

Application of imitation learning in human-robot interactions

Knežević, Tara

Master's thesis / Diplomski rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:235:151455>

Rights / Prava: [In copyright / Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-05-14**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



UNIVERSITY OF ZAGREB
FACULTY OF MECHANICAL ENGINEERING AND NAVAL
ARCHITECTURE

MASTER'S THESIS

Tara Knežević

ZAGREB, 2023.

UNIVERSITY OF ZAGREB
FACULTY OF MECHANICAL ENGINEERING AND NAVAL
ARCHITECTURE

MASTER'S THESIS

APPLICATION OF IMITATION LEARNING IN HUMAN-ROBOT
INTERACTION

Mentors:

Asst. Prof. Filip Šuligoj, Ph.D., Mech. Eng.

Marija Radmilović, Ph.D., El. Eng.

Student:

Tara Knežević

ZAGREB, 2023.

I would like to express my gratitude to my mentor, Filip Šuligoj, for proposing the topic of my master's thesis and providing valuable guidance during its development. I would also like to thank Marija Radmilović, my mentor, for her professional assistance, availability, and responsiveness despite the physical distance.

Additionally, I extend my thanks to Marko Švaco for the support he has provided throughout my studies. Furthermore, a big thank you to everyone at the Regional Center of Excellence for Robotic Technology (CRTA) and the Mihajlo Pupin Institute for the pleasant working atmosphere and valuable advice. I am grateful to those who volunteered to help complete this master's thesis, including the design of the exercise set, the creation of the suit, and the recording of the exercises. A special thanks to all of my friends and colleagues who have made my student days more enjoyable and less stressful.

Finally, I want to express my heartfelt gratitude to my family for their trust, support, and love, which have served as my greatest motivation. Thank you very much!

Statement | Izjava

I hereby declare that I have made this thesis independently using the knowledge acquired during my studies and the cited references.

Izjavljujem da sam ovaj rad radila samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zagreb, December 2023

Tara Knežević



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske ispite studija strojarstva za smjerove:
Proizvodno inženjerstvo, inženjerstvo materijala, industrijsko inženjerstvo i menadžment,
mehatronika i robotika, autonomni sustavi i računalna inteligencija

Sveučilište u Zagrebu	
Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: 602 - 04 / 23 - 6 / 1	
Ur.broj: 15 - 23 -	

DIPLOMSKI ZADATAK

Student: Tara Knežević

JMBAG: 0035223365

Naslov rada na hrvatskom jeziku: **Primjena imitacijskog učenja u interakcijama između čovjeka i robota**

Naslov rada na engleskom jeziku: **Application of imitation learning in human-robot interactions**

Opis zadatka:

In the last decade, the interaction between humans and robots has become ubiquitous in various aspects of life, encompassing sectors such as healthcare, education, rehabilitation, and social interaction. This is a consequence of the growing need for automation and intelligent systems capable of operating in close collaboration with humans. Essential challenges in this domain include enabling robots to mimic human movements and ensuring the reliability of the instructions provided by robots to humans. In light of these challenges, there is a need for the development of methods for mapping human movements onto robots and algorithms for reliably estimating errors when humans follow robotic instructions.

Within the scope of the work, it is necessary to:

1. Establish a laboratory framework for recording human movements, which includes an OptiTrack motion capture system (MCS) and corresponding markers, and to select exercises to be used in the experiments.
2. Develop a kinematic model of the selected robot and create an imitation algorithm for mapping human movements onto the robot, considering the robot's limitations.
3. Implement an algorithm for trajectory mapping and motion execution with the humanoid robot Pepper.
4. Conduct experimental evaluations to test the quality and applicability of the developed system in the context of a specific application.

In the work, it is necessary to cite the literature used and any assistance received.

Zadatak zadan:

28. rujna 2023.

Zadatak zadao:

Doc. dr. sc. Filip Šuligoj

Dr. sc. Marija Radmilović

Datum predaje rada:

30. studenoga 2023.

