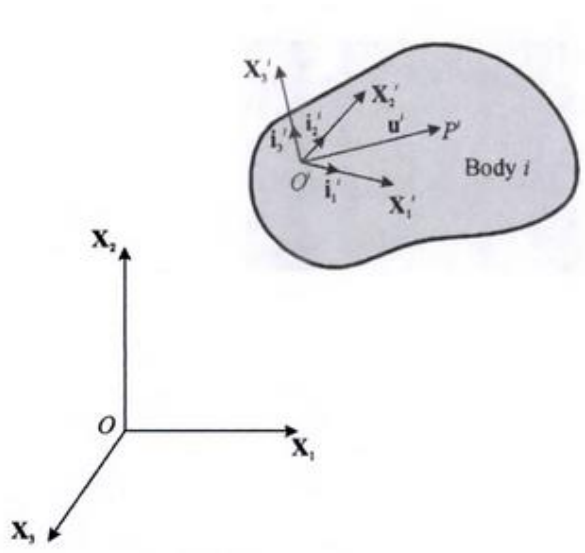


Figure 4. Global and body reference coordinate systems (Shabana 1998, p. 5.).

2.1.2 General displacement of the body

Six coordinates are used to determine the configuration of a rigid body in 3D space. Three of the coordinates are used to introduce body translation, and three coordinates are used to introduce the orientation of a rigid body. Figure 5 depicts the mechanics of rigid body i in space. Position $\mathbf{r^i}$ with respect to a global frame of reference O of the random point $\mathbf{P^i}$ on the body in 3D space can be expressed through equation 3:

$$\mathbf{r^i} = \mathbf{R^i} + \mathbf{u^i} \quad (3)$$

where $\mathbf{R^i}$ is global displacement vector of local body coordinate system and $\mathbf{u^i}$ is position vector of the point $\mathbf{P^i}$ in relation to O^i. (Shabana 1998, p. 11.)

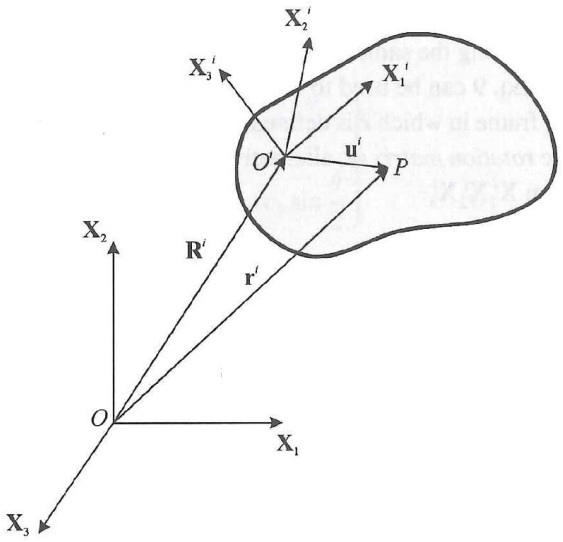


Figure 5. Mechanics of a rigid body (Shabana 1998, p. 34.).

It is possible to define the general displacement of the body i with rotation and translation of the body. In this case, vector $\mathbf{u^i}$ can be represented as:

$$\mathbf{u^i} = \mathbf{A^i}\bar{\mathbf{u}}^\mathbf{i} \quad (4)$$

where $\mathbf{A^i}$ is general rotation matrix, which determines the orientation of the body regarding global frame of reference, and $\bar{\mathbf{u}}^\mathbf{i}$ is position vector of point $\mathbf{P^i}$, and it is constant with respect to body i local coordinate system. Then, position vector $\mathbf{r^i}$ of the point $\mathbf{P^i}$ can be defined as:

$$\mathbf{r^i} = \mathbf{R^i} + \mathbf{A^i}\bar{\mathbf{u}}^\mathbf{i} \quad (5)$$

Equation 5 can be used to define general displacement in MBS that are consisted of several rigid bodies in 3D space. (Shabana 1998, pp. 33-34.)

2.1.3 General rotation matrix

The orientation of the body or the coordinate system of the body in 3D space is determined by a 3x3 rotation matrix. The prevalent way to determine the total rotation of the body reference with respect to the global coordinate system is to obtain rotation matrices $\mathbf{A^i_x}$, $\mathbf{A^i_y}$, and $\mathbf{A^i_z}$ which define successive rotations about x, y and z-axes respectively. The rotations around x, y and z-axes are called yaw, pitch and roll. Yaw, pitch and roll rotations are represented with rotation matrices shown in equations 6, 7 and 8 respectively:

$$\mathbf{A^i_x}(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\alpha & -sin\alpha \\ 0 & sin\alpha & cos\alpha \end{bmatrix} \quad (6)$$

$$\mathbf{A^i_y}(\beta) = \begin{bmatrix} cos\beta & 0 & sin\beta \\ 0 & 1 & 0 \\ -sin\beta & 0 & cos\beta \end{bmatrix} \quad (7)$$

$$\mathbf{A^i_z}(\gamma) = \begin{bmatrix} cos\gamma & -sin\gamma & 0 \\ sin\gamma & cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

where $\alpha,\ \beta,\ \gamma$ are yaw, pitch and roll angles. (Ipfs.io, 2018.)

General rotation matrix $\mathbf{A^i}$ describes the total rotation of the body reference with respect to the global coordinate system. General rotation matrix can be formed through multiplying three rotation matrices of rotation about x, y and z axes as seen in equation 9:

$$\mathbf{A^i} = \mathbf{A^i_x}(\alpha)\mathbf{A^i_y}(\beta)\mathbf{A^i_Z}(\gamma) \quad (9)$$

(Ipfs.io, 2018.).

2.1.4 Euler angles

One of the most common ways of explaining body orientation in 3D space is using Euler angles. Euler angles are three independent angles that describe rigid body rotation in space. Euler angles include three successive rotations at which a transformation occurs between the initial and final coordinate system orientations of the rotating body. (Shabana 1998, p. 67.)

Firstly, it is assumed that there are two coinciding coordinate systems $X_1X_2X_3$ and $\xi_1\xi_2\xi_3$. Coordinate system $\xi_1\xi_2\xi_3$ rotates at φ degrees around X_3 axis. The result of the rotation is depicted in figure 6 (a). The rotation matrix $\mathbf{D_1}$ describes the first rotation:

$$\mathbf{D_1} = \begin{bmatrix} cos\varphi & -sin\varphi & 0 \\ sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Then, it is assumed that the coordinate system $\eta_1\eta_2\eta_3$ coincides with the coordinate system $\xi_1\xi_2\xi_3$. Coordinate system $\eta_1\eta_2\eta_3$ rotates at the amount of θ degrees around ξ_1 axis, which is called a line of nodes in this case. The result of the rotation is depicted in figure 6 (b). Matrix $\mathbf{D_2}$ describes second rotation:

$$\mathbf{D_2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & sin\theta & cos\theta \end{bmatrix}$$

Next, it is assumed that the coordinate system $\zeta_1\zeta_2\zeta_3$ coincides with the coordinate system $\eta_1\eta_2\eta_3$. System $\zeta_1\zeta_2\zeta_3$ rotates around η_3 axis at the number of ψ degrees. Figure 6 (c) depicts the result of the rotation around the axis η_3. Matrix $\mathbf{D_3}$ is used to describe third successive rotation:

$$\mathbf{D_3} = \begin{bmatrix} cos\psi & -sin\psi & 0 \\ sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Finally, transformation matrix $\mathbf{A^T}$ shows the total amount of rotation of the coordinate system $\zeta_1\zeta_2\zeta_3$ with respect to the initial coordinate system $X_1X_2X_3$ and equal to the multiplication of three rotation matrices $\mathbf{D_1}, \mathbf{D_2}$ and $\mathbf{D_3}$ as shown in equation 10. (Shabana 1998, p. 67-68.)

$$\mathbf{A^T} = \mathbf{D_1 D_2 D_3} \quad (10)$$

Thus, transformation matrix $\mathbf{A^T}$ is equal to (equation 11):

$$\mathbf{A^T} = \begin{bmatrix} cos\psi cos\varphi - cos\theta sin\varphi sin\psi & -sin\psi cos\varphi - cos\theta sin\varphi cos\psi & sin\theta sin\varphi \\ cos\psi sin\varphi + cos\theta cos\varphi sin\psi & -sin\psi sin\varphi + cos\theta cos\varphi cos\psi & -sin\theta cos\varphi \\ sin\theta sin\psi & sin\theta cos\psi & cos\theta \end{bmatrix} \quad (11)$$

where angles φ, θ and ψ are Euler angles. (Shabana 1998, p. 68.)

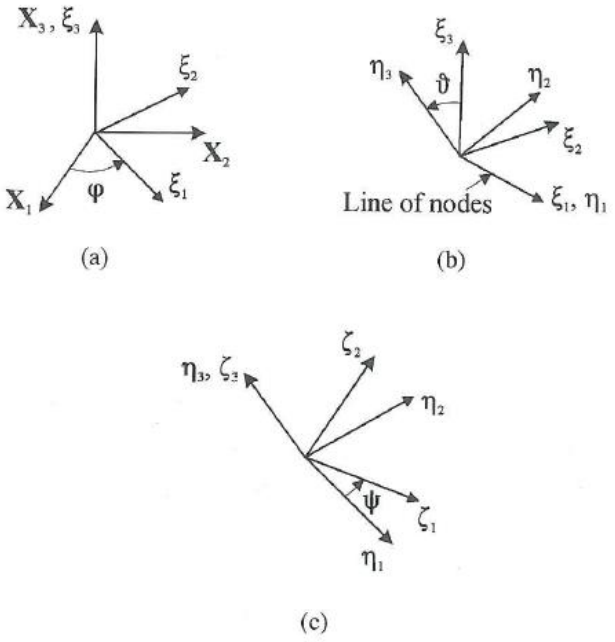


Figure 6. Euler angles: (a) – First rotation, (b) – Second rotation, (c) – Third rotation (Shabana 1998, p. 67.).

2.1.5 Denavit-Hartenberg transformation

Equation 5 describing the position of a random point $\mathbf{P^i}$of rigid body i with respect to the global frame of reference can be written in an alternative way by using a 4x4 transformation matrix. Then, equation 12 represents the position of the point $\mathbf{P^i}$:

$$\mathbf{r_4^i} = \mathbf{T_4^i}\bar{\mathbf{u}}_\mathbf{4}^\mathbf{i} \quad (12)$$

where $\mathbf{r_4^i}$ and $\bar{\mathbf{u}}_\mathbf{4}^\mathbf{i}$ are 4x1 vectors $\mathbf{r_4^i} = [r_1^i\ r_2^i\ r_3^i\ 1]^T$and $\bar{\mathbf{u}}_\mathbf{4}^\mathbf{i} = [\bar{u}_1^i\ \bar{u}_2^i\ \bar{u}_3^i\ 1]^T$and matrix $\mathbf{T_4^i}$ is 4x4 transformation matrix also called homogeneous transformation matrix. 4x4 homogeneous transformation matrix includes the 3x3 general rotation matrix and 3x1 displacement vector of body local coordinate system as seen from equation 13:

$$\mathbf{T_4^i} = \begin{bmatrix} \mathbf{A^i} & \mathbf{R^i} \\ \mathbf{0_3^T} & 1 \end{bmatrix} \quad (13)$$

where $\mathbf{A^i}$ is general rotation matrix of the body local coordinate system, $\mathbf{R^i}$ is displacement vector of body local coordinate system and $\mathbf{0_3^T}$ is null vector $\mathbf{0_3} = [0\ 0\ 0]^T$. (Shabana 1998, p. 76.)

The 4x4 homogeneous transformation matrix combines both translation and orientation in space in one matrix. This matrix is not orthogonal. 4x4 transformation matrix also can be used to explain the robotic manipulator joints and their links connected into large kinematic chains. (Shabana 1998, p. 76.)

Denavit-Hartenberg transformation is the most common method that can be used in description of relative translation and rotation of the body. This method is based on utilization of a 4x4 homogeneous transformation matrix, which is a function of four parameters. Figure 7 illustrates two connected links: link i and link i-1. It is assumed that there are two coordinate systems $X_iY_iZ_i$ and $X_{i-1}Y_{i-1}Z_{i-1}$, which z-axes coincide with joint axes of the joint i+1 and joint i respectively. The distance between these joint axes measured along the line perpendicular to both axes is called link length a_{i-1} and it is the first Denavit-Hartenberg parameter. Link twist α_{i-1} is the second Denavit-Hartenberg parameter and it is an angle measured with respect to X_i from Z_{i-1} to Z_i. Distance d_i is called link offset and

it determines the distance between X_i and X_{i-1} along the z-axis. Angle θ_i is the last parameter used in Denavit-Hartenberg transformation. It describes the rotation of link i with respect to link i-1 about the axis of joint i. It is called a joint angle. (Shabana 1998, pp. 81-82.)

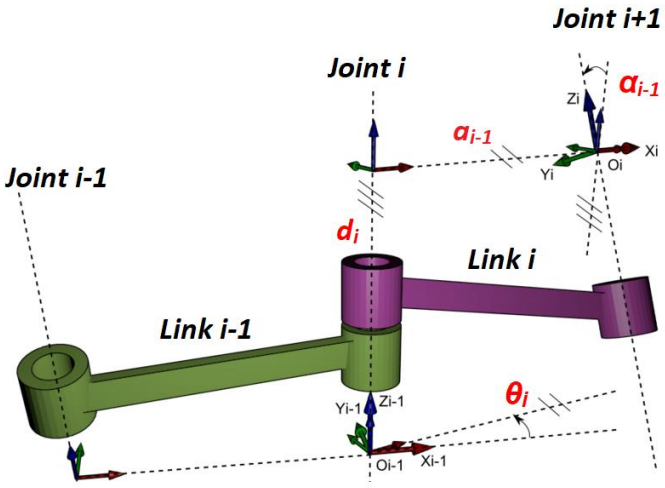


Figure 7. Denavit-Hartenberg parameters (Abdeetedal 2014.).

To describe the position and orientation of $X_iY_iZ_i$ frame of reference with respect to $X_{i-1}Y_{i-1}Z_{i-1}$ frame of reference all four parameters can be used in one 4x4 transformation matrix. This matrix is a multiplication of the 4x4 transformation matrix of every Denavit-Hartenberg parameter. Link offset d_i represents only the distance along z-axis and it means that displacement vector is introduced only with z-coordinate. In this case, matrix $\mathbf{T_1}$ looks as shown in equation 14:

$$\mathbf{T_1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

Joint angle θ_i is rotation about the z-axis of the joint i. Second transformation matrix $\mathbf{T_2}$ that depends on a joint angle is introduced as shown in equation 15.

$$\mathbf{T_2} = \begin{bmatrix} cos\theta_i & -sin\theta_i & 0 & 0 \\ sin\theta_i & cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

Third transformation matrix $\mathbf{T_3}$ is based on link length a_{i-1}. It is shown in equation 16.

$$\mathbf{T_3} = \begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

The last transformation matrix $\mathbf{T_4}$ shows the orientation of Z_i to Z_{i-1} with respect to x-axis and it is based on link twist parameter α_{i-1}, as shown in equation 17

$$\mathbf{T_4} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\alpha_{i-1} & -sin\alpha_{i-1} & 0 \\ 0 & sin\alpha_{i-1} & cos\alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (17)$$

Then, the resultant 4x4 transformation matrix is a multiplication of four transformations based on 4 Denavit-Hartenberg parameters:

$$\mathbf{T^{i,i-1}} = \mathbf{T_1 T_2 T_3 T_4}$$

where $\mathbf{T^{i,i-1}}$ is 4x4 Denavit-Hartenberg transformation matrix that is equal to (equation 18):

$$\mathbf{T^{i,i-1}} = \begin{bmatrix} cos\theta_i & -sin\theta_i & 0 & a_{i-1} \\ sin\theta_i cos\alpha_{i-1} & cos\theta_i cos\alpha_{i-1} & -sin\alpha_{i-1} & -d_i sin\alpha_{i-1} \\ sin\theta_i sin\alpha_{i-1} & cos\theta_i sin\alpha_{i-1} & cos\alpha_{i-1} & d_i cos\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (18)$$

(Shabana 1998, pp. 83-84.)

2.1.6 Degrees of freedom

The ability of bodies in MBS to rotate and translate in space is limited by the degrees of freedom (DOF) of the system. Then, the number of degrees of freedom of the body depends

on the mechanical joints between the bodies, which create constraints for movements. In the kinematic chain, the motion of bodies cannot be independent of each other. Mechanical joints can be described mathematically through using a set of constraint equations, where each equation limits possible motion of the whole MBS. Number of DOF in MBS can be calculated as shown in equation 19 also called Kutzbach criterion:

$$DOF = 6 \times n_b - n_c \quad (19)$$

where n_b is a number of bodies in a rigid body system and n_c is a number of independent constraint equations. (Shabana 1998, p. 19.)

2.1.7 Mechanical joints

Mechanical or kinematic joints are used to provide the connection between the bodies in MBS. There are several types of joints that constraint the motion. Figure 8 shows the most common types of mechanical joints used in multibody systems modeling. These joints are revolute, spherical, translational, cylindrical, fixed, planar and universal. (Blundell & Harty 2004, p. 95.)

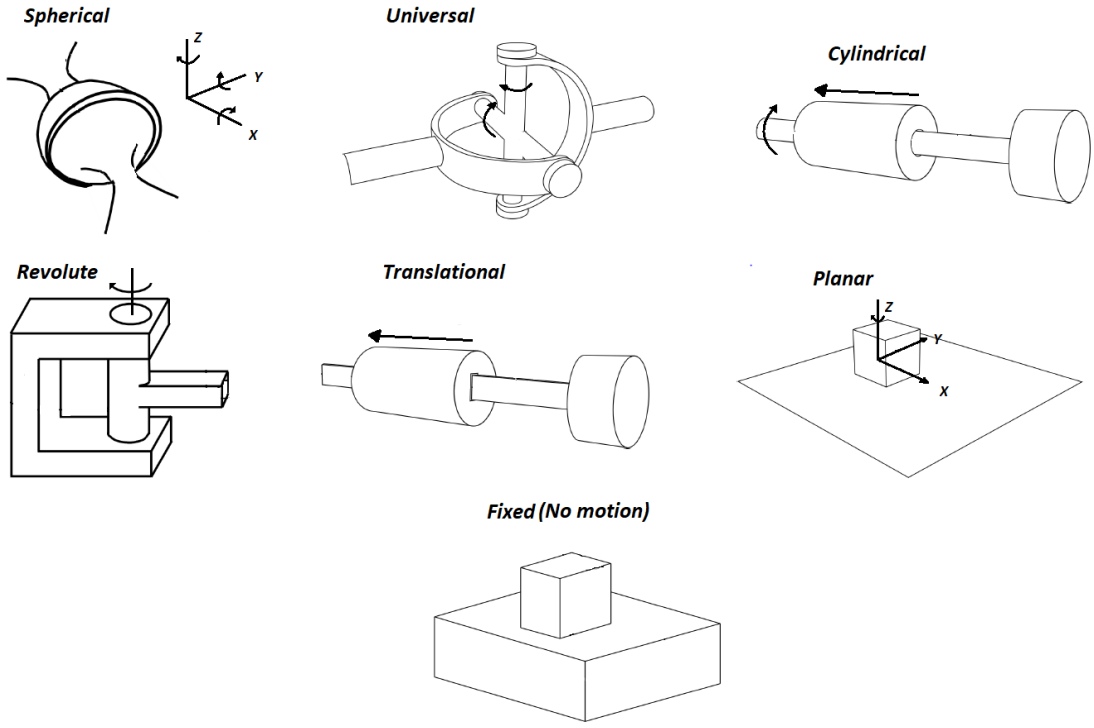


Figure 8. Commonly used mechanical joints (Mevea Ltd. 2018c, pp. 20-27).

Each joint type has a specific set of constraint equations that limit the movement of the joint. Table 1 shows joints constraints that are important while calculating system degrees of freedom. It is also possible to determine what type of constraint (rotational or translational) is inherent in a particular mechanical joint type. (Blundell & Harty 2004, p. 99.)

Table 1. Mechanical joints constraints (Blundell & Harty 2004, p. 99.).

Constraint element	**Translational constraint**	**Rotational constraint**	**Total constraints**
Cylindrical joint	2	2	4
Fixed joint	3	3	6
Planar joint	1	2	3
Revolute joint	3	2	5
Spherical joint	3	0	3
Translational joint	2	3	5
Universal joint	3	1	4

2.2 Biomechanics

The dictionary defines biomechanics as a section of science that deals with the study of various internal and external forces that act on the bodies of living organisms. Moreover, biomechanics is committed to studying the mechanical nature of biological processes inside the organism such as muscles movement. (Dictionary.com 2018) This chapter explains all the biomechanical issues that are related to particular research such as a description of the human skeletal system, the role of muscles and biomechanical research methods that can be used in research implementation.

2.2.1 Human skeleton

Normally, the adult person skeleton contains 206 bones. All bones are divided into two groups - bones of the axial skeleton and bones of the appendicular skeleton. Figure 9 shows

the distribution of the bones of the axial (blue) and appendicular (red) skeleton. Bones of axial skeleton compose imaginary axis of the human body, which passes through the human body center of gravity. These bones are skull bones, ribs, vertebral column. Main function of the axial skeleton is protection of the important organs. Appendicular skeleton includes bones of lower and upper limbs and girdles that link limbs to the axial skeleton. The main function of the appendicular skeleton is to support the body and locomotion. (Tortora & Grabowski 1993, pp. 166, 169.)

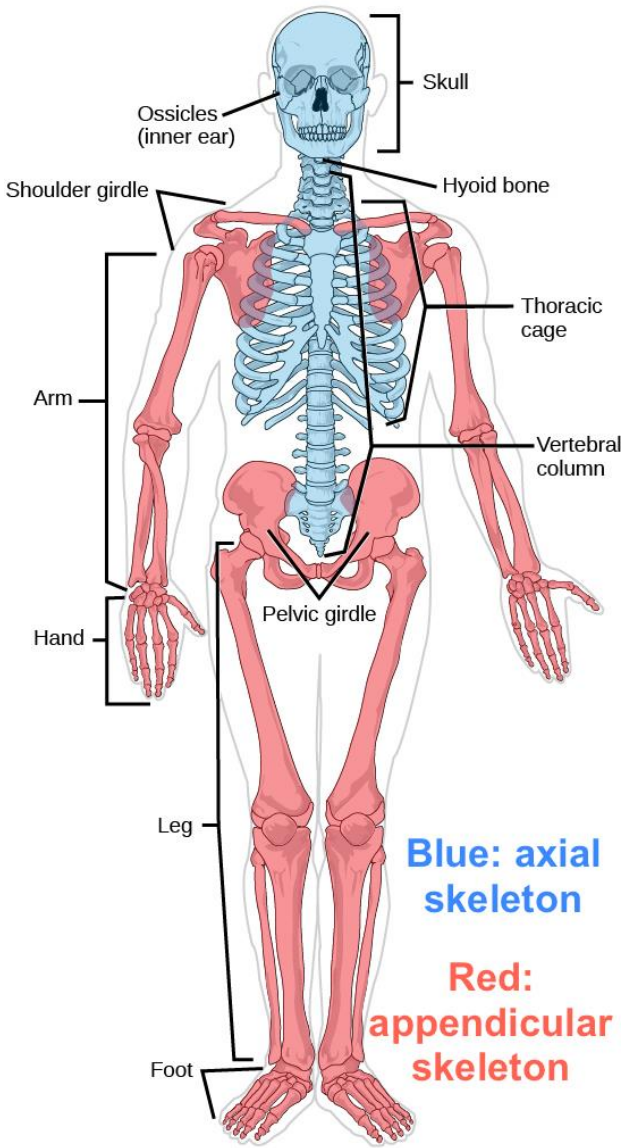


Figure 9. Axial and appendicular skeleton (Khan Academy 2018.).

A skeletal bone is composed of several types of osseous tissue: dense or cortical bone and cancellous or trabecular bone. Cortical bone is rather dense and compact compared to

cancellous bone, which structure looks like a sponge. (Hamill et al. 2015, pp. 31-32.) The density of wet cortical bone is equal to 1990 kg/m^3. The density of cancellous bone varies from 0.05 to 1.1 g/cm^3 (Murphy et al. 2016, pp. 4, 15.). Low porosity (less than 15%) is also a feature of the cortical bone, unlike cancellous, which porosity may reach more than 70%. (Hamill et al. 2015, pp. 31-32.) Figure 10 shows the cross-sectional difference between cortical and cancellous bones in the human femur.

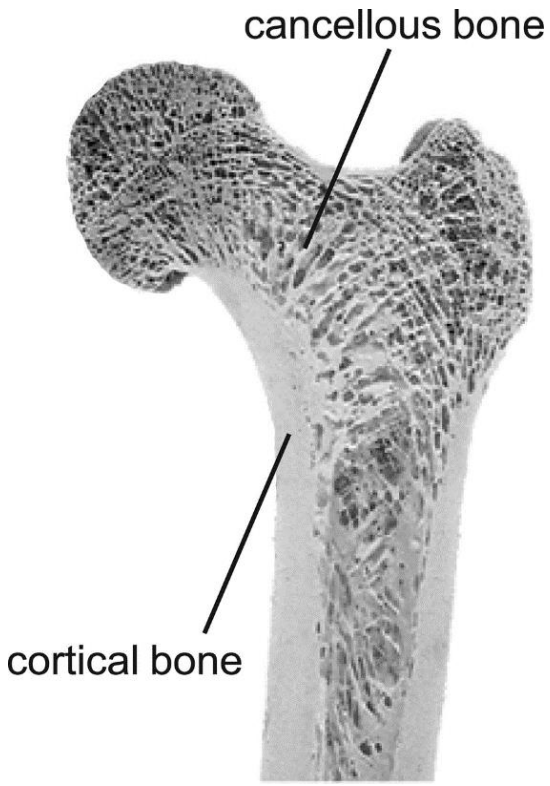


Figure 10. Difference between the cortical and cancellous bone in the human femur (Willems et al. 2013, p. 480.).

Bones are divided into several groups based on anatomical classification. Bones can be long, short, flat or irregular. The relation of the bones to a particular group determines their function. Bones which length is much greater than width are called long bones. The main functions of long bones are to support the whole skeleton and to provide body movements. Long bones include such bones as hand bones (humerus, radius, ulna, metacarpals, finger phalanges), leg bones (femur, tibia, fibula, metatarsals, toe phalanges) and clavicles. (Hamill et al. 2015, p. 32.)

According to Hamill, short bones consist mainly of the cancellous bone, which is covered by a thin layer of cortical bone. The main function of these bones is the transmission of forces. Examples of such bones are the carpals and tarsals of the arms and legs, respectively. Separately it is necessary to highlight sesamoid bone. These bones are embedded exactly in the tendon. The function of the sesamoid bone is to change the angle of insertion of the muscle and reduce the friction created by the muscle. The main example of such a bone is patella. (Hamill et al. 2015, pp. 32-33.)

Another two types of bones are flat and irregular bones. According to Hamill, flat bones consist of two layers of cortical bone and cancellous bone between these two layers. Flat bones include such bones as bones of the chest (sternum, ribs), scapula, and ilium. The main function is to protect the internal organs and organ systems. Bones that have an unusual shape and do not fit into other groups can be attributed to irregular bones. Irregular bones have many different functions, such as protecting the spinal cord and brain, providing attachment of muscles and supporting the weight. Most significant irregular bones are pelvis, various skull bones, and vertebral column. (Hamill et al. 2015, p. 33.)

2.2.2 Types of skeletal moving joints

Biologically, the human skeletal system has its own joints. Body joints called articulations. Articulations can be diarthrodial or synovial, fibrous and cartilaginous. Fibrous and cartilaginous articulations are restricted in movement. Synovial articulations or joints play a major role in the movement of the skeleton. This type of joint is a very low-friction joint and it has the capability to withstand wear. It is also very stable due to the surrounding ligaments, capsule, and tendons of the joint. The negative atmospheric pressure produces the vacuum inside the articulation. Types of synovial joints are similar to well-known mechanical joints, which are used in the creation of mechanical multibody systems. Synovial joints are classified into seven types in terms of biology. Figure 11 shows the main types of human articulation (Hamill et al. 2015, pp. 49-51.).

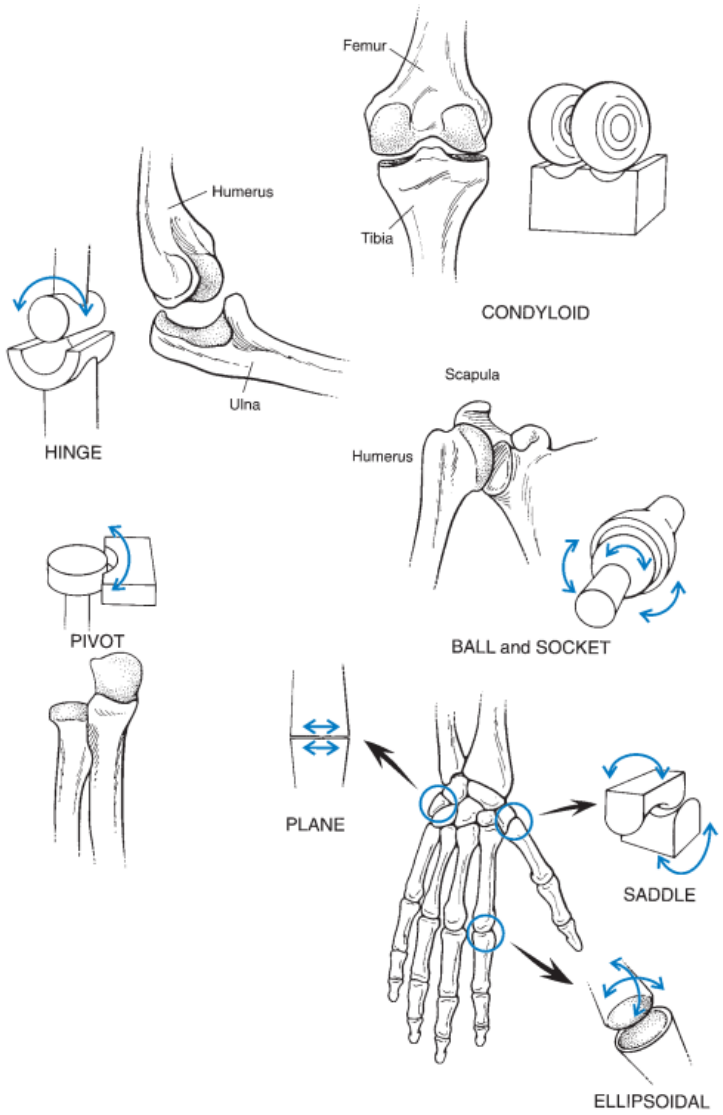


Figure 11. Synovial articulation types (Hamill et al. 2015, p. 52.).

The hinge joint is analogue of a revolute mechanical joint due to its ability to move in one plane, rotating around the axis. Pivot joint has also one degree of freedom It is able to rotate around one joint axis as depicted in figure 11. Ball-and-Socket articulation is analogue of spherical mechanical joint and has three degrees of freedom with the ability to rotate around three axes. Plane or gliding joint has the ability to move in one plane in three different sides as a planar mechanical joint. Saddle, ellipsoid and condyloid articulations have two degrees of freedom and ability to move in two planes that are flexion and extension, abduction and adduction. The function of these joints is almost similar with the only difference in anatomical structure, which can be seen in figure 11. The mechanical analogue of these joints is universal, which also has two rotational degrees of freedom. All examples of synovial

articulations of different types in the human skeleton are shown in table 2. (Hamill et al. 2015, p. 51.)

Table 2. Examples of synovial articulations (Hamill et al. 2015, p. 53.).

Joint	**Type**	**Degrees of freedom**
Hip	Ball-and-Socket	3
Shoulder	Ball-and-Socket	3
Knee	Condyloid	2
Wrist	Ellipsoid	2
Elbow	Hinge	1
Ankle	Hinge	1
Carpometacarpal (thumb)	Saddle	2
Radioulnar	Pivot	1

2.2.3 Muscle mechanics

Muscles make a huge contribution to the movement of the human body, and this function of the muscles in the body is certainly the most significant. In addition, muscles maintain the stability of joints due to the tendons that transmit muscle forces to the joints. However, there are also several other functions that are not directly related to the movement. Muscles serve as additional protection for organ systems and maintain the pressure inside body cavities. Many muscles can also support processes that control inputs and outputs of the human body, such as swallowing and urination. Finally, an important function is to maintain body temperature by muscles, due to the production of heat in the process of contraction. (Hamill et al. 2015, p. 62.)

Moreover, in the mechanics point of view, muscles can develop a torque in human body joints. Force vector that acting along the muscle can create a rotation in the joint axis by applying that force to a bone. The initial and final points of the line of action of the force are

determined by the places of attachment of the muscle to the bones. Then, torque can be obtained as a product of muscle force magnitude and the perpendicular distance from the axis of rotation to a line of action of the muscle force. (Hamill et al. 2015, p. 72.) Mathematically, the torque acting in body joints can be defined as shown in equation 20:

$$T = F \times r_F \quad (20)$$

where T is a torque, F is applied muscle force and r_F is moment arm which is perpendicular distance from torque axis to the force vector (Hamill et al. 2015, p. 72.). The amount of generated torque is strongly dependent on muscle capacity to produce force and cannot exceed its ability. According to Hamill, moment arm may vary depending on the force vector line of pull of a particular muscle relative to the joint. (Hamill et al. 2015, p. 72.)

To perform a biomechanical study, it is necessary to derive a mathematical model of the muscle. Scientist A. V. Hill described one of the frequently used biomechanical muscle models. This model consists of three components that can describe the natural mechanics of the muscle. The scheme of Hill's model is depicted in figure 12. (Hamill et al. 2015, p. 70.)

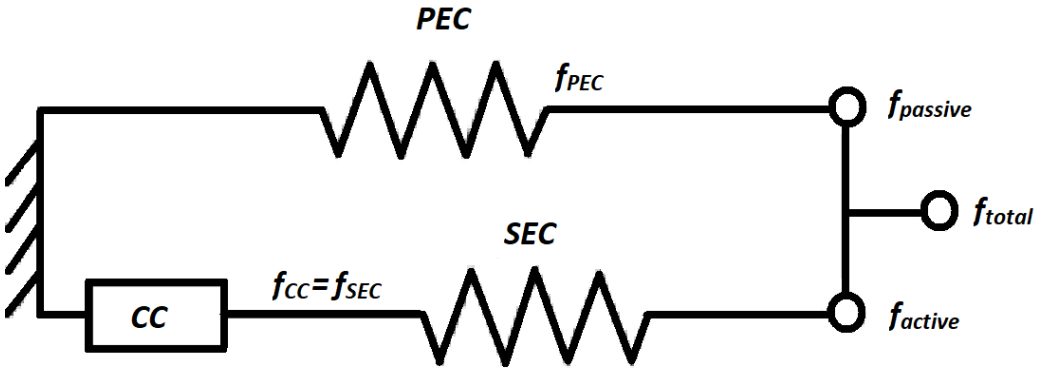


Figure 12. Hill's model of muscle (Hamill et al. 2015, p. 71.).

The scheme shows three components of the model that are parallel elastic component (PEC), series elastic component (SEC) and CC. CC component is the component of the nervous system stimulation signal that is converted to the force. CC component also measures how the signal from human brain is converted into the force. SEC and PEC are nonlinear elastic

components that represent elastic elements of the muscle. They are both behavioral models and there cannot be made associations with any muscular structures. Basically, SEC and PEC are an elastic response to the muscle contractions with the main difference that SEC acts in series with CC element and takes the force generated by CC into account. PEC works only when the CC component is not producing force, and an external force causes the muscle to resist. In other words, PEC is responsible for the passive reaction of the muscle, while SEC is responsible for active reaction, based on higher nervous activity. (Hamill et al. 2015, pp. 70-71.)

2.2.4 Inertial properties of body segments

Each segment of the body such as every bone has a moment of inertia. In 3D space the moment of inertia of the segment can be defined with 3x3 matrix **I** called a moment of inertia tensor:

$$\mathbf{I} = \begin{bmatrix} I_x & P_{xy} & P_{xz} \\ P_{yx} & I_y & P_{yz} \\ P_{zx} & P_{zy} & I_z \end{bmatrix} \quad (21)$$

where $I_x, I_y,$ and I_z are principal mass moments of inertia in 3D space and $P_{xy}, P_{xz}, P_{yx}, P_{yz}, P_{zx}, P_{zy}$ are mass products of inertia (Robertson et al. 2004, pp. 68-69.). Ideally, all nine elements should be calculated for the most accurate results, but in body segments calculations the inertia tensor can also be simplified to a diagonal 3x3 matrix, where mass products of inertia are equal to zero. (Robertson et al. 2004, p. 69.)

Human body segments can be represented as solid geometrical shapes. One of the well-known methods for dividing the human body into segments is the Hanavan geometric model of the body. The model was developed in 1964 and is relevant nowadays. Figure 13 depicts the Hanavan model with 15 segments of the human body. Thus, this approach greatly simplifies the calculation of the moment of inertia tensor, making possible to use formulas for simple geometric shapes. Then, it is assumed that every bone of the human skeletal system can be represented as a solid geometrical shape. (Robertson et al. 2004, p. 69.)

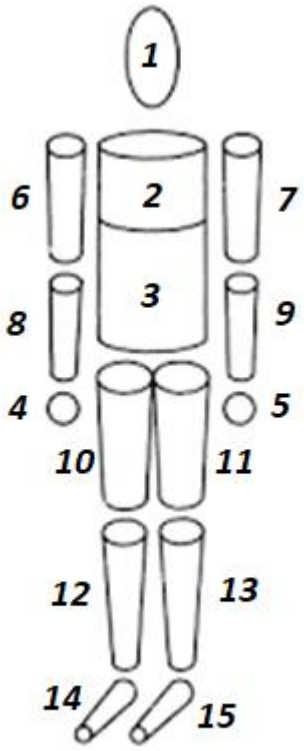


Figure 13. Hanavan human body model (Schüler et al. 2015, p. 147.).

It is possible to represent all types of bones with four types of solid geometrical shapes, which are rectangular prism, ellipsoid, sphere and a circular cylinder. Principal moments of inertia $I_x, I_y,$ and I_z for different geometrical shapes are defined on equations 22-30 that are shown in table 3. (Robertson et al. 2004, p. 69.)

Table 3. Principal mass moments of inertia of solid geometrical shapes (Robertson et al. 2004, pp. 70-71.).

Geometrical shape	$\boldsymbol{I_x}$	$\boldsymbol{I_y}$	$\boldsymbol{I_z}$
Rectangular prism	$I_x = \frac{1}{12}m(b_p^2 + c_p^2)$ (22)	$I_y = \frac{1}{12}m(a_p^2 + c_p^2)$ (23)	$I_z = \frac{1}{12}m(a_p^2 + b_p^2)$ (24)
where m=mass; a_p=depth (x); b_p=height (y); c_p=width (z)			
Circular cylinder	$I_x = \frac{1}{2}mr_c^2$ (25)	$I_y = I_z = \frac{1}{12}m(3r_c^2 + l_c^2)$ (26)	
where m=mass; l_c=length of cylinder; r_c=radius of cylinder			
Sphere	$I_x = I_y = I_z = \frac{2}{5}mr_s^2$ (27)		

Table 3 continues. Principal mass moments of inertia of solid geometrical shapes (Robertson et al. 2004, pp. 70-71.).

where m=mass; r_s=radius of sphere			
Ellipsoid	$I_x = \frac{1}{5}m(b_e^2 + c_e^2)$ (28)	$I_y = \frac{1}{5}m(a_e^2 + c_e^2)$ (29)	$I_z = \frac{1}{5}m(a_e^2 + b_e^2)$ (30)
where m =mass; a_e =depth of ellipsoid (x); b_e =height of ellipsoid (y); c_e =width of ellipsoid (z)			

3 RESEARCH METHODS

The present chapter explains the systematic implementation of the research. The research implementation is divided into several stages, which are performed separately. These stages are skeleton multibody modeling, implementation of real-time connection between hardware and software and testing of the system. The present chapter also contains the information about system hardware and software, which were used during the research implementation.

Skeleton multibody modeling stage includes the assembling of MBS in Mevea simulation software based on the human skeletal system with mechanical joints related to anatomical articulations. The MBS also includes an external interface for real-time simulation in Simulink.

The next stage of the research explains the method of hardware and software connection. The hardware is Kinect for Windows v2.0 sensor that is compatible with Matlab and Simulink. The purpose of this stage of research is to define the way to allow motion data transfer from Kinect to the Matlab and Simulink.