Predviđeni datumi obrane:

4. – 8. prosinca 2023.

Predsjednik Povjerenstva:

Prof. dr. sc. Ivica Garašić

Contents

Contents	v
List of Figures	vii
List of Tables	x
Summary	xi
Prošireni sažetak	xii
1. Introduction	1
1.0.1. Approach to the task	4
2. Humanoid robot Pepper	7
3. Motion Capture System	10
3.0.1. Applications of human motion capture	11
3.0.2. MCS in robotics	12
3.0.3. Description of the selected system	13
3.0.4. Post-processing of the data	17
3.0.5. Selected exercises	22
4. Kinematic Model of a Robot	25
4.0.1. Robot kinematics	25
4.0.2. Kinematics of humanoid robot Pepper	28

5. The Imitation Algorithm	32
5.0.1. Initialization algorithm	33
5.0.2. Imitation algorithm	38
6. Exercises performed by the robot	46
7. Results	48
8. Conclusion	58
Bibliography	60

List of Figures

1.1	Types of robotic systems (a) Industrial robot [5] (b) Mobile robot [6] (c) Humanoid robot [7] (d) Robot-drone [8]	2
1.2	Five levels of human-robot interaction [9] (a) No collaboration, (b) Co-existence, (c) Synchronisation, (d) Cooperation, (e) Collaboration	3
1.3	Phases of the thesis	6
2.1	Humanoid robot Pepper [15]	8
3.1	Classification of motion capture systems [17]	11
3.2	Primex13 camera [30]	13
3.3	Camera setup view 1 (with cameras circled in red)	14
3.4	Camera setup view 2 (with cameras circled in red)	15
3.5	(a) Marker positions on the body's front (b) Marker positions on the body's back	16
3.6	Zero position	17
3.7	(a) Marker before editing (b) Labeled marker sets (c) Marker with a lost label	18
3.8	The position of eight cameras in relation to the human	18
3.9	(a) Untrimmed data (b) Trimmed data	20
3.10	(a) Error due to the wrong interpolation method (b) Hand marker set (c) Error in the data due to the wrong interpolation method (d) Deleted part of the data (e) Correctly reconstructed data	21

3.11 (a) Error due to the peak of a signal (b) Deleted part of the data (c) Correctly reconstructed data	22
3.12 Exercises captured using a smartphone camera (top row) and the Opti-Track system (bottom row)	23
3.13 Pepper's degrees of freedom [33]	24
4.1 (a) Tree-structured robot notations [34] (b) Tree-structured robot MDH parameters definition [34]	26
4.2 Pepper's kinematic configuration [33]	29
5.1 Kinematic model of an upper body of a human [39]	33
5.2 Initialization algorithm flowchart	34
5.3 Illustration of 3D matrix	35
5.4 Markers used in defining the segment lengths of a human	36
5.5 Imitation algorithm flowchart	40
5.6 Example of calculating the transformation matrix that describes the relation between the marker on the elbow and forearm segment	41
5.7 3D data visualization	44
5.8 (a) Boundaries and trajectory (b) Raw and filtered trajectory	45
6.1 Exercises in a simulator (top row) and implemented on the robot (bottom row).	47
7.1 Three positions that alternate during the exercise	49
7.2 Generalized coordinates for the left arm in exercise number 5	50
7.3 Generalized coordinates for the right arm in exercise number 5	51
7.4 Error in following virtual markers with real markers on the upper arm, forearm, and hand during the exercise	52
7.5 Spatial representation of desired and achieved trajectories for segment markers	53
7.6 Shoulder translation along the z-axis [39]	54
7.7 Error in following the virtual marker with the real marker of the shoulder during the exercise	55
7.8 Error in following the virtual marker with the real marker of the elbow during the exercise	55

7.9	Error in following the virtual marker with the real marker of the wrist and spatial representation of desired and achieved trajectories during the exercise	56
7.10	Error in following virtual markers with real markers on the thumb, middle finger and pinky finger during the exercise	57