The last stage of the project is testing of the system and results obtaining. The testing of the system is divided into three separate case studies that are elbow joint motion case study, thoracic joint motion case study, and full skeleton testing. The results of the experiment are converted into data plots and are presented in the Results and Analysis section of the thesis (Ch. 4).

3.1 System under investigation

The system under investigation is the human skeletal system consisted of 135 rigid bodies, which represent bones of the skeleton. The average human skeleton of an adult person consists of 206 bones. The number of bones is reduced to 135 in order to decrease the probability of computer overloading, due to the large simulation model, and to simplify the modeling process. Several bones are combined in one rigid body, as it happens with numerous skull bones that were combined only in two actual rigid bodies that are skull and jaw. The same approach is used for ribs, carpals, tarsals, metacarpals, metatarsals, and

phalanges. A list of all the bodies used in MBS and their bones analogues in the human skeletal system is shown in Appendix I.

Graphics for the system was created in Autodesk 3ds Max 3D modeling software. After that, the graphics were converted to Mevea simulation software. Figure 14 (a) shows graphics that was made in 3ds Max. Figure 14 (b) depicts the MBS created in Mevea simulation software.

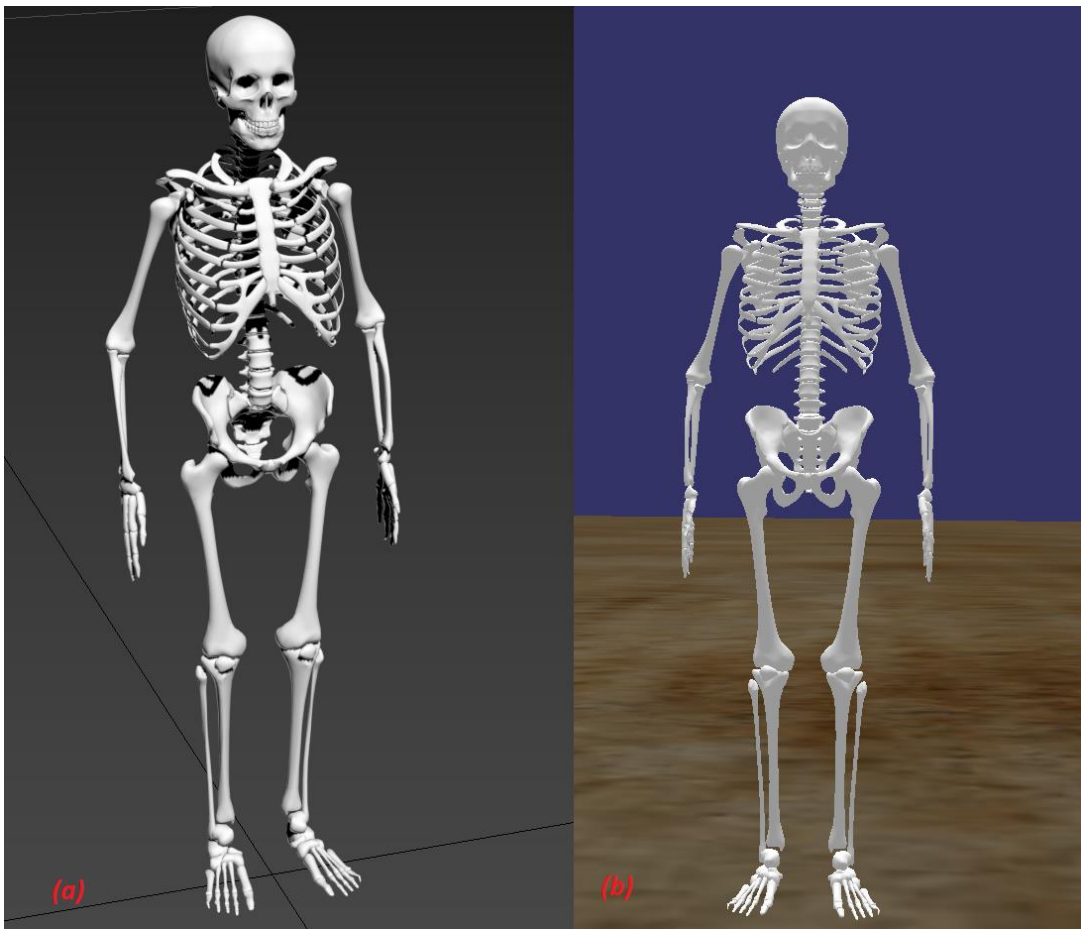


Figure 14. System under investigation: (a) – 3ds Max, (b) – Mevea.

The system is modeled according to the assumption that created skeletal system belongs to 24 years old male human with a height of 165 cm and body weight of 65 kg. This data is used to model needed height of skeleton and to calculate the skeletal mass that is estimated as 17% of full body mass of adult person with normal BMI (Body Mass Index) (Malina et al. 2004, p. 121.).

3.2 Software

Software packages that are used during the research are introduced in this sub-chapter. These software packages are Mevea Modeller, Mevea Solver, Autodesk 3ds Max, Matlab Simulink, and Microsoft Kinect SDK. The main simulation software packages of the project are Mevea Modeller and Mevea Solver. Autodesk 3ds Max is used for visualization graphics creation and also for several bone parameters detection such as the volume of the bone for average bone density calculations. Matlab Simulink is used as a connection between the Mevea software and Kinect for Windows v2.0 and for controlling the model. Microsoft Kinect SDK (Software Development Kit) function is to visualize the body joints of a human who moves in front of Kinect camera.

3.2.1 Mevea simulation software

Mevea is a modern real-time simulation software that offers a great opportunity and tools for real-time modeling and simulation of various MBS, especially heavy machinery or vehicles. Mevea physics engine is based on principles of multibody dynamics with scientifically proven formulas and calculations. (Mevea Ltd. 2018a) Mevea simulation software consists of four sub-software: Mevea Modeller, Mevea Solver, Mevea Visual Channel and Mevea Remote Access (Mevea Ltd. 2018d, p. 6.). This sub-chapter will cover information about Mevea Modeller and Mevea Solver simulation software.

Mevea Modeller is a software that allows creating real-time simulation models of various multibody systems. Modeling process includes multibody system creation with rigid or flexible bodies with local frames of references, mass, and inertial properties, motion constraints or body joints, body collisions, various forces and torques acting in the MBS. It is also possible to create power transmission and a hydraulic circuit, add an electrical motor or create inputs for joystick control. It is also possible to create an environment (ground, soil particles, sky, etc.) for the simulation.

The software also supports the graphics and CAD (Computer Aided Design) 3D models. Graphics cannot be created in such software because it is not 3D modeling software. (Mevea Ltd. 2018e, p. 5.) It is possible to add graphics and CAD models by importing it from different 3D modeling software such as 3ds Max or Solidworks.

Mevea Modeller user interface consists of four main windows. These windows are Preview Window, Body Preview Window, Model Tree and Object View. Figure 15 depicts Mevea Modeller user interface. Preview Window shows the whole simulation model with MBS and environment. Body Preview Window shows one selected body from the MBS with its local frame of reference. Model Tree shows components that are used in the simulation model. Object View window shows the parameters of one selected component from the Model Tree. (Mevea Ltd. 2018e, p. 6.)

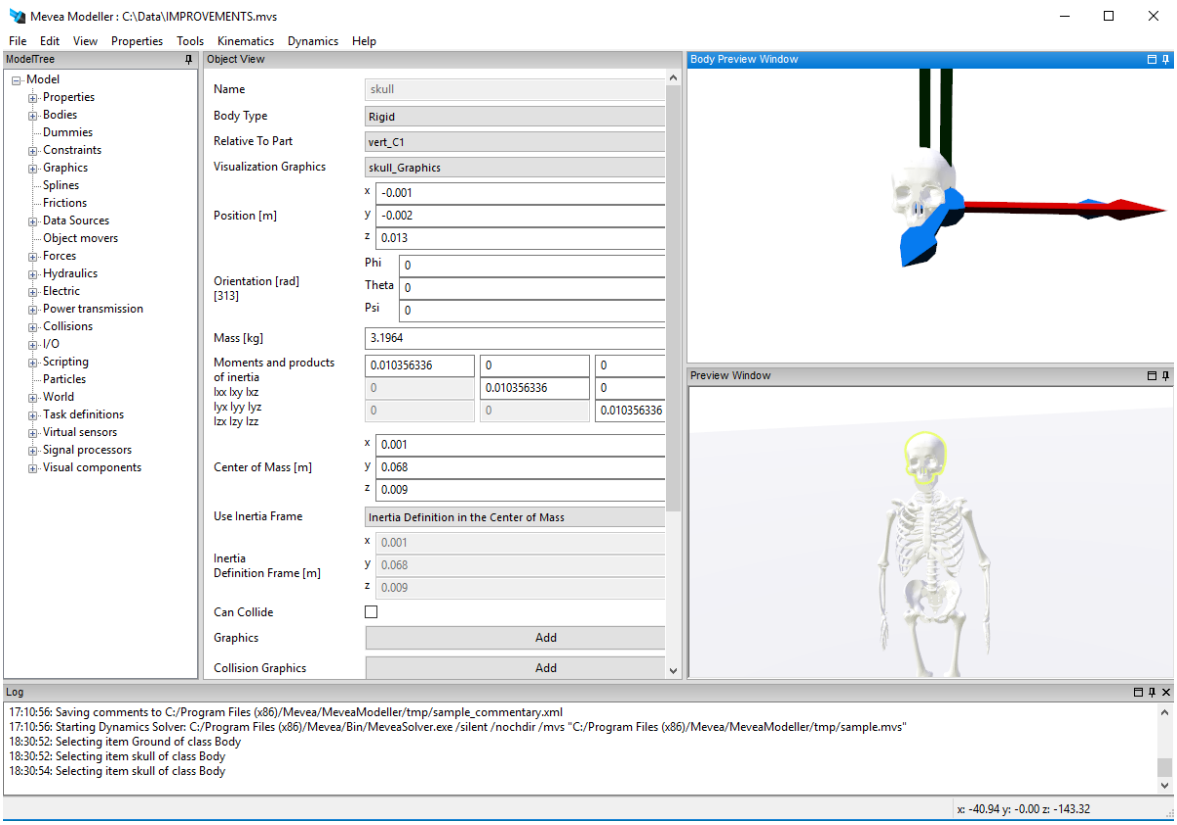


Figure 15. Mevea Modeller user interface.

Mevea Solver is a software that allows simulating and controlling the real-time simulation model that was developed in Mevea Modeller software (Mevea Ltd. 2018d, p. 6.). The controlling is based on controller inputs modeled in Mevea Modeller software that can be connected to the joystick or to the external interface (e.g. Simulink external interface). The simulation can also be controlled by the keyboard. (Mevea Ltd. 2018d, pp. 7-8.) Mevea Solver provides a possibility to obtain plots for different variable or components of the simulation model such as forces, torques, position, linear and angular velocity, etc. (Mevea Ltd. 2018c, p. 10.)

The interface of Mevea Solver is presented in figure 16. Mevea Solver user interface represents the window with a real-time simulation model. On the left, there are simulation control buttons that enable control of the simulation and also creates the possibility to choose the camera for the simulation view. (Mevea Ltd. 2018d, p. 14.) Keyboard control buttons can also be seen from figure 16 on the right-bottom corner of the simulation model window.

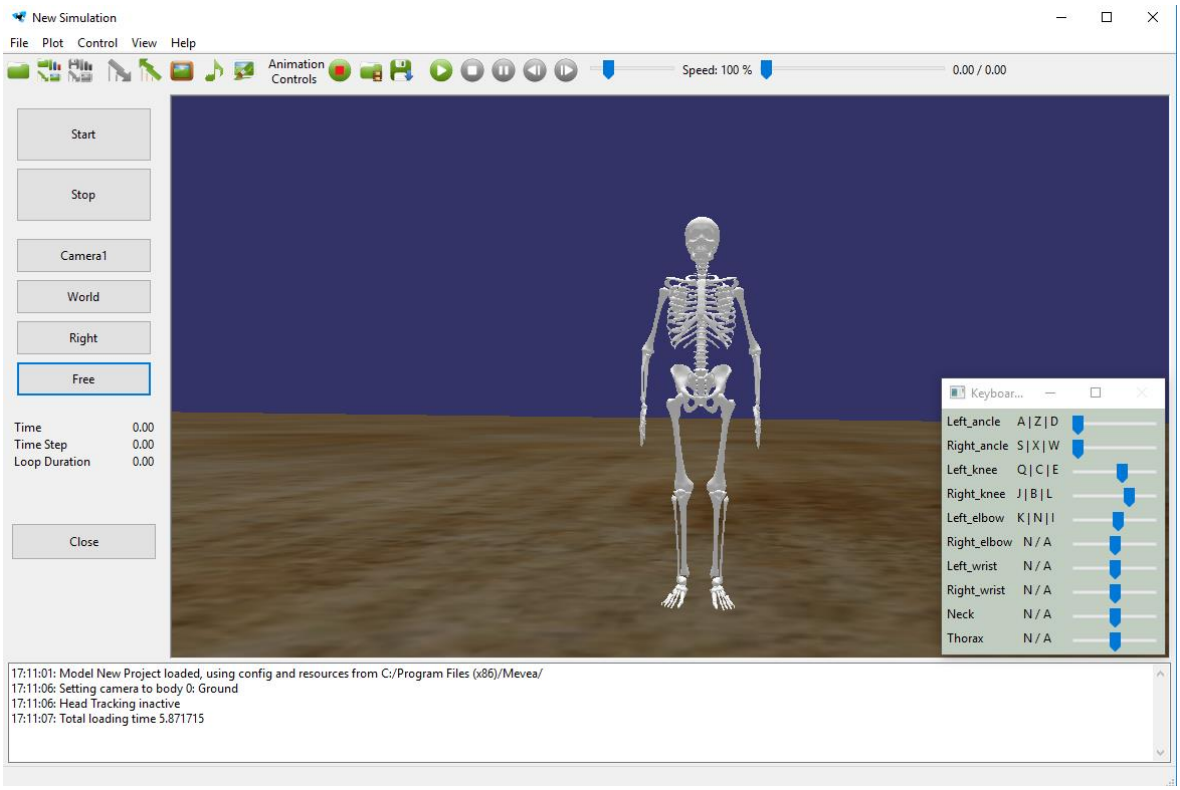


Figure 16. Mevea Solver user interface.

3.3 Kinect for Windows v2.0

Kinect for Windows v2.0 is used as a hardware for the research. Kinect for Windows v2.0 is a sensor that main function is to detect human skeletons and their tracking and send the motion data to the computer. The main advantage of the sensor is its good compatibility with Matlab and Simulink software.

Figure 17 represents Kinect for Windows v2.0. The device has RGB (Red Green Blue) color camera that allows creating color image frames, depth sensor that consists of Infrared (IR) Camera and IR Emitters and has a possibility to create depth image frames. The microphone array is used for audio data recognition. (Microsoft 2014, p. 6.)

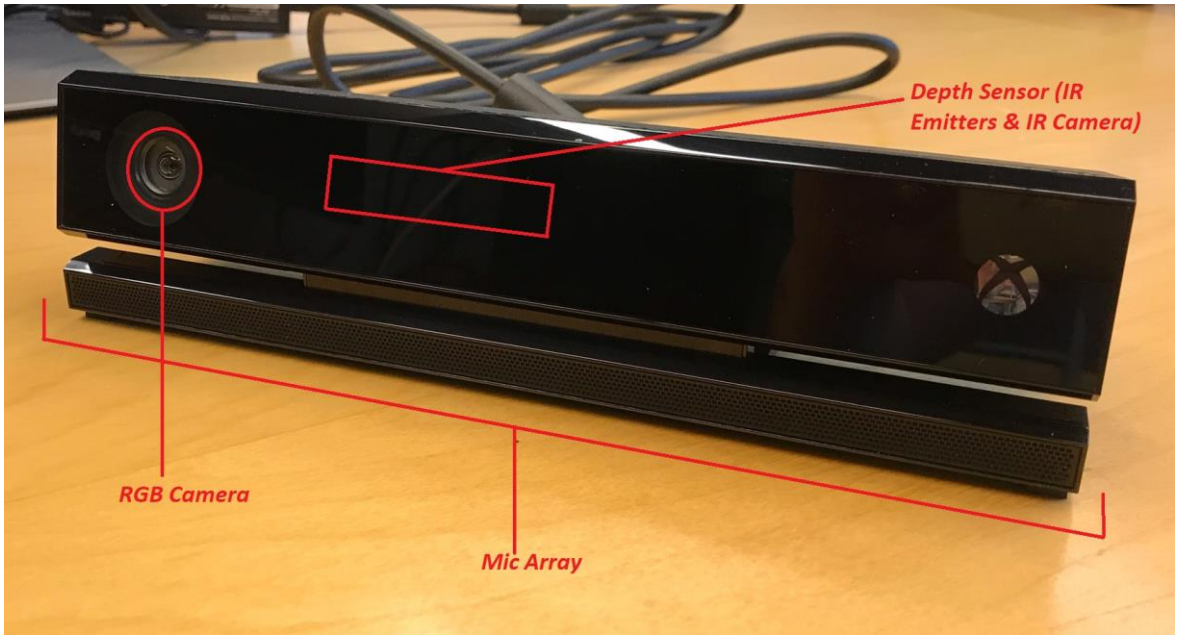


Figure 17. Kinect for Windows v2.0 sensor.

To connect the device to computer based on Windows operation system adapter for Kinect for Windows is needed. The need for an adapter is explained by the fact that the sensor itself can be directly connected only to the Xbox One and using the adapter making it possible to connect the camera to the computer.

The device is able to track up to six people on the screen at the same time. It creates a skeletal structure that consists of 25 skeletal joints. (Microsoft 2014, p. 8.) Figure 18 shows frames obtained from the depth sensor and RGB camera of the device with all 25 joints (marked with white spheres, cubes, and pyramids) and their positions in 3D space. The device is also able to understand and transmit the motion data like a position of joints in space in Cartesian coordinates to the computer. In this case, the global coordinate system origin is located exactly on the RGB Camera, and joints positions are calculated with respect to this origin.

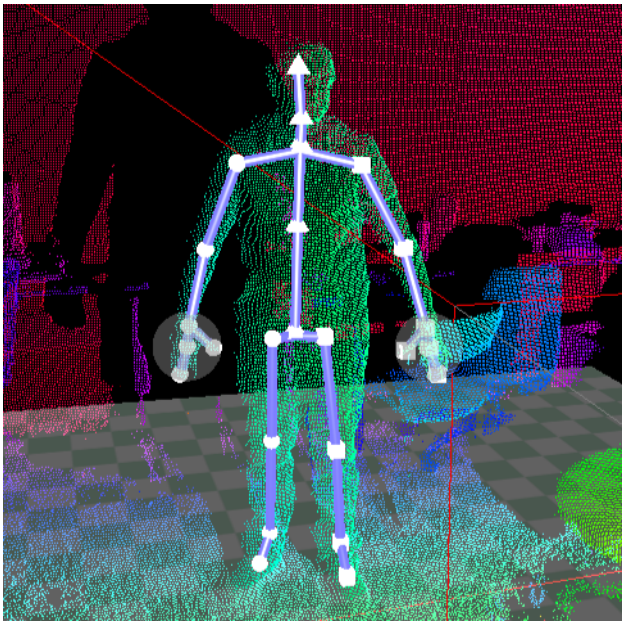


Figure 18. Skeletal joints obtained from depth sensor and RGB camera (Windows SDK 2.0)

3.4 MBS modeling process

This subchapter explains the human skeleton modeling. The skeleton modeling process consists of seven steps. First of all, graphics should be created and imported to Mevea from 3ds Max. The second step of the modeling is creating bodies and setting parameters for these bodies. The third step is associated with motion constraints creation and determination of DOF of the system. Fourth and fifth steps explain the creation of collisions of full MBS with the ground and joint torques development, respectively. The sixth step describes the creation of system inputs. The last step of the modeling process represents a detailed method of creating an external interface for Mevea in Simulink in order to control the real-time simulation through this software.

3.4.1 Obtaining graphics

Mevea software is able to support different file formats and file extensions such as .3ds, .fbx, .obj, .stl, .dae, .osg, .osgt, .osgb (Mevea Ltd. 2018c, p. 33.). Graphics in Mevea are used for two main purposes: showing a visualization of the body and the creation of contact between bodies via collision graphics. Certain file extensions can be used as visualization graphics only, and several of them can be utilized as collision graphics at the same time.

The graphic files for the project were created in Autodesk 3ds Max 3D modeling software. Autodesk 3ds Max is well-known commercial software that is used for 3D modeling, rendering and animation creation (Autodesk 2018.).

According to Mevea Reference Manual, .3ds file extension can be used for both visualization and collision graphical elements (Mevea Ltd. 2018c, p. 33.). This fact makes the graphics created in 3ds Max the best choice for the modeling. Examples of graphics created for the project in 3ds Max are shown in figure 19 (a), (b), (c), (d).

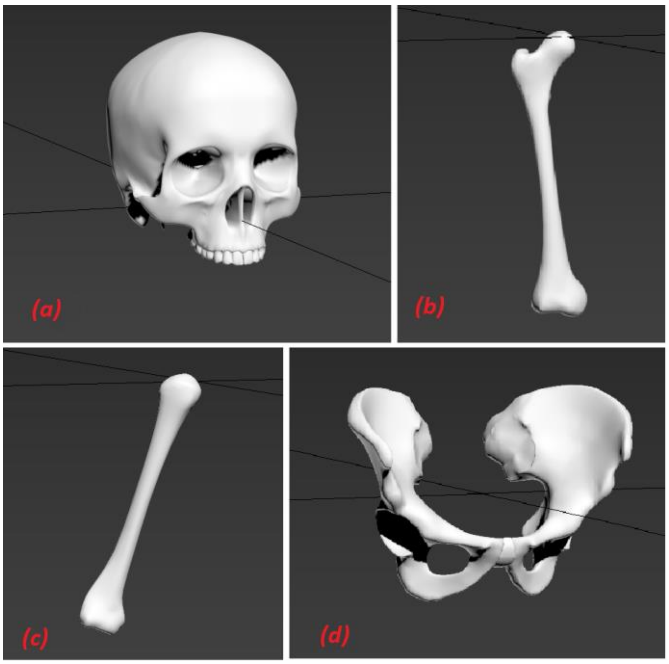


Figure 19. Graphics made in 3ds Max: (a) – Skull, (b) – Femur, (c) – Humerus, (d) – Pelvis.

Then, created graphics has to be imported to the Model Tree of Mevea Modeller software by drag-and-drop method. Generally, 135 graphic files were created and transferred to Mevea Modeller. The ground graphic file was obtained from Mevea tutorials folder.

3.4.2 Parameters of bodies

Mevea Modeller provides the possibility to define parameters of bodies inside the software. Each body should have own local coordinate system, position, and orientation with respect to the global coordinate system. The local coordinate system of the body, as well as other parameters of the body, can be set in Object View of Mevea Modeller software by clicking

on the body name. Position and orientation of each bone are set according to the anatomical structure of a human skeleton, creating kinematic chain consisting of 135 bones.

Next, the body type has to be also selected. The body can be rigid or flexible. Particular MBS is rigid, and this means that each body in the system is selected as rigid.

Then, body masses have to be calculated. Theoretically, the mass of each bone m_b can be calculated using the density and volume of the bone as is shown in equation 31:

$$\rho_b = \frac{m_b}{V_b} \quad (31)$$

where ρ_b is average density of bone and V_b is volume of bone. In practice, cortical and cancellous bones has different density due to its structure and porosity. It means that average density can be calculated by using full skeleton mass and volume. In Ch. 3.1 it is assumed that the skeleton belongs to a human with weight of 65 kg. However, it is stated that skeleton mass is approximately 17% of the human body (Malina et al. 2004, p. 121.). It means that the mass of the full skeleton is equal to:

$$0.17\ [\%] \times 65\ [\text{kg}] = 11.05\ [\text{kg}]$$

The volume of the full skeleton is obtained from Autodesk 3ds Max in the section of measurements, and it is equal to 0.00802 [m^3]. The value of volume of the skeleton is reliable due to graphics scale similar to a real person. Then, the average density is calculated by using full skeleton mass and volume. It is equal to:

$$\frac{11.05\ [\text{kg}]}{0.00802\ [\text{m}^3]} = 1377.8\ [\frac{\text{kg}}{\text{m}^3}]$$

Knowing the volume and average density of each bone, it becomes possible to calculate the mass of each bone.

Moments of Inertia are calculated according to equations 22-30 from the Ch. 2.2.4. Bones can be imaginarily presented as simple solid geometrical shapes. Then, it is possible to define a moment of inertia tensor for each bone according to the moment of inertia equations for simple geometrical shapes. After that, centers of mass of the body have to be estimated as centers of mass of solid geometrical shapes, which they are presented.

Table 4 shows all the parameters that are needed to create a body in Mevea on the example of pelvis bone. From this table, it is seen that pelvis is a rigid body. The pelvis is relative to the ground, and it means that it is the first body in the kinematic chain. The pelvis is presented as a rectangular prism with dimensions taken from Autodesk 3ds Max. Inertial properties of the pelvis are calculated according to corresponding equations for rectangular prism moments of inertia calculations. Then, table 4 shows the position of the pelvis with respect to ground (or global) coordinate system. Three Euler angles are equal to zero, and it means that there is no any rotation with respect to the ground coordinate system. Center of the mass of the pelvis is located exactly in the local coordinate system that is why three components of the center of mass are equal to zero.

Table 4. Body parameters of Pelvis bone (Mevea)

Name	Pelvis
Body type	Rigid
Relative to part	Ground
Position [m]	$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0.003 \\ 0.943 \\ 0.023 \end{bmatrix}$
Orientation [rad] 313	$\begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$
Mass [kg]	0.8418
Moments and Products of Inertia	$\mathbf{I} = \begin{bmatrix} 0.005025 & 0 & 0 \\ 0 & 0.006919 & 0 \\ 0 & 0 & 0.00887 \end{bmatrix}$
Center of Mass [m]	$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$
Use Inertia Frame	Inertia Definition in the Center of Mass

3.4.3 Joints of the system

The next step of the modeling process is associated with the creation of motion constraints for the MBS. This step is implemented by using mechanical joints. Bodies are connected between each other in Mevea Modeller software, according to an anatomical structure of the human skeleton. Totally, three types of joints are used in skeletal joints modeling. These types of joints are fixed, revolute and spherical. Table 5 shows the joints, number of constraints and DOF of the system.

The whole kinematic chain consists of 135 bodies. Thus, the MBS uses the same number of mechanical joints to connect bodies, including the connection of the skeleton with the ground. According to table 5, most of the joints are fixed, and there are only 14 moving joints in the system. All movable joints are depicted in figure 20 (marked with red circles). Hip and shoulder joints are introduced as spherical connections, the rest movable joints from figure 20 are revolute. The number of degrees of freedom of such system is 22.

Table 5. Mechanical joints of the system

Joint type	**Number of Joints**	**Constraints per Joint**	**Total number of constraints**	**DOF per Joint**	**Total number of DOF**
Revolute	10	5	50	1	10
Spherical	4	3	12	3	12
Fixed	121	6	726	0	0
TOTAL	135	-	788	-	**22**

Movable joints locations are created in such places in order to coincide with Kinect for Windows v2.0 joints. It is needed to implement the real-time simulation of the skeleton through Kinect for Windows device.

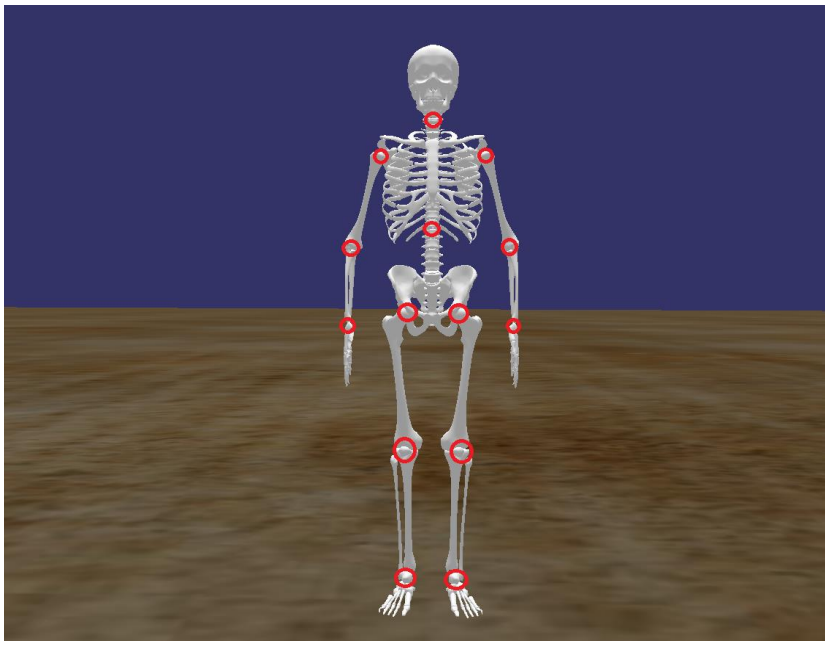


Figure 20. Locations of moving joints

3.4.4 Collision with the ground

Collision creation is the function that allows establishing contacts between the bodies in MBS. Mevea Modeller offers a tool for creating collisions inside the software. Contact with the ground is impossible, without creating collisions. Thus collisions of several bodies with the ground are created. The multibody system contains 29 collisions. Feet bones (metatarsals, tarsals, toe phalanges, calcaneus) are added to the collisions to support the skeleton and to develop a standing-balancing position. However, such bones as skull, tibias, and femurs were also added to the collisions in order to provide contact with the ground without falling through it, when simulation runs. Collisions are made by implementing graphics of bones as a collision or contact graphics. Table 6 shows an example of the collision parameters between left calcaneus and ground made in Mevea.

Table 6. The collision between ground and left calcaneus (Mevea)

Name	Coll_Ground_L_foot_calcaneus
Body A	Ground
Graphics A	Ground_Graphics
Body B	L_foot_calcaneus
Graphics B	L_foot_calcaneus_Graphics
Spring constant	2e5

Table 6 continues. The collision between ground and left calcaneus (Mevea)

Damping coefficient	1e5
Restitution coefficient	0.1

Two bodies with two collision graphics are used to set the collision parameters. Spring constant, damping and restitution coefficients are used to describe the collision impact of two bodies. In this case, these coefficients and constants are adjusted according to structure behavior. A selection is made according to a smooth collision of bone with the ground. The friction of collisions is neglected in order to create an almost ideal situation. All the constants and coefficients are similar for each of 29 collisions made in Mevea Modeller.

3.4.5 Torques

One of the important steps in this project is the creation of torques in moving joints. Torques are implemented in Mevea Modeller by using a rotational spring-damper-torque actuator (Mevea Ltd. 2018c, p. 44.). Equation 32 shows the total torque T_a which can be produced by rotational actuator n:

$$T_a = K_n \times (\theta_n - \theta_{0n}) + d_n\dot{\theta}_n + T_n + T_{fn} \quad (32)$$

where K_n is the spring constant, θ_n and θ_{0n} are actual and initial angular positions in radians, respectively, d_n is damping constant, $\dot{\theta}_n$ is the first derivative of actual angular position (angular velocity), T_n is the constant torque applied to the actuator and T_{fn} is the friction torque (Mevea Ltd. 2018c, p. 44.).

Table 7 illustrates the main parameters of the torque actuator in Mevea Modeller software on the left elbow joint. First, two jointed bodies have to be selected, where these bodies are left humerus and left ulna. Then, torque directions have to be chosen via direction vectors selection. Direction vector **f** is a vector perpendicular to torque axis which defined in the local coordinate system of body 1 or body 2. Direction vector **g** is the vector perpendicular to both torque axis and vector **f** where it can be defined in the local coordinate system of body 1. (Mevea Ltd. 2018c, p. 44.) According to these vectors, direction of torque is selected.

Table 7. Example of torque parameters in left elbow joint (Mevea)

Name	T_L_elbow
Body 1	L_arm_humerus
Direction vector f, body 1	$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$
Direction vector g, body 1	$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$
Body 2	L_arm_ulna
Direction vector f, body 2	$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$
Spring constant [Nm/rad]	3.462
Damping constant [Nms/rad]	0.6924
End damper spring constant [Nm/rad]	3.462
End damper damping constant [Nms/rad]	0.6924
Spring minimum angle [rad]	-1
Spring maximum angle [rad]	2
Spring initial angle [rad]	0
Constant torque [Nm]	0

Then, spring and damping constants have to be chosen by adjusting. The main goal of adjusting is to select such spring and damper constants that allow the smooth and fast response of the joint movement. End damper spring and damper constants have to be set similar to general spring and damping constants. Spring minimum and maximum angles set the restrictions of movement. Restrictions are approximated according to the anatomical ability of joint to move. Spring initial angle sets the initial angular position for the joint. Then, constant torque adds additional torque to the torque generated in the actuator. In a particular case, constant torque is equal to zero.

Totally, ten torques are applied to the MBS. It means that one torque is applied to every revolute joint. Thus, every joint has own torque parameters. Parameters of every joint torque are shown in details in Appendix II.

3.4.6 Inputs of the system

Next step of the modeling process is the creation of inputs for torques in order to control them. Type of the control signal depends on input type. Input type can be analog or digital. Type of signal also depends on the Primitive type that describes the input variable. In case of torque control, the Primitive type can be Initial spring angle, Constant torque or Constant angular velocity. (Mevea Ltd. 2018c, pp. 146-147.) Initial spring angle is chosen as an input for torques in this project, due to its direct similarity to the joint angle, which is used for position control. Adjusting of joint angle allows controlling the torque produced by rotational torque spring-damper actuator according to equation 32 from Ch. 3.4.5.

The output value of the analog input signal is strictly dependent on the input value from the controller (Mevea Ltd. 2018c, p. 148.). The final value of the signal y can be calculated using equation 33:

$$y = \begin{cases} B, & |x| \leq D_z \\ \dfrac{A(x - D_z)}{1 - D_z} + B, & |x| > D_z, x > 0 \\ \dfrac{A(x + D_z)}{1 - D_z} + B, & |x| > D_z, x < 0 \end{cases} \quad (33)$$

where A is scale coefficient of the input, B is offset of the input, D_z is a dead zone of the input and x is a raw value of the input from the controller (Mevea Ltd. 2018c, p. 148.). In particular case, signal value y is equal to Initial spring angle of actuator n due to selection of initial spring angle as a Primitive type for input (equation 34):

$$y = \theta_{0n} \quad (34)$$

Ten inputs are used to provide control for each torque. Table 8 shows an example of such input based on Initial spring angle related to the torque acting in left elbow joint. From the table, it is seen that primitive type is B2BM or torque and a primitive sub is Initial spring angle in radians. The torque that is used by input (T_L_elbow) is the torque operating in left elbow joint. Input type is analog. Scale, offset, and dead zone of the input are set as default. All ten inputs are based on the same parameters. However, there are several differences. For

different inputs, different torques have to be assigned, and I/O channel numbers for joystick or keyboard control are varied from 0 to 9.

Table 8. Example of input parameters for torque acting in left elbow (Mevea)

Name	IN_Left_elbow
Input type	Analog
Primitive type	B2BM (Torque)
Primitive name	T_L_elbow
Primitive sub	Initial spring angle [rad]
Scale	1
Offset	0
Dead zone	0.25

3.4.7 Simulink external interface

The last step of the multibody modeling is the development of Simulink external interface for the system control. The connection of the Simulink and Mevea simulation software have to be implemented by using TCP/IP (Transmission Control Protocol/Internet Protocol) socket interface. First, a socket interface is created in Mevea Modeller. IP address and port number are automatically generated while creating. Table 9 presents the parameters of the created socket interface.

Table 9. Parameters of the socket interface (Mevea)

Name	Simulink_External_Interface
IP Adress	127.0.0.1
Port Number	5110

Next, socket signals have to be added. Totally, 20 socket signals are added: ten signals for inputs and ten signals for outputs. Ten inputs that were created earlier are added to the input socket signals and are set as input initial angle of the joint. Ten outputs are obtained from data sources. Totally, ten data sources are created in Mevea Modeller in order to output the torque obtained from the joints. Table 10 shows an example of data source and parameters of it. All data sources have to be made similar, with the only difference that is the torque that has to be shown as output.

Table 10. Torque Data Source example (Mevea)

Name	OUT_DATA_L_elbow
Component Type	B2BM (Torque)
Component Name	T_L_elbow
Variable Name	T_a (Total torque of the actuator)

Then, the Simulink external interface was received from Mevea company. The interface consists of three C++ based Simulink s-functions: MServerInitializer, MSendingData, and MReceivingData. These functions allow managing signals in real-time simulation. The corresponding Simulink external interface circuit can be found in Appendix III.

The function of MServerInitializer is to transmit the signal through a TCP/IP socket interface and to provide a connection between Mevea and Simulink. Function parameters of this s-function are a port number similar to the port number from table 9 and simulation time step that is chosen as 0.001. MSendingData function is to send the data from Simulink to Mevea. The function has twelve inputs. Two of them are used for connection with MServerInitializer s-function. In this case, initial angles of the joints are presented as sending data in another ten inputs of the function. MReceivingData function is to receive the processed data from Mevea. This function has two inputs that are signals from MServerInitializer. The function also has ten outputs that are output socket signals from Mevea based on the torque data sources.

To sum up, the principal function of the external interface is to send the initial angle of the joints to the Mevea in order to control the motion of the joints. Torques are calculated inside the Mevea software as it is shown in equation 32 in Ch. 3.4.5. After that, torque values are sent back to the Simulink through output socket signals. The main purpose of sending torque values back to Simulink is the visualization of torque changes in real-time while applying different initial angles of the joints.

3.5 Hardware and software connection

The connection between Kinect for Windows sensor and Mevea simulation software is established via Matlab Simulink. The compatibility of the Kinect sensor with Matlab

Simulink is the key reason for using this software. The goal of the connection is to make possible the conversion of joints coordinates from the Kinect sensor to the angular position of the joints, which are the basis of inputs from the Simulink external interface to Mevea.

The communication of Kinect for Windows v2.0 and Matlab Simulink is provided via Matlab Image Acquisition Toolbox. Image Acquisition Toolbox is a toolbox that is used for hardware that has a video input. It provides additional Matlab functions and Simulink blocks for convenient image and data acquisition. (Mathworks 2018a.) The toolbox can be applied with Kinect for Windows v2.0 device and Matlab version 2016a and later (Mathworks 2018b.).

The direct connection between Kinect and Simulink was developed with the help of run-time s-function downloaded from GitHub open source (GitHub 2016.). S-function was based on C++ code. The compilation of C++ code with Matlab was done, and function started to work. This function allowed acquiring the motion data from the Kinect sensor and sent it to the Simulink and Matlab in real-time.

The skeletal joints positions in Cartesian coordinates are obtained as a 25x4 matrix, where a number of columns are a number of skeletal joints, which were recognized by the Kinect sensor. A number of rows are x, y and z coordinates of the body and joint tracking state correspondingly. Joint tracking state can be defined by four numbers. Zero is used to show that the body is not tracked. One is used to determine if the body is inferred. Two is used to show that the body is tracked. Minus one value means that Kinect cannot recognize the full body. (GitHub 2016.)

The 25x4 matrix of joints coordinates is transformed into a vector and divided into 25 3x1 matrices with the help of Demux Simulink block inside the subsystem. Figure 21 shows the full circuit of Kinect skeletal data acquisition model in Simulink.