List of Tables

2.1	Technical specifications of a humanoid robot Pepper [15]	9
3.1	Technical specifications of Primex13 camera [30]	14
4.1	MDH parameters of a humanoid robot Pepper	30
4.2	Additional parameters for SYMORO+ software input	31
5.1	Overview of the markers utilized to measure the length of each section	36
5.2	Boundaries of Pepper's joints (left and right arm) [40]	38
5.3	The pairing of the transformation matrices of joint markers and corresponding segment markers	41

Summary

The goal of this thesis is to capture and examine human movements throughout the set of defined exercises and analyze the movements. Furthermore, to transfer these movements from the task space to the joint space using an imitation algorithm, and then map the movements to a humanoid robot. For that purpose, a series of exercises aimed at enhancing the upper body's gross motor skills of children were created with the assistance of kinesiology specialists. An OptiTrack system and an upper-body suit were utilized to capture human movements using markers placed on the suit. The humanoid robot Pepper was used for the implementation of the exercises. The thesis is organized as follows: initially, an introduction is provided, explaining the thoughts behind robots, human-robot interactions, and task elaboration. Following are the chapters that go into each step of the process needed to finish the assignment. Additionally, an analysis is conducted to determine the degree to which the human movement translated to the robot is accurate. Accuracy is measured based on how closely the virtual markers (specified on the extended kinematic model of the human) tracked the positions of the real markers. Based on these numerical data, a conclusion regarding the effectiveness of humanoid robot motion imitation was drawn.

Keywords: human-robot interaction, humanoid robot, motion capture system, imitation algorithm

Prošireni sažetak

Cilj diplomskog rada je snimiti i proučiti ljudske kretnje tijekom niza definiranih tjelesnih vježbi, analizirati te kretnje, te ih potom prenijeti iz vanjskih koordinata koje opisuju kretnju čovjeka u unutrašnje koordinate robota pomoću imitacijskog algoritma i implementirati na humanoidnom robotu. U tu svrhu je uz pomoć stručnjaka iz područja kineziologije osmišljen set od ukupno pet vježbi usmjerenih na poboljšanje grubih motoričkih vještina gornjeg dijela tijela kod djece.

Struktura rada je sljedeća: prvo je dan uvod u kojem su objašnjeni pojmovi robota, interakcije robota i čovjeka te razrada zadatka. Nakon toga, slijede poglavlja u kojima je opisan postupak za svaki dio procesa potrebnog za izvršavanje zadatka. Dodatno, provedena je analiza kako bi se utvrdila točnost prijenosa ljudskog pokreta na robota. Točnost se mjeri prema tome koliko su dobro virtualni markeri (definirani na proširenom kinematskom modelu čovjeka) pratili položaje stvarnih markera. Na temelju tih numeričkih podataka donesena je zaključna ocjena o učinkovitosti imitacije pokreta humanoidnih robota.

Istraživanje [1] ukazuje da su ljudi više potaknuti i motivirani ući u interakciju s robotom kada im se obrati prijateljsko lice sposobno za govornu interakciju i kada njihovi pokreti i geste podsjećaju na ljudske. U ovom diplomskom radu, za implementaciju generiranih kretnji te interakciju s čovjekom odabran je humanoidni robot Pepper (Slika 2.1, Tablica 2.1). Interakcija čovjeka i robota je multidisciplinarno područje čiji je primarni cilj omogućiti sigurnost čovjeka pri interakciji s robotom, pretpostavljajući da se radni prostori čovjeka i robota smiju preklapati, te da je odnos između čovjeka i robota suradnički. Ako trajektorije koje robot izvodi podsjećaju na ljudske radnje, čovjek može procijeniti radni prostor robota, čime se minimizira rizik od kolizije između čovjeka i robota. Ovaj cilj postiže se raznim pristupima, od kojih je jedan imitacijski algoritam. U ovom pristupu roboti uče pokrete promatranjem i oponašanjem, često koristeći demonstracije ljudi.