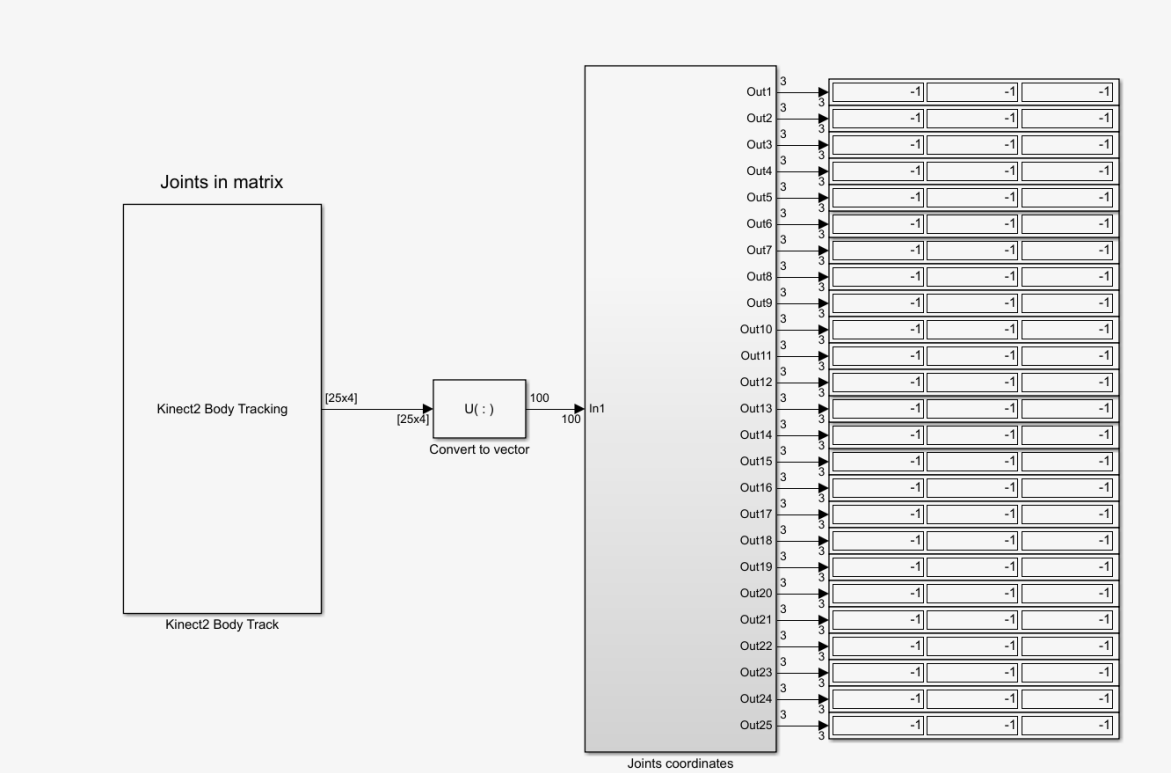


Figure 21. Simulink Real-time skeleton tracking data acquisition

The visualization of a human skeleton with its joints positions was implemented by using Matlab code. The code was downloaded from the Matlab File Exchange open source (Mathworks 2018c.). The authors of the code are Juan R. Terven and Diana M. Cordova-Esparza (Terven J. and Cordova-Esparza 2016.). The code is partially based on C++ programming language and partially on Matlab script.

The compilation of C++ files was done in order to make the Matlab code to work. The purpose of the code is to show skeletal joints in Matlab. The additional purpose is a visualization of joint coordinates data changing in real-time in Matlab as well as in Simulink. The code can be found in Appendix IV. Figure 22 depicts an image generated by the Matlab code. The image was obtained from a depth sensor of Kinect for Windows v2.0 and shows all the skeletal joints positions obtained from the device. Joints are marked with red circles.

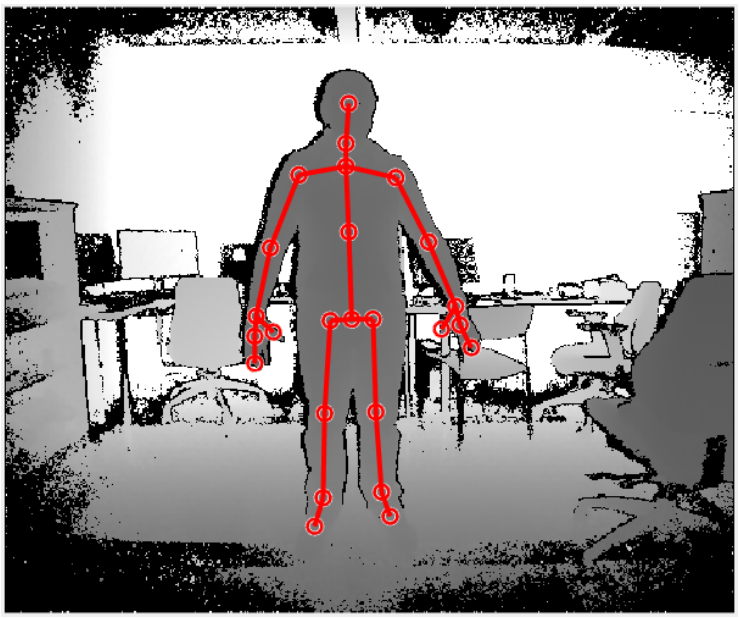


Figure 22. Frames obtained from depth sensor with Kinect body joints (Matlab)

4 RESULTS AND ANALYSIS

The present chapter contains a description of the last stage of the research which is testing of the system and results obtaining. The section also includes analysis of the results. The experiment is divided into three case studies: elbow joint movement, thoracic joint movement, and full body test.

The purpose of experiments was to test the movement in several skeletal joints: elbow and thoracic and also to test the whole body. It was an important issue to observe the behavior of torques in the real-time simulation during angular position changes. The purpose of the third experiment was to review the torques and angular positions of the joints in the whole skeleton model while the Kinect sensor is connected to the system and human moves in front of the camera.

4.1 Case study 1: Elbow joint.

The first case study was directed to test and analyze left elbow joint motions during real-time simulation of the model. For this purpose, the spherical joint of the shoulder was completely fixed. Four tests were studied. The goal of the first three tests was to illustrate the configuration and torques of the model at different constant angular positions. The goal of the fourth test was to check the torque changes at the variable angular position.

The first test was made at the angular position of 0 radians, which is the initial position of the elbow joint. Figure 23 shows the orientation of the elbow joint (marked with red circle) and the plot of angular position in degrees, while the angular position value inputs through the Simulink external interface. At the start of the simulation, the angle shifts slightly from 0 to -1.66 degrees, creating an insignificant error. After 0.5 seconds of the simulation, the position becomes constant.

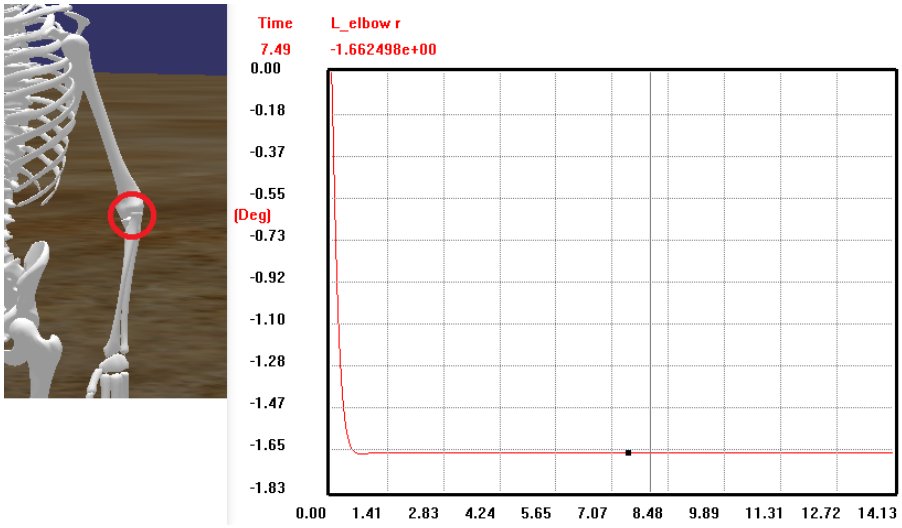


Figure 23. Response and orientation of the left elbow joint at an Initial angle of 0 rad.

Figure 24 depicts the graph of torque at an angular position of 0 radians. On the graph, it is observed that the torque goes down and generates a significant overshoot. On 0.5 second of the simulation, torque takes a constant value of -0.1 Nm. This value of the torque is needed to support the initial angle of the elbow at full body standing-balancing position.

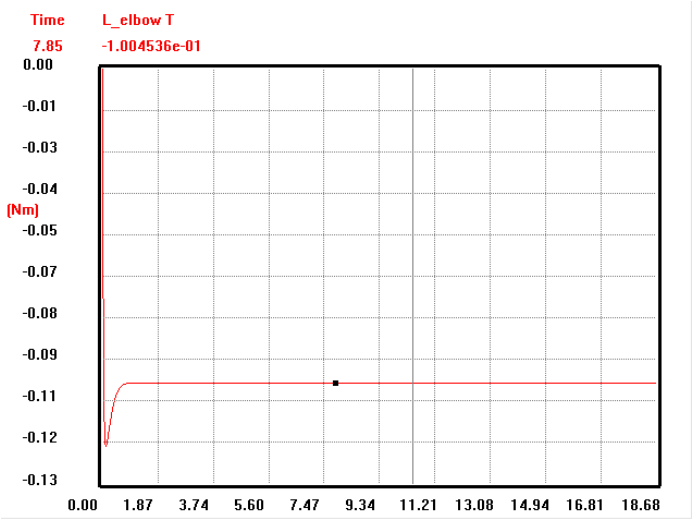


Figure 24. Supporting elbow torque at an Initial angle of 0 rad.

The second test was done to check the orientation of the elbow joint at the angular position of 1 radian. The position was set before the simulation had been started. Figure 25 illustrates the orientation of the elbow (marked with red circle) and the angular position graph, showing the behavior of the joint. The rise time of the function is approximately 0.6 seconds. Theoretically, the angle has to be approximately 57 degrees due to the input value of 1 radian.

However, after 0.6 seconds of the simulation, the value of position becomes constant and equals 49 degrees.

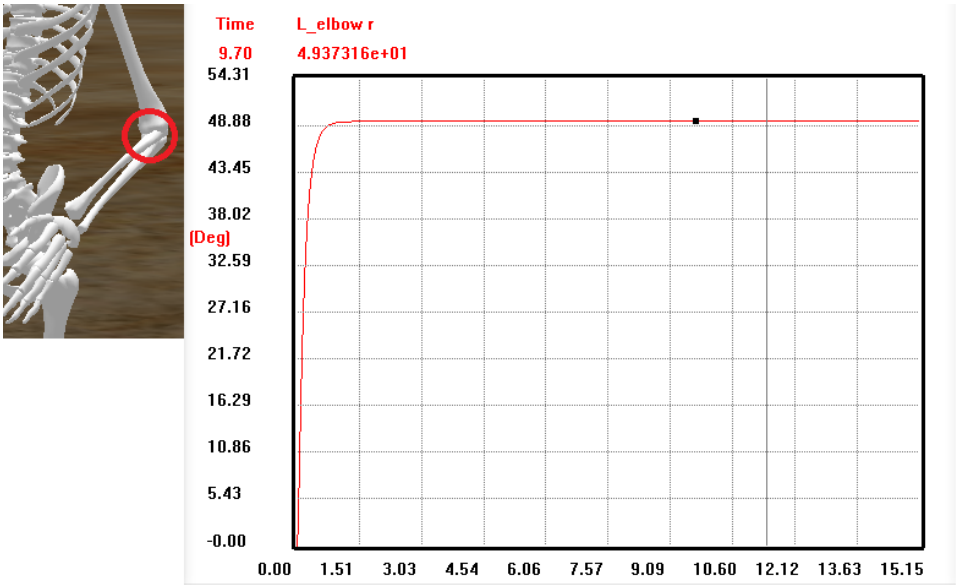


Figure 25. Response and orientation of the left elbow joint at an Actual angle of 1 rad.

Figure 26 depicts the torque plot at an angular position of 1 radian. At the beginning of the simulation, the torque fluctuates, creating high overshoot. After 0.5 seconds of simulation, the torque becomes constant and equals -0.4787 Nm.

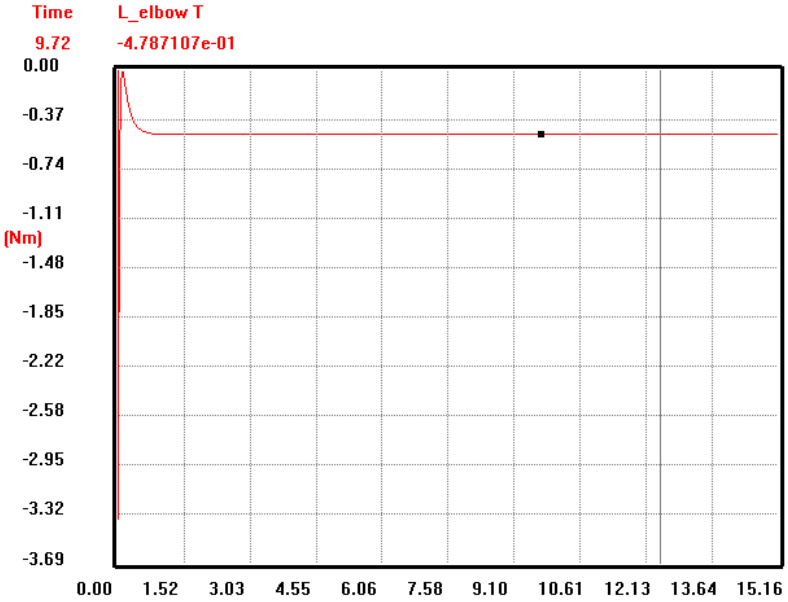


Figure 26. Supporting elbow torque at an Actual angle of 1 rad.

The angular position of 1.57 radians (90 degrees) was implemented in the third test. The position was set before the simulation has been started. Figure 27 illustrates the orientation of the joint as well as angular position graph. According to the graph, the position becomes constant after 1 second from the beginning of the simulation. The angle is equal to 92 degrees, and it means that the value of error of the input is relatively low.

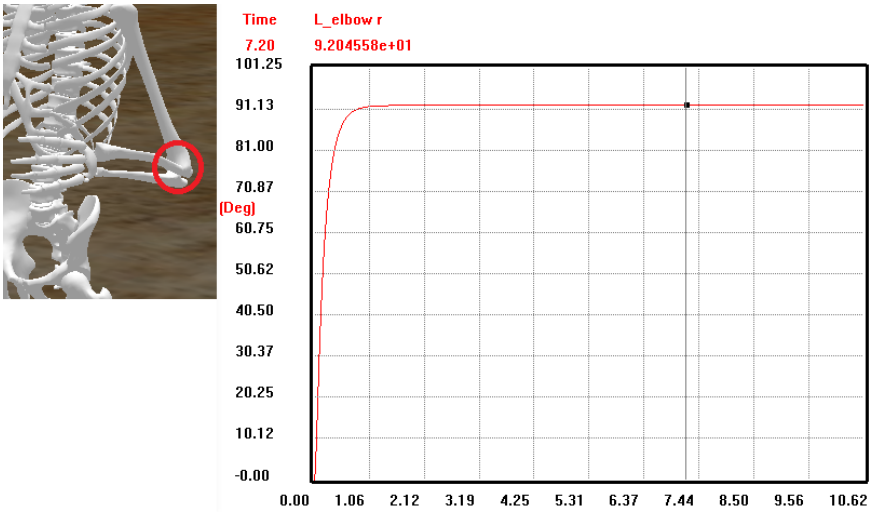


Figure 27. Response and orientation of the left elbow joint at an Actual angle of 1.57 rad.

Figure 28 shows the graph of torque acting in elbow joint at an angular position of 1.57 radians. According to the graph, there is a similar problem as it was in the second test. The signal generates large overshoot at the beginning of the simulation, and after 0.5 seconds of simulation, the torque becomes constant and equals -0.5314 Nm.

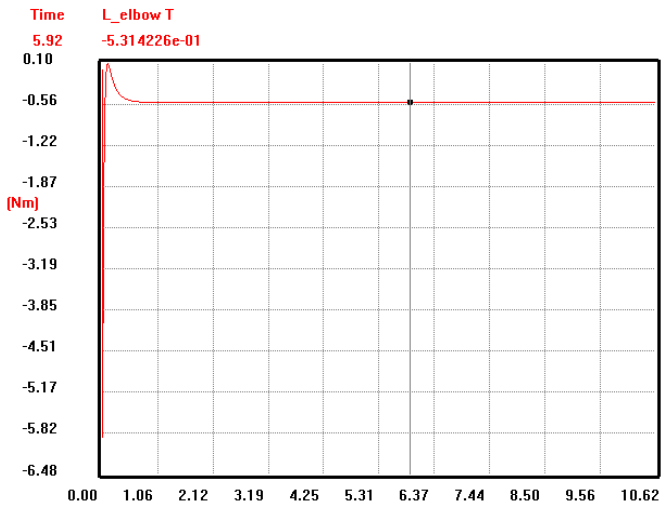


Figure 28. Supporting elbow torque at an Actual angle of 1.57 rad.

The general purpose of the fourth test was to explain the structure behavior during variable motion implementation. The first objective of the test was the observation of the torque in elbow joint during a continuous and variable signal. The second objective was to check torque changes in dependent wrist joint while the position of elbow joint was changing.

The angular position spline was created for this purpose. The spline was applied via Simulink Signal builder block. Signal duration was 10 seconds. The angular position of the elbow joint had been constantly changing throughout this time.

Figure 29 depicts angular position and torque graphs obtained from the simulation. There are two peaks on the position graph. Torque changes similarly to the position of the elbow with small fluctuations. Torque fluctuations are marked in the torque graph with blue circles. However, the system is stable.

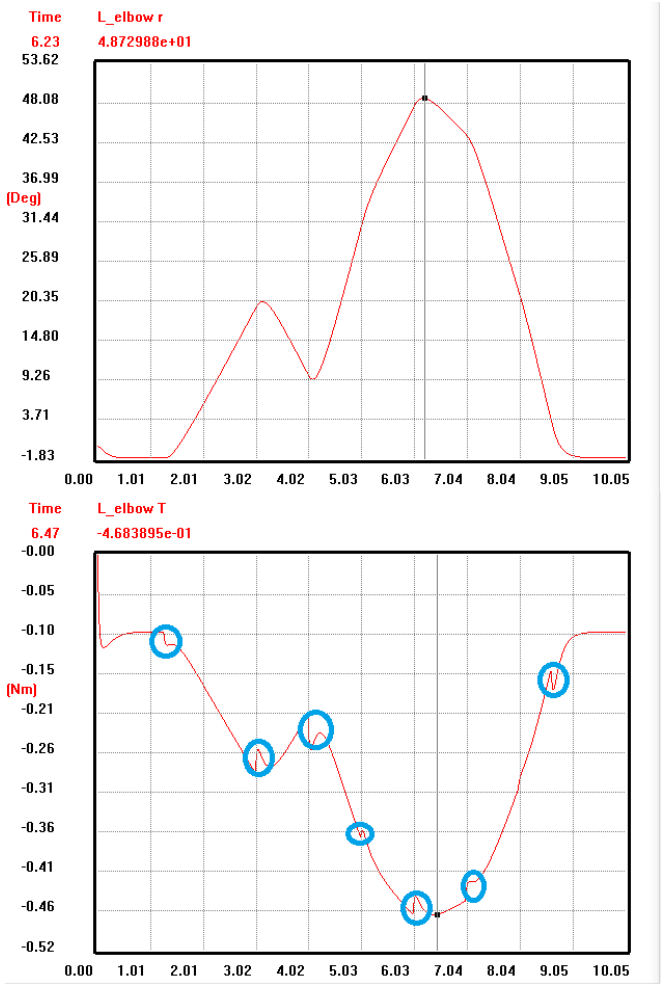


Figure 29. Angular position and torque of elbow joint at variable position input.

Figure 30 depicts a graph of the torque in the wrist joint. The wrist torque changes simultaneously with the position of the elbow. There are similar fluctuations of the torque in the graph. Fluctuations of the torque are marked with yellow circles. The magnitude of the torque at the highest angular position of the elbow joint is equal to 0.0003729 Nm. A change in wrist torque confirms that all joints in the kinematic chain are dependent on each other, and torques change to support the balance of the whole structure.

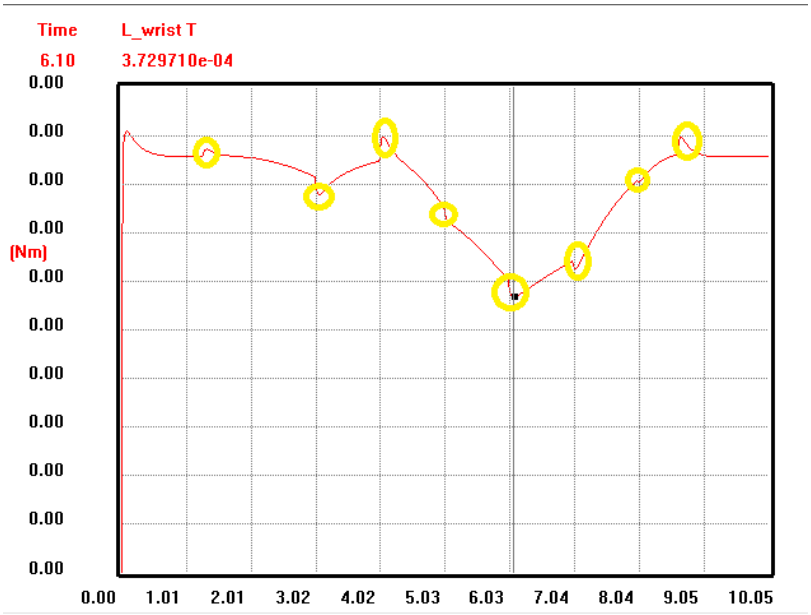


Figure 30. Torque in wrist joint during elbow joint movement at variable position input.

4.1.1 Analysis of the case study 1

First three tests have similar problems. The main problem is the relatively significant difference between the input value of the angular position and the actual position. The solution of this problem might be the tuning of the offset, scale, and dead zone values of the input in Mevea Modeller to obtain the most precise response. The second problem is overshooting of torque at the beginning of the simulation. This problem could be solved by adjusting spring and damping constants in Mevea Modeller.

The fourth test shows that the main problem is elbow and wrist torque fluctuations at the moments of position changes. Adjustment of spring and damping constants could be the solution to that problem in order to obtain smooth movements. The next problem is a low amplitude of wrist joint torque. It can be associated with the different rotation axis of the wrist joint, and it means that elbow influence on the wrist torque is not significant.

4.2 Case study 2: Thoracic joint

The second case purpose was to check the movement of thoracic joint at exact angular positions of the joint. The case study used similar tests that were done for the first case study. First, three independent tests of torques were done at constant angular positions of 0, -0.5 and 0.5 radians. Secondly, the continuous variable signal was applied to the angular position to observe the torques of thoracic and neck joints.

The first test was made at constant angular position input of 0 radians. It means, that the joint is located in the initial position. Figure 31 illustrates the orientation of the joint (marked with red circle) and the graph of angular position. The graph shows that there is a relatively low value of error of angular position compared to the input value of angle. The input value is 0 radian, and the actual value is -0.63 degree.

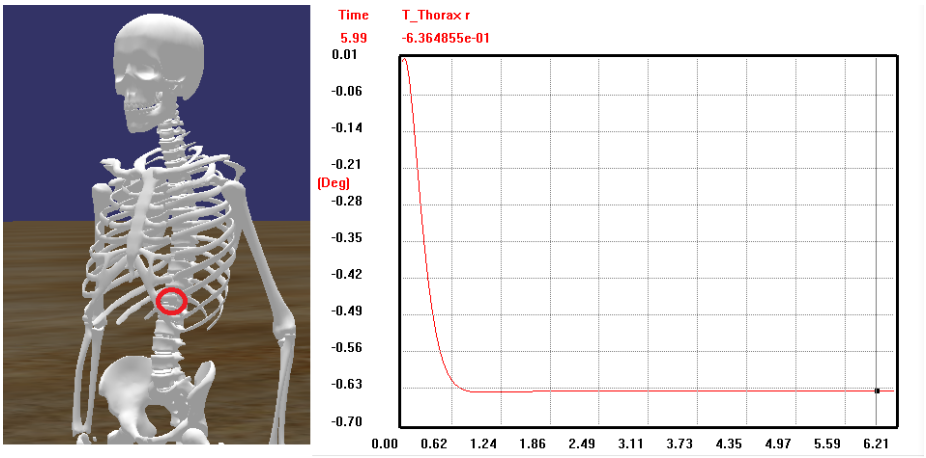


Figure 31. Response and orientation of the thoracic joint at an Initial angle of 0 rad.

Figure 32 illustrates the plot of the torque at the initial angular position of 0 radians. Torque magnitude is equal to 0.6409 Nm. This value is needed to support the joint initial balancing position. At the beginning of the simulation, there is an insignificant torque overshoot, which is replaced by constant torque value after 0.8 seconds of simulation.

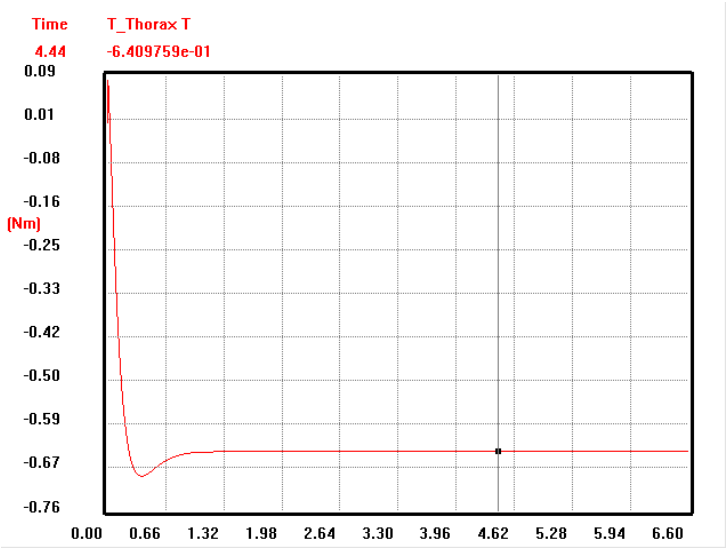


Figure 32. Supporting thoracic torque at an Initial angle of 0 rad.

The second test was done to check the torque and orientation of the joint at the angular position of -0.5 radians. Figure 33 depicts the orientation of the upper part of the skeleton at this angular position and shows behavior of the joint (marked with red circle). Figure 33 also illustrates the graph of angular position. The position becomes constant approximately at 0.8 second of simulation. The angular position equals -27.3 degrees, and that value is rather close to the input angular position value of -0.5 radian.

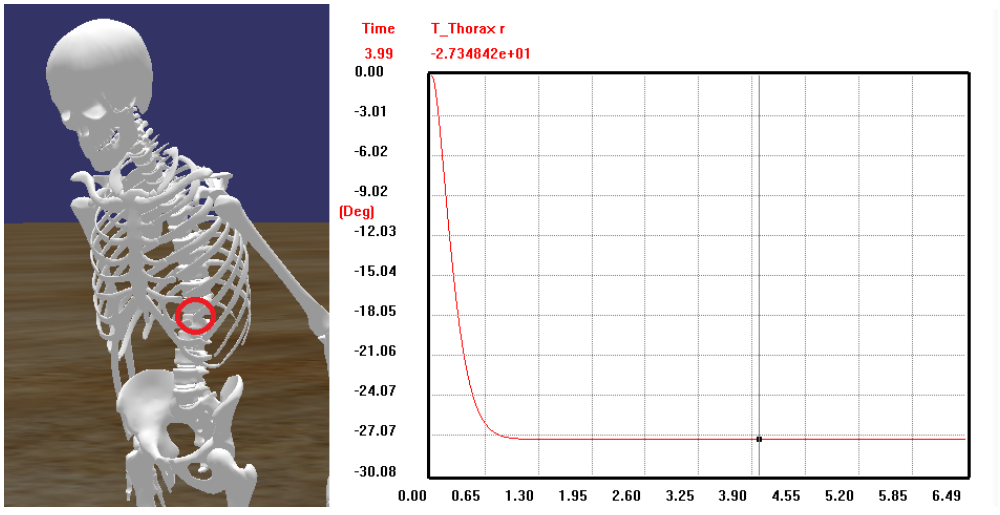


Figure 33. Response and orientation of the thoracic joint at an Actual angle of -0.5 rad.

Figure 34 shows plot of torque. The torque is measured at the angle of -0.5 radian, and it is equal to -8.3 Nm. The torque fluctuates at the beginning of the simulation. After 0.9 seconds

from the start of the simulation, the torque becomes constant. The system has small overshoot, but overall it is stable.

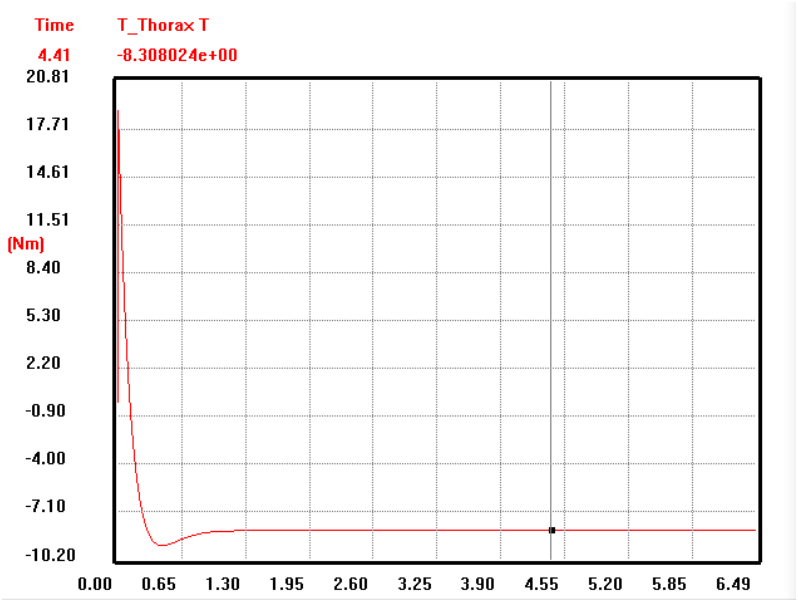


Figure 34. Supporting thoracic torque at the Actual angle of -0.5 rad.

The third test was done to check the torque at the angular position of the joint that is equal to 0.5 radians. The orientation and position plot of the joint is depicted in figure 35. The graph describes the angular position behavior. The position becomes constant after 0.9 seconds from the beginning of the simulation. The position is equal to 26 degrees whereas the input value of the angular position is 0.5 radian or 28.6 degrees. It means that there is a low value of error.

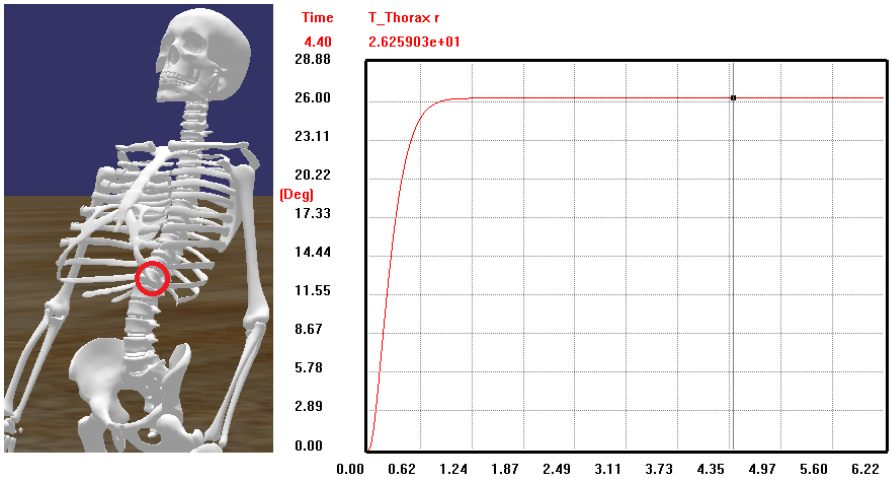


Figure 35. Response and orientation of the thoracic joint at the Actual angle of 0.5 rad.

Figure 36 depicts a graph that describes torque behavior at an angular position of 0.5 radians. The problem of torque fluctuating at the beginning of simulation is still actual. Torque becomes constant at 1 second of simulation, and it equals 7.2 Nm. The system is stable.

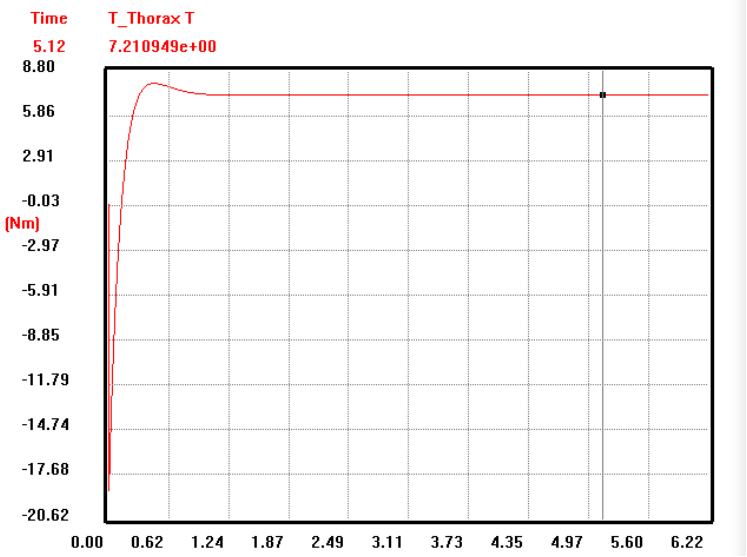


Figure 36. Supporting thoracic torque at the Actual angle of 0.5 rad.

The fourth test was associated with examination of the torque in the thoracic joint at continuous and variable angular position input. This test used the similar approach that was used in case study 1. The angular position spline was created in Simulink via Signal builder block. The time for the signal was set as ten seconds.

Figure 37 illustrates two graphs, which show thoracic angular position and torque. The position graph shows that there are two peaks of the angular position. However, the angle does not exceed 25 degrees as also can be seen from the graph.

The behavior of the torque from the graph is strongly dependent on angular position due to the similarity of these two graphs. The torque acting in the thoracic joint has more peaks and drops than position. Peaks are marked with green circles.

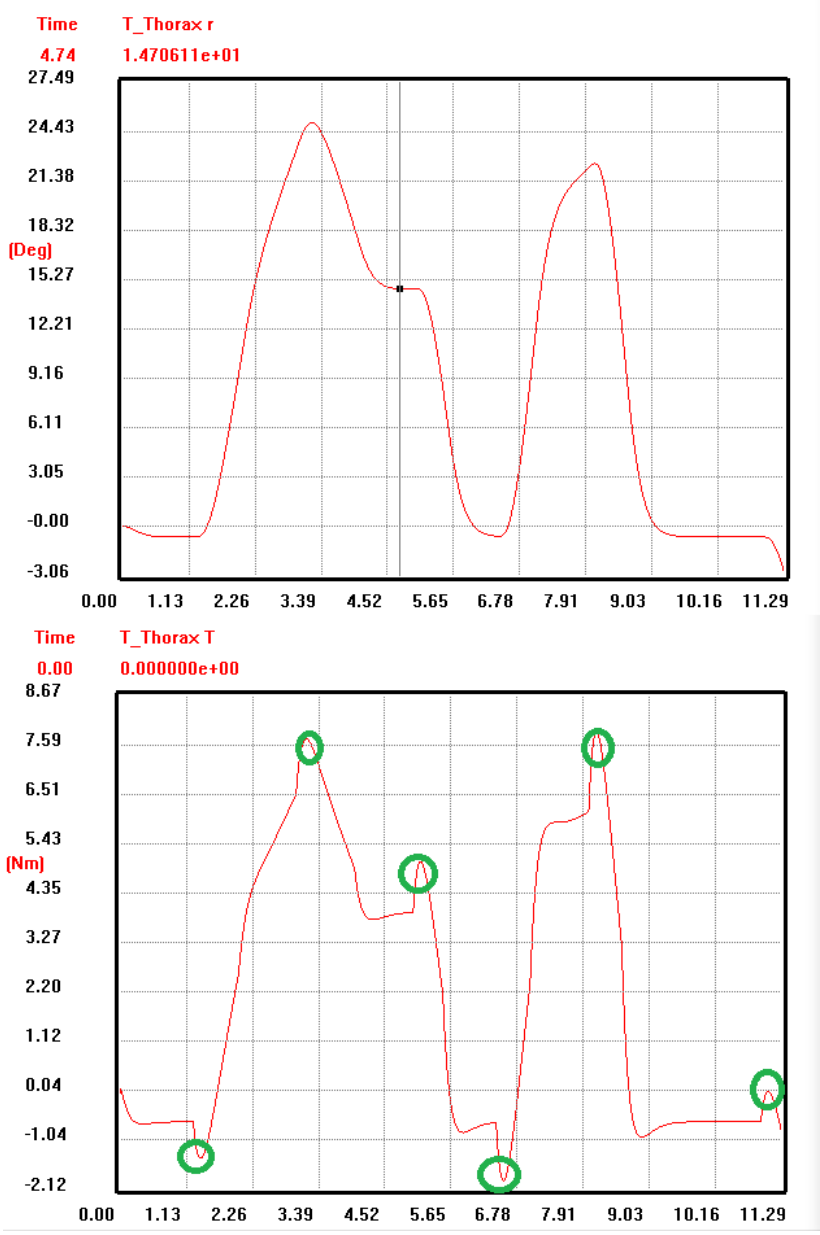


Figure 37. Angular position and torque of thoracic joint at variable position input.

The neck joint torque depends on a thoracic joint orientation. While the angular position of the thoracic joint changes, the torque of the neck joint also changes to support the head while the torso is tilted. Figure 38 depicts the graph which describes the torque of the neck joint. From the graph is seen that the torque of neck joint behaves similarly to the torque of the thoracic joint. The graph has similar peaks and drops (marked with orange circles), the only difference is the magnitude of the torque. Neck joint needs less torque to support the head due to lower mass of the head, whereas thoracic joint supports the whole upper part of the body, including the head.

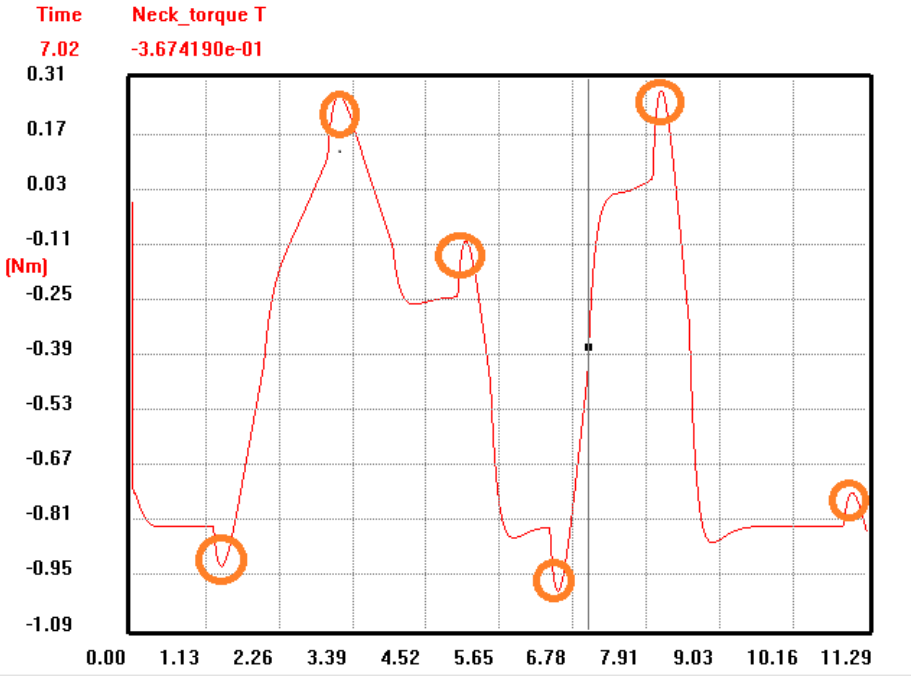


Figure 38. Torque in the neck joint during elbow joint movement at variable position input.

4.2.1 Analysis of the case study 2

First three tests show that there is the same problem as was in the first case study. This problem is the presence of errors of output angular position. In the present case study, the value of the errors is lower compared to the first case study. Tuning of scale, offset, and dead zone of the input can bring in serious contribution to the solution of this problem. The second problem is the fluctuating torque at the beginning of the simulation. It means that the parameters of the joint need an adjustment.

The last test shows that the system is stable and there are no torque fluctuations at the variable angular position. It means that spring and damping constants are adjusted reliably, but the system still needs a little tuning.

4.3 Case study 3: Full body

The third experiment was carried out in office room, located in the Lappeenranta University of Technology. The main goal of the third experiment was to test the whole model in real time using Kinect for Windows sensor. One person participated in the experiment. The person is male at the age of 24 and a height of 165 cm.

First, the Kinect sensor recognized the body of the participant of experiment. The Matlab code generated the video output and showed the person's skeletal joints. Figure 39 illustrates the person standing in front of the camera and the main skeletal joints recognized by the Kinect sensor.

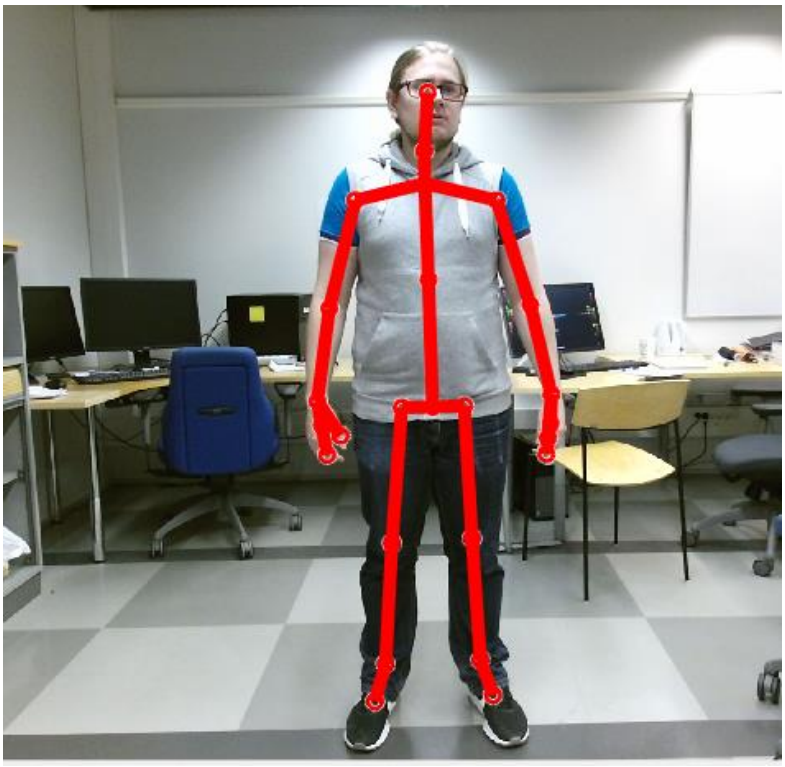


Figure 39. Joints from Kinect sensor (RGB camera).

Then, the person made a movement. Obtained skeletal data was transferred from the depth sensor of Kinect directly to the Simulink in real time. Figure 40 shows the position of the joints in coordinates.

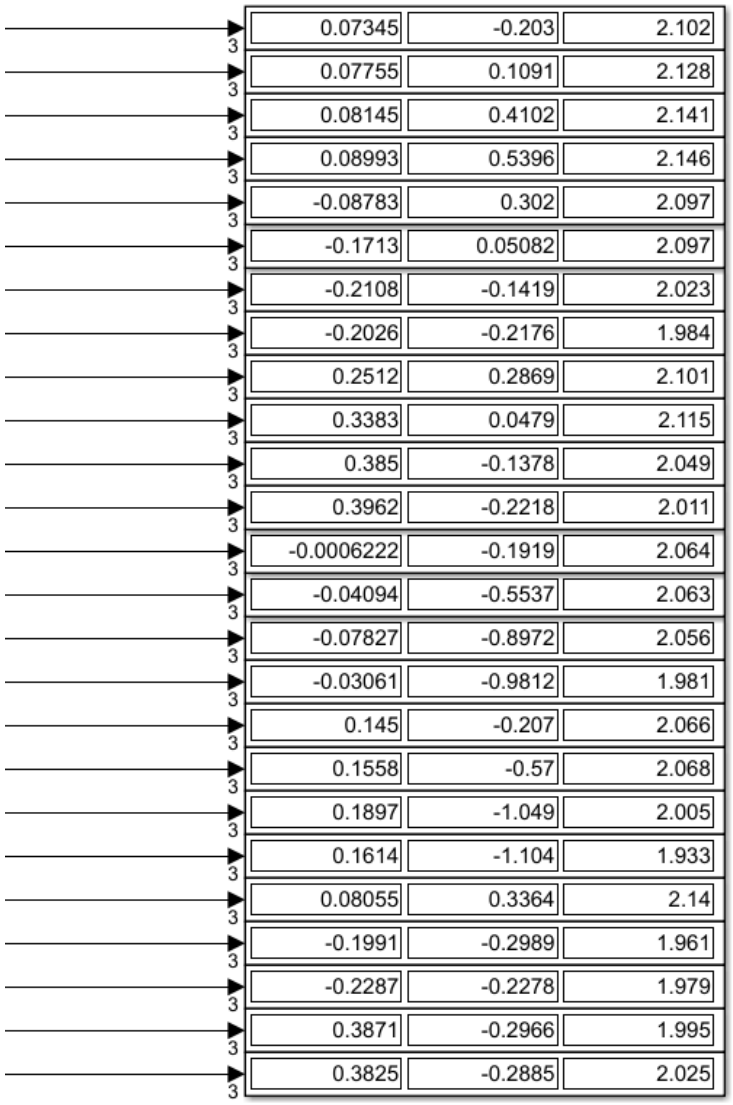


Figure 40. Coordinates of joints in 3D space

Next, spatial coordinates of ten main joints were automatically converted to the angular positions of the joints inside the Simulink. The change of angular positions caused the movement of the skeleton model. Figure 41 illustrates the response of the skeleton according to changes in angular positions of the joints. Figure 41 (a) shows angular positions of the joints, and figure 41 (b) depicts the model configuration according to the person movement.

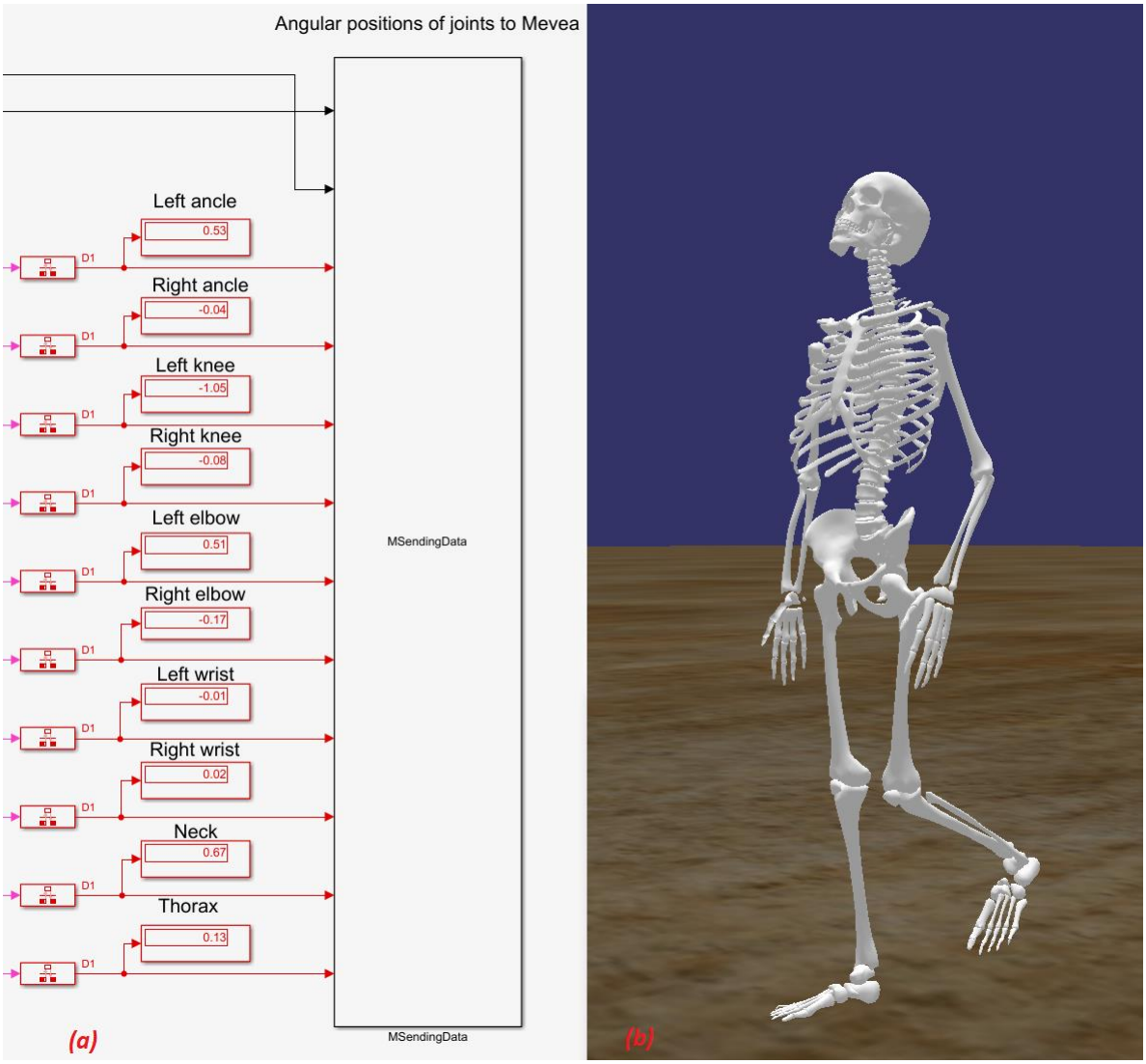


Figure 41. Response of the model: (a) - Angular positions of the joints (b) - The configuration of the model controlled via Kinect sensor.

The corresponding torques were automatically calculated inside the Mevea simulation software. Afterwards, torques were sent back to Simulink external interface. Figure 42 illustrates the values of torques obtained from Mevea simulation software.

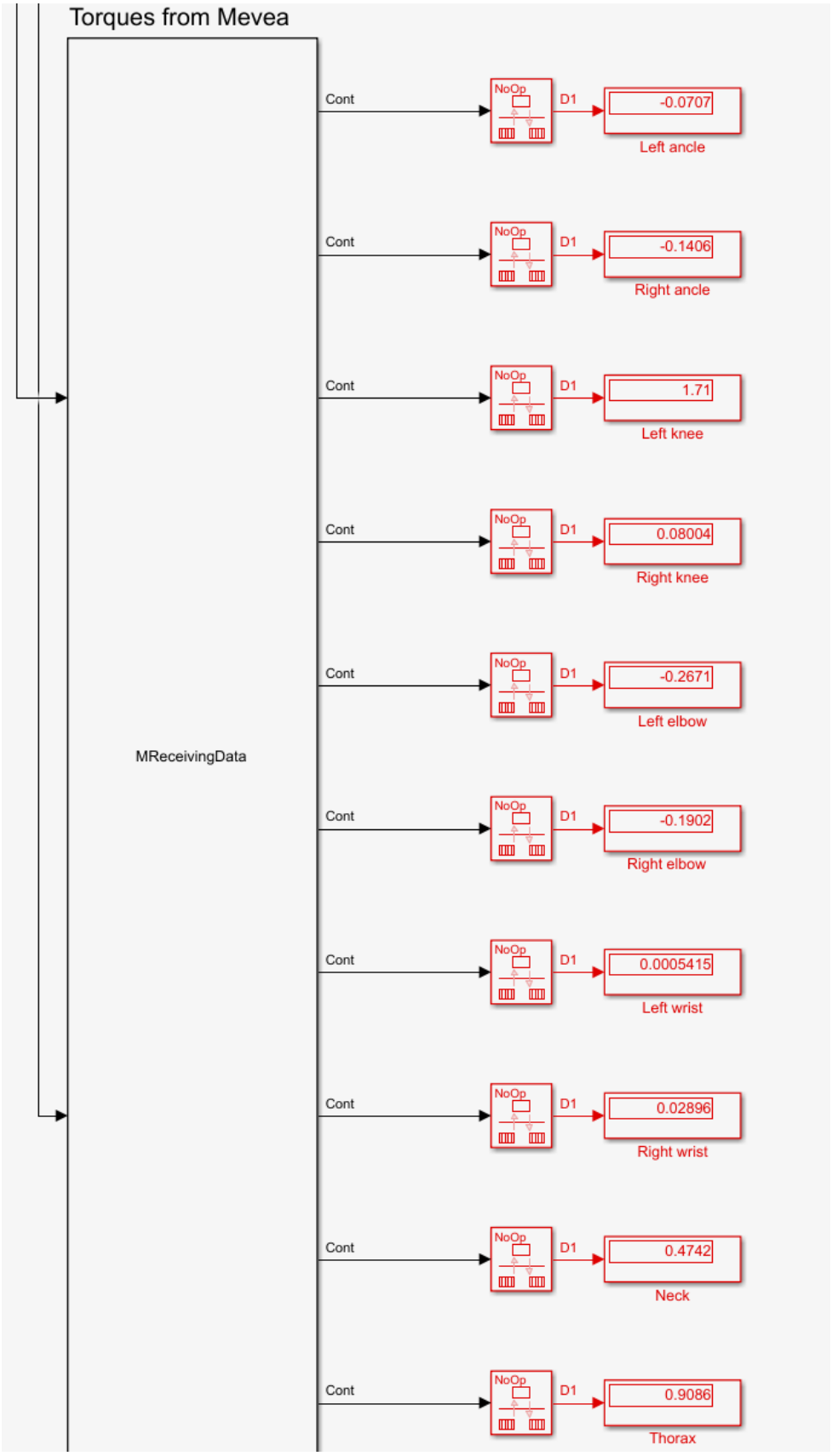


Figure 42. Torques obtained from Mevea simulation software.

4.3.1 Analysis of the case study 3

The system works as it was expected. The camera recognizes the coordinates of human skeletal joints, and the model responds to the person's movements. However, it should be mentioned that the system has two significant drawbacks. The first drawback is associated with the system delay. It means that the skeletal data of the real person cannot be processed properly at the same time with the simulation in Mevea Solver. The possible reason for the system delay is sophisticated simulation model.

The second limitation of the model is associated with difficulties to create a standing equilibrium of the full body model. The reason for that is heavy machinery orientation of the Mevea simulation software. Mevea software does not have a special option for creating the ground reaction forces for structures that are different from machinery.

5 DISCUSSION

The purpose of this chapter is to analyze the whole study and make conclusions about the research project and model. Initially, the main application of the model is discussed. Then, the advantages and limitations of the model are presented. Finally, possible improvements of the model and possible future researches, associated with the model, are described.

5.1 Application of the model

The main application field of the real-time model is physiotherapy. The model can be utilized as a rehabilitation tool for disabled and injured people. For instance, it is possible to connect the model to the physiotherapeutic rehabilitating games, shown in figure 43. Physiotherapist will be able to observe and evaluate the articular torques of the person, who moves in front of the camera. Then, a physiotherapist can draw conclusions about further treatment, based on the results of the simulation.

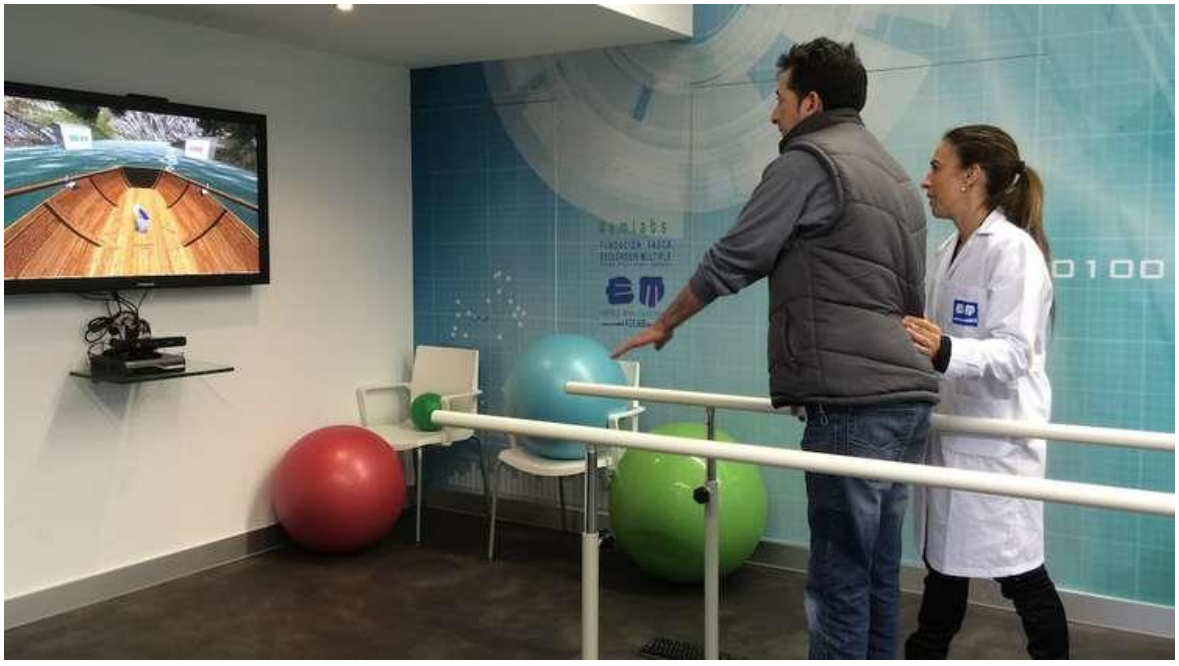


Figure 43. Physiotherapeutic rehabilitating games (Fitness Gaming 2014.).

5.2 Model advantages and limitations

First, it is necessary to say that the model can be presented as a successful real-time model of the human skeleton. The first reason is that the simulation model of the skeleton is based

on a real human skeleton. The skeleton behavior is close to the behavior of real person due to similar bone structures, joints, and mathematical model.

The main advantage of the model is the possibility to obtain torques in human joints via motion capture control in real time. The approach is reliable due to Mevea software scientifically proven physical engine.

Despite the number of advantages, the created multibody model has several limitations. The first limitation of the model is associated with accuracy. The main problem of accuracy is observed between the input and output values inside the embedded Mevea controller. There is an error between these values. The initial and actual angular positions of the joint are different due to this error. There is also a problem with torque fluctuating, but the overall system is stable.

The next limitation is associated with the strict specifications of the model. The created skeleton has exact height and weight. It can be problematic to adjust it in accordance with height and weight of different people, participating in physiotherapeutic rehabilitation sessions. In this case, conventional motion capture technology is much more convenient, compared to the Mevea real-time model.

The third limitation of the model was explained in Ch. 4.3.1. The variety of movements that can be applied to the model is strictly limited due to the absence of constant equilibrium of the model. It means, that the model cannot be utilized for physiotherapeutic sessions of walking due to impossibility to support the equilibrium.

5.3 Possible improvements and future researches

Possible improvements of the model can be associated with establishing a valid connection with video device. It means that the delay of the system has to be eliminated. The next improvement of the system can be the tuning of the spring and damper constants of spring-damper-torque actuator in joints to achieve smooth movements without torque fluctuations. Mevea embedded controller also has to be tuned.

Future researches can be associated with the continuation of the work related to the development of standing equilibrium of the human multibody model. This is a great opportunity to develop the ability to walk.

Creation a valid model for muscles and muscle forces can be also a huge contribution to the scientific community. It would be more practical to use muscle forces rather than torques in order to expand the field of possible applications of the model. There is also an opportunity to make the skeleton model movements more realistic. For this purpose, it is reasonable to make more joints that are movable.

Another research, which can be implemented in future, is associated with the development of flexible bones in Mevea, according to flexible multibody dynamics rules and approaches. This feature will give the possibility to study stresses and strains of bones that can be helpful for osteoporosis diagnosis.

6 CONCLUSION

The first objective of the research was to develop a real-time model of human skeleton in Mevea simulation software for clinical application. The next goal was to obtain and analyze the motion data of the human standing in front of motion capture device. In the introduction chapter of this thesis, research questions of the project were stated. These research questions were:

1. How joint torques can change in real-time when the skeleton model moves?
2. How to develop a skeletal multibody system in Mevea?
3. Is it possible to build a real-time model of the human skeleton in a simulation environment which is oriented to heavy machinery?
4. What challenges are possible during modeling?

It is possible to say that the research answered to the research questions. The answer to the first question is given in Ch. 4 of the thesis. In Mevea software, torques can change in real time. The torque is dependent on the angular position of the joint due to the mathematical model of the spring-damper-torque actuator. In terms of kinematics, every torque in the kinematic chain is dependent on each other, and torques of the whole system can change according to the angular position of even one joint. Case studies 1 and 2 showed the dependence of the torque on angular position in the multibody system. Case study 3 described the connection between the real human movements and the torques obtained from the simulation.

The answer to the second question is given in Ch. 3 of this thesis. Main steps of development of a skeletal multibody model are graphics obtaining, bodies and joints development, forces, collisions, and inputs creation, and building a Simulink external interface with Mevea data sources and socket signals.

The answer to the third question is positive. Mevea simulation software provides a possibility to build and simulate a human skeleton model. However, the modeling process is complicated, and it has several challenges compared to heavy machinery multibody modeling. Main challenges of the modeling are the development of standing equilibrium of

the human skeleton and right tuning of the embedded control system in order to obtain precise torque and angular position values.

Finally, the results of research can be admitted as positive. The results provide the possibility to use the multibody model for physiotherapeutic rehabilitation. Moreover, the research work opens an opportunity for further researches based on this topic. There is an opportunity for research related to the improvement of the model for clinical purposes as well as the development of muscles and flexible bones.

LIST OF REFERENCES

Abdeetedal, M. (2014). *DH Parameters*. [online] Etedal.net. Available at: http://www.etedal.net/2014/03/dh-parameters.html [Accessed 13 Nov. 2018].

Al Nazer, R., Klodowski, A., Rantalainen, T., Heinonen, A., Sievänen, H. and Mikkola, A. (2011). A full body musculoskeletal model based on flexible multibody simulation approach utilised in bone strain analysis during human locomotion. *Computer Methods in Biomechanics and Biomedical Engineering*, 14(6), pp.573-579.

Autodesk. (2018). *3ds Max | 3D Modeling, Animation & Rendering Software | Autodesk*. [online] Available at: https://www.autodesk.com/products/3ds-max/overview# [Accessed 24 Oct. 2018].

Bauer, A., Neog, D., Dicko, A., Pai, D., Faure, F., Palombi, O. and Troccaz, J. (2017). Anatomical augmented reality with 3D commodity tracking and image-space alignment. *Computers & Graphics*, 69, pp.140-153.

Blundell, M. and Harty, D. (2004). *The multibody systems approach to vehicle dynamics*. Elsevier Science and Technology, pp. 95-96, 99.

Chao, E. (2003). Graphic-based musculoskeletal model for biomechanical analyses and animation. *Medical Engineering & Physics*, 25(3), pp.201-212.

Cision. (2015). *Motion capture set to become more mainstream in medical science predicts Vicon*. [online] Available at: http://news.cision.com/vicon/r/motion-capture-set-to-become-more-mainstream-in-medical-science-predicts-vicon,c9722569 [Accessed 21 Nov. 2018].

Dictionary.com. (2018). *The definition of biomechanics*. [online] Available at: https://www.dictionary.com/browse/biomechanics [Accessed 14 Oct. 2018].

Fitness Gaming. (2014). *Virtualware Joins IISART*. [online] Available at: https://www.fitness-gaming.com/news/health-and-rehab/virtualware-joins-iisart.html [Accessed 23 Nov. 2018].

GitHub. (2016). *alesgraz/kinect2-SDK-for-Simulink*. [online] Available at: https://github.com/alesgraz/kinect2-SDK-for-Simulink [Accessed 26 Oct. 2018].

Hamill, J., Knutzen, K. and Derrick, T. (2015). *Biomechanical basis of human movement*. 4th ed. Wolters Kluwer, pp. 31-33, 49-53, 62, 70-72.

Ipfs.io. (2018). *Rotation matrix*. [online] Available at: https://ipfs.io/ipfs/QmXoypizjW3WknFiJnKLwHCnL72vedxjQkDDP1mXWo6uco/wiki/Rotation_matrix.html [Accessed 12 Oct. 2018]

Khan Academy. (2018). *The musculoskeletal system review*. [online] Available at: https://www.khanacademy.org/science/high-school-biology/hs-human-body-systems/hs-the-musculoskeletal-system/a/hs-the-musculoskeletal-system-review [Accessed 13 Nov. 2018].

Kinect for Windows Product Blog. (2014). *Revealing Kinect for Windows v2 hardware*. [online] Available at: https://blogs.msdn.microsoft.com/kinectforwindows/2014/03/27/revealing-kinect-for-windows-v2-hardware/ [Accessed 23 Nov. 2018].

Malina, R., Bouchard, C. and Bar-Or, O. (2004). *Growth, maturation, and physical activity*. 2nd ed. Champaign, IL.: Human Kinetics, p.121.

Mathworks. (2018a). *Image Acquisition Toolbox*. [online] Available at: https://se.mathworks.com/products/imaq.html [Accessed 26 Oct. 2018].

Mathworks. (2018b). *Microsoft Kinect for Windows Support from Image Acquisition Toolbox*. [online] Available at: https://se.mathworks.com/hardware-support/kinect-windows.html [Accessed 26 Oct. 2018].

Mathworks. (2018c). *Kinect 2 Interface for Matlab - File Exchange - MATLAB Central*. [online] Available at: https://se.mathworks.com/matlabcentral/fileexchange/53439-kinect-2-interface-for-matlab [Accessed 26 Oct. 2018].

Mevea Ltd. (2018a). *Software for Real-Time Simulation | Mevea*. [online] Mevea. Available at: https://mevea.com/solutions/software/ [Accessed 20 Oct. 2018].

Mevea Ltd. (2018b). *Hardware | Mevea*. [online] Mevea. Available at: https://mevea.com/solutions/hardware/ [Accessed 20 Oct. 2018].

Mevea Ltd. (2018c). *Reference Manual for Solver Library*. V.7.70.2664. Lappeenranta, Finland: Mevea Ltd., pp. 20-27, 33, 44, 146-148.

Mevea Ltd. (2018d). *Mevea Solver User manual.* V.2.2.10. Lappeenranta, Finland: Mevea Ltd., pp. 6-8, 10, 14.

Mevea Ltd. (2018e). *Mevea Modeller User manual*. V.2.1.0. Lappeenranta, Finland: Mevea Ltd., pp. 5-6.

Microsoft (2014). *Kinect for Windows. Human Interface Guidelines v2.0*, pp. 6, 8. [online] Available at: http://download.microsoft.com/download/6/7/6/676611B4-1982-47A4-A42E-4CF84E1095A8/KinectHIG.2.0.pdf [Accessed 23 Oct. 2018]

Murphy, W., Black, J. and Hastings, G. (2016). *Handbook of biomaterial properties*. 2nd ed. New-York: Springer Science, pp.4, 15.

Nedel, L., P., Thalman, D. (2000). Anatomic modeling of deformable human bodies. *The Visual Computer*, 16(6), pp.306-321.

Neto, M. and Ambrosio, J. (2003). Stabilization Methods for the Integration of DAE in the Presence of Redundant Constraints. *Multibody System Dynamics*, 10, pp.81-105.

Pathirana, P., Li, S., Trinh, H. and Seneviratne, A. (2016). Robust Real-Time Bio-Kinematic Movement Tracking Using Multiple Kinects for Tele-Rehabilitation. *IEEE Transactions on Industrial Electronics*, 63(3), pp.1822-1833.

Rantalainen, T., Klodowski, A. (2011). Estimating Lower Limb Skeletal Loading. *Theoretical Biomechanics, Dr. Vaclav Klika (Ed.),* ISBN: 978-953-307-851-9, InTech, Available from: https://www.intechopen.com/books/theoretical-biomechanics/estimating-lower-limb-skeletal-loading

Robertson, D., Caldwell, G., Hamill, J., Kamen, G. and Whittlesey, S. (2004). *Research methods in biomechanics*. Champaign, IL: Human Kinetics, pp.68-71.

Schüler, A., Kerner, S. and Witt, M. (2015). A comparison of two body segment parameter models via angular momentum at takeoff in diving. In: *33rd International Conference on Biomechanics in Sports*. [online] Poitiers, France: Floren Colloud, Mathieu Domalain & Tony Monnet (Editors), p.147. Available at: https://ojs.ub.uni-konstanz.de/cpa/article/viewFile/6346/5715 [Accessed 15 Oct. 2018].

Shabana, A. (1998). *Dynamics of multibody systems*. 2nd ed. Cambridge: Cambridge University Press, pp. 1, 4-5, 11, 19, 33-34, 67-68, 76, 81-84.

Terven J. and Cordova-Esparza, D. (2016). Kin2. A Kinect 2 Toolbox for MATLAB, *Science of Computer Programming.* DOI: http://dx.doi.org/10.1016/j.scico.2016.05.009

Tortora, G. and Grabowski, S. (1993). *Principles of anatomy and physiology*. 7th ed. New York: HarperCollins College Publishers, pp. 166,169.

Willems, N., Langenbach, G., Everts, V. and Zentner, A. (2013). The microstructural and biomechanical development of the condylar bone: a review. *The European Journal of Orthodontics*, 36(4), p. 48

APPENDIX I, 1

List of bodies and their anatomical analogues (bones).

Name of the body in MBS	**№**	**Anatomical analogue in human skeleton**	**№**
Pelvis	1	2 Pelvic bones	2
Sacrum	1	Sacrum	1
Coccyx	1	Coccyx	1
L_leg_femur; R_leg_femur	2	Left and Right Femur	2
L_leg_patella; R_leg_patella	2	Left and Right Patella	2
L_leg_tibia; R_leg_tibia	2	Left and Right Tibia	2
L_leg_fibula; R_leg_fibula	2	Left and Right Fibula	2
L_leg_tarsus; R_leg_tarsus; L_foot_calcaneus; R_foot_calcaneus	4	Left and Right Calcaneus	2
L_foot_metatarsus; R_foot_metatarsus	2	6 Left and 6 Right tarsals	12
L_toeA_1 – L_toeA_3; L_toeB_1 – L_toeB_4; L_toeC_1 – L_toeC_4; L_toeD_1 – L_toeD_4; L_toeE_1 – L_toeE_3; R_toeA_1 – R_toeA_3; R_toeB_1 – R_toeB_4; R_toeC_1 – R_toeC_4; R_toeD_1 – R_toeD_4; R_toeE_1 – R_toeE_3	36	5 Left and 5 Right Metatarsals; 14 Left and 14 Right Toe Phalanges	38
Vert_L1 – Vert_L5	5	Lumbar Vertebras	5
Vert_T1 – Vert_T12	12	Thoracic Vertebras	12
Vert_C1 – Vert_C7	7	Cervical Vertebras	7
Rib_1 – Rib_12; Rib_cartilage	13	24 Ribs	24
Sternum	1	Sternum	1
L_clavicle; R_clavicle	2	Left and Right Clavicle	2
L_scapula; R_scapula	2	Left and Right Scapula	2
Skull	1	8 Cranial bones; 13 Facial bones	21
Jaw	1	Mandible	1

APPENDIX I, 2

List of bodies and their anatomical analogues (bones).

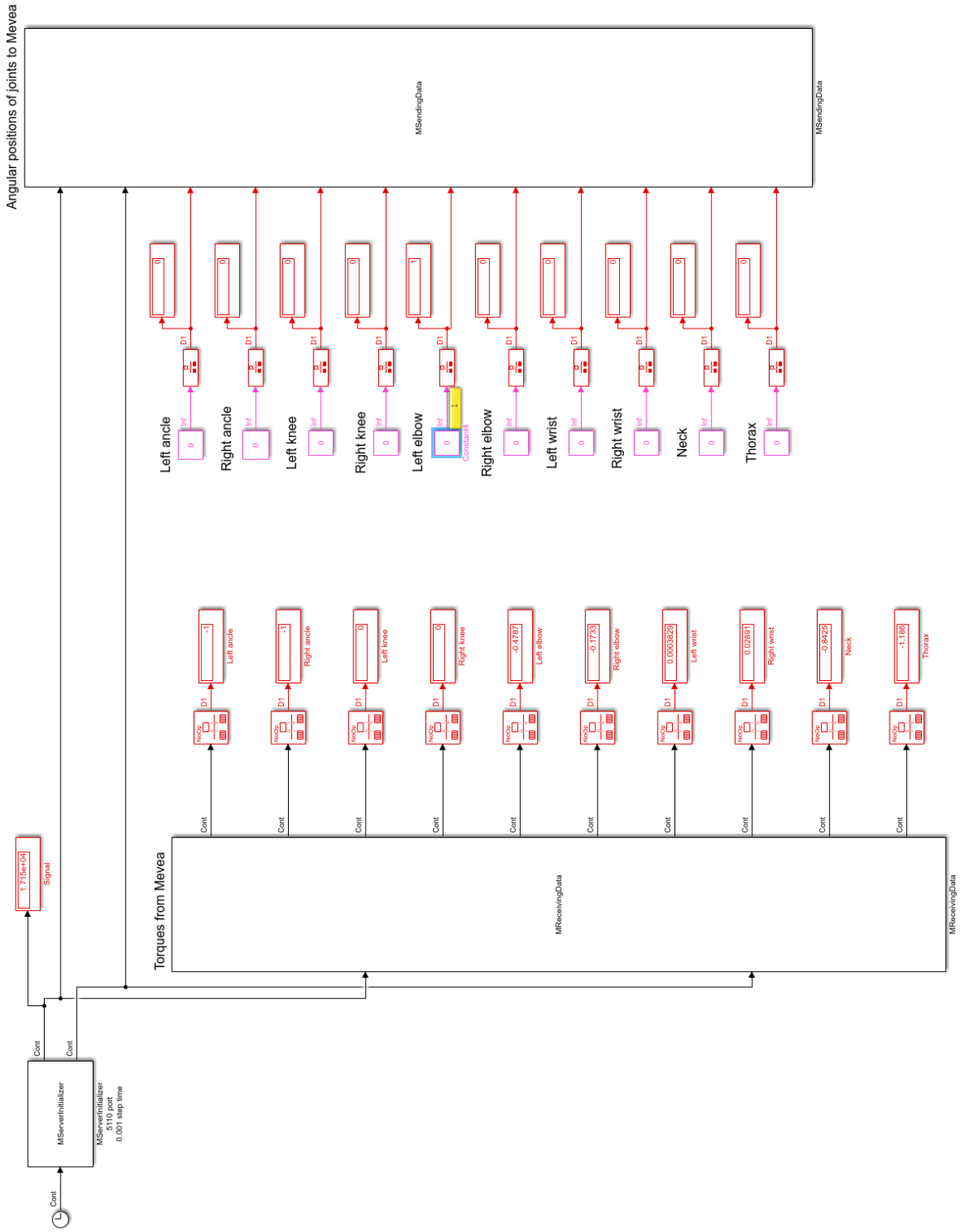
L_arm_humerus; R_arm_humerus	2	Left and Right Humerus	2
L_arm_ulna; R_arm_ulna	2	Left and Right Ulna	2
L_arm_radius; R_arm_radius	2	Left and Right Radius	2
L_hand_carpus; R_hand_carpus	2	8 Left and 8 Right Carpals	16
L_fingerA_1 – L_fingerA_3; L_fingerB_1 – L_fingerB_3; L_fingerC_1 – L_fingerC_3; L_fingerD_1 – L_fingerD_3; L_thumb_1 – L_thumb_3; R_fingerA_1 – R_fingerA_3; R_fingerB_1 – R_fingerB_3; R_fingerC_1 – R_fingerC_3; R_fingerD_1 – R_fingerD_3; R_thumb_1 – R_thumb_3;	30	5 Left and 5 Right Metacarpals; 14 Left and 14 Right Finger Phalanges	38
		Hyoid bone	1
		Auditory Ossicles	6
TOTAL	**135**		**206**

APPENDIX II

Parameters of joint torques of the MBS.

Torque	**Spring constant [Nm/rad]**	**Damping constant [Nms/rad]**	**Spring minimum angle [rad]**	**Spring maximum angle [rad]**
T_L_ancle	12.117	1.2117	-1.5	1.5
T_R_ancle	12.117	1.2117	-1.5	1.5
T_L_knee	15.579	3.1158	-3	0
T_R_knee	15.579	3.1158	-3	0
T_L_elbow	3.462	0.6924	-1	2
T_R_elbow	3.462	0.6924	-1	2
T_L_wrist	1.731	0.3462	-1.5	1.5
T_R_wrist	1.731	0.3462	-1.5	1.5
T_Neck	57.7	11.54	-1.2	1.5
T_Thorax	57.7	11.54	-1	1.5

Simulink external interface for Mevea simulation software.



Matlab code for Kinect for Windows v2.0 skeletal joints visualization (Terven J. and Cordova-Esparza 2016). Authors: Juan R. Terven, Diana M. Cordova-Esparza

```
addpath('Mex');
clear all
close all

% Create Kinect 2 object and initialize it
% Availqable sources: 'color', 'depth', 'infrared', 'body_index', 'body',
% 'face' and 'HDface'
k2 = Kin2('color','depth','body');

% images sizes
d_width = 512; d_height = 424; outOfRange = 4000;
c_width = 1920; c_height = 1080;

% Color image is to big, let's scale it down
COL_SCALE = 1.0;

% Create matrices for the images
depth = zeros(d_height,d_width,'uint16');
color = zeros(c_height*COL_SCALE,c_width*COL_SCALE,3,'uint8');

% depth stream figure
d.h = figure;
d.ax = axes;
d.im = imshow(zeros(d_height,d_width,'uint8'));
%hold on;

title('Depth Source (press q to exit)')
set(gcf,'keypress','k=get(gcf,''currentchar'');'); % listen keypress

% color stream figure
c.h = figure;
c.ax = axes;
c.im = imshow(color,[]);
title('Color Source (press q to exit)');
set(gcf,'keypress','k=get(gcf,''currentchar'');'); % listen keypress
%hold on

% Loop until pressing 'q' on any figure
k=[];

disp('Press q on any figure to exit')

while true
    % Get frames from Kinect and save them on underlying buffer
    validData = k2.updateData;

    % Before processing the data, we need to make sure that a valid
    % frame was acquired.
```

Matlab code for Kinect for Windows v2.0 skeletal joints visualization (Terven J. and Cordova-Esparza 2016). Authors: Juan R. Terven, Diana M. Cordova-Esparza

```
if validData
        % Copy data to Matlab matrices
        depth = k2.getDepth;
        color = k2.getColor;

        % update depth figure
        depth8u = uint8(depth*(255/outOfRange));
        depth8uc3 = repmat(depth8u,[1 1 3]);
        d.im = imshow(depth8uc3, 'Parent', d.ax);

        %set(d.im,'CData',depth8uc3);

        % update color figure
        color = imresize(color,COL_SCALE);
        c.im = imshow(color, 'Parent', c.ax);

        %set(c.im,'CData',color);

        % Get 3D bodies joints
        % Input parameter can be 'Quat' or 'Euler' for the joints
        % orientations.
        % getBodies returns a structure array.
        % The structure array (bodies) contains 6 bodies at most
        % Each body has:
        % -Position: 3x25 matrix containing the x,y,z of the 25 joints in
        %   camera space coordinates
        % - Orientation:
        %   If input parameter is 'Quat': 4x25 matrix containing the
        %   orientation of each joint in [x; y; z, w]
        %   If input parameter is 'Euler': 3x25 matrix containing the
        %   orientation of each joint in [Pitch; Yaw; Roll]
        % -TrackingState: state of each joint. These can be:
        %   NotTracked=0, Inferred=1, or Tracked=2
        % -LeftHandState: state of the left hand
        % -RightHandState: state of the right hand
        [bodies, fcp, timeStamp] = k2.getBodies('Quat');

        % Number of bodies detected
        numBodies = size(bodies,2);
        %disp(['Bodies Detected: ' num2str(numBodies)])

        % Example of how to extract information from getBodies output.
       if numBodies > 0
            % first body info:
            %disp(bodies(1).TrackingState)
            %disp(bodies(1).RightHandState)
            %disp(bodies(1).LeftHandState)

            %disp('Right Hand Orientation') % see Kin2.m constants
```

Matlab code for Kinect for Windows v2.0 skeletal joints visualization (Terven J. and Cordova-Esparza 2016). Authors: Juan R. Terven, Diana M. Cordova-Esparza

```
            %disp('Orientation')
            %disp(bodies(1).Orientation(:,k2.JointType_HandRight));

            disp('Position')
            disp(bodies(1).Position)

            disp('Floor Clip Plane')
            disp(fcp);

            disp('Body Timestamp')
            disp(timeStamp);


            % To get the joints on depth image space, you can use:
            %pos2D = k2.mapCameraPoints2Depth(bodies(1).Position');
        end


        %To get the joints on color image space, you can use:
        %pos2D = k2.mapCameraPoints2Color(bodies(1).Position');

        % Draw bodies on depth image
        % Parameters:
        % 1) image axes
        % 2) bodies structure
        % 3) Destination image (depth or color)
        % 4) Joints' size (circle raddii)
        % 5) Bones' Thickness
        % 6) Hands' Size
        k2.drawBodies(d.ax,bodies,'depth',5,3,15);

        % Draw bodies on color image
        k2.drawBodies(c.ax,bodies,'color',10,6,30);

    end

    % If user presses 'q', exit loop
    if ~isempty(k)
        if strcmp(k,'q'); break; end;
    end

    pause(0.02)
end


% Close Kinect object
k2.delete;

close all;
```