Sustav za snimanje pokreta OptiTrack koristi se za prikupljanje podataka. To je sustav sačinjen od osam infracrvenih kamera (Slika 3.2), odijela za čovjeka te retro-reflektivnih sfera. Za potrebe rada, snimljeno je ukupno pet različitih vježbi koje je izveo volonter. Pokreti su snimani i kasnije obrađeni u softveru Motive [2]. Tijekom snimanja, 3D položaji markera bilježe se u odnosu na globalni koordinatni sustav koji je postavljen na tlu, između nogu volontera. Podaci su prikupljeni frekvencijom od 120 Hz te su nakon snimanja obrađeni kako bi se uklonile eventualne greške u snimljenom signalu.

Kako bi se prenijeli snimljeni ljudski pokreti na robota, potreban je model transformacije između vanjskih i unutrašnjih koordinata. Kinematika robota opisana je korištenjem MDH parametara (Slika 4.2, Tablice 4.1, 4.2). Taj model proširen je s dvije dodatne rotacije po ruci i prilagođen duljinama segmenata volontera. Prošireni kinematski model zatim je korišten kao pojednostavljeni kinematski model čovjeka. Nakon toga, algoritam inicijalizacije upotrijebljen je za povezivanje virtualnih markera s kinematskim modelom čovjeka. Bitno je napomenuti da su 'stvarni markeri' oni čije su pozicije snimljene sustavom za snimanje pokreta, dok su 'virtualni markeri' oni definirani na proširenom kinematskom modelu robota.

Korištenjem inverzne kinematike, algoritam imitacije se primjenjuje kako bi se cijeli snimljeni pokret prenio s čovjeka na robota. Algoritam imitacije formiran je kao minimizacija razlike u pozicijama između stvarnih i virtualnih markera, uz primjenu inverzne kinematike.

Nakon generiranja trajektorija za sve vježbe, napisana je Python skripta koja je omogućila komunikaciju između računala i robota. Generirane su trajektorije zatim implementirane na humanoidnom robotu Pepper-u.

Na kraju je provedena analiza točnosti praćenja virtualnih markera u odnosu na stvarne pozicije. Rezultati analize ukazali su da su pogreške u položajima markera izravna posljedica nedostajućih translacijskih stupnjeva slobode u kinematskom modelu čovjeka. Taj model definiran je kao proširenje kinematskog modela robota, što je uvelo određena pojednostavljenja. Kinematski model čovjeka je složeniji od napravljenog, uključujući tri rotacije i tri translacije u ramenu. Odabrane vježbe obuhvaćaju različite pokrete ramena, što čini razlike primjetnijima u usporedbi s pokretima usredotočenim na lakat ili šaku.

Rotacijski stupnjevi slobode gibanja su uspješno mapirani s čovjeka na robota čime je ostvareno mapiranje odabranih vježbi za poboljšanje grube motorike kod djece.

1 Introduction

Robotics is a multidisciplinary engineering field whose primary goal is the design, manufacturing, and control of robots and robotic systems and their application in the human environment. Robots and their capabilities and performance have evolved over time as a result of the development of hardware and software components. Historically, the first versions of robots were powered by pneumatics or hydraulics and were manually controlled by workers rather than computers. Over time, the robot's drive became electric, and simple control algorithms were developed, allowing the robots to be guided in space using so-called 'point-to-point' algorithms. The integration of sensors and robots resulted in the robotics revolution, allowing for the development of a perception of the space around the robots. Furthermore, the use of machine learning algorithms enabled the robots to draw conclusions about their surroundings based on data collected from sensors. As a result, the robots gained the ability to act autonomously in their surroundings [3]. Because of the advancements mentioned, robotics has spread into diverse areas of application. As a result, robotic field has evolved from its initial industrial applications to domestic, educational, medical, and military applications [4]. Also, in the beginning, there were only robotic arms, but as time passed, mobile robots, cobots, drones, and humanoids developed. Figure 1.1 depicts various robotic systems.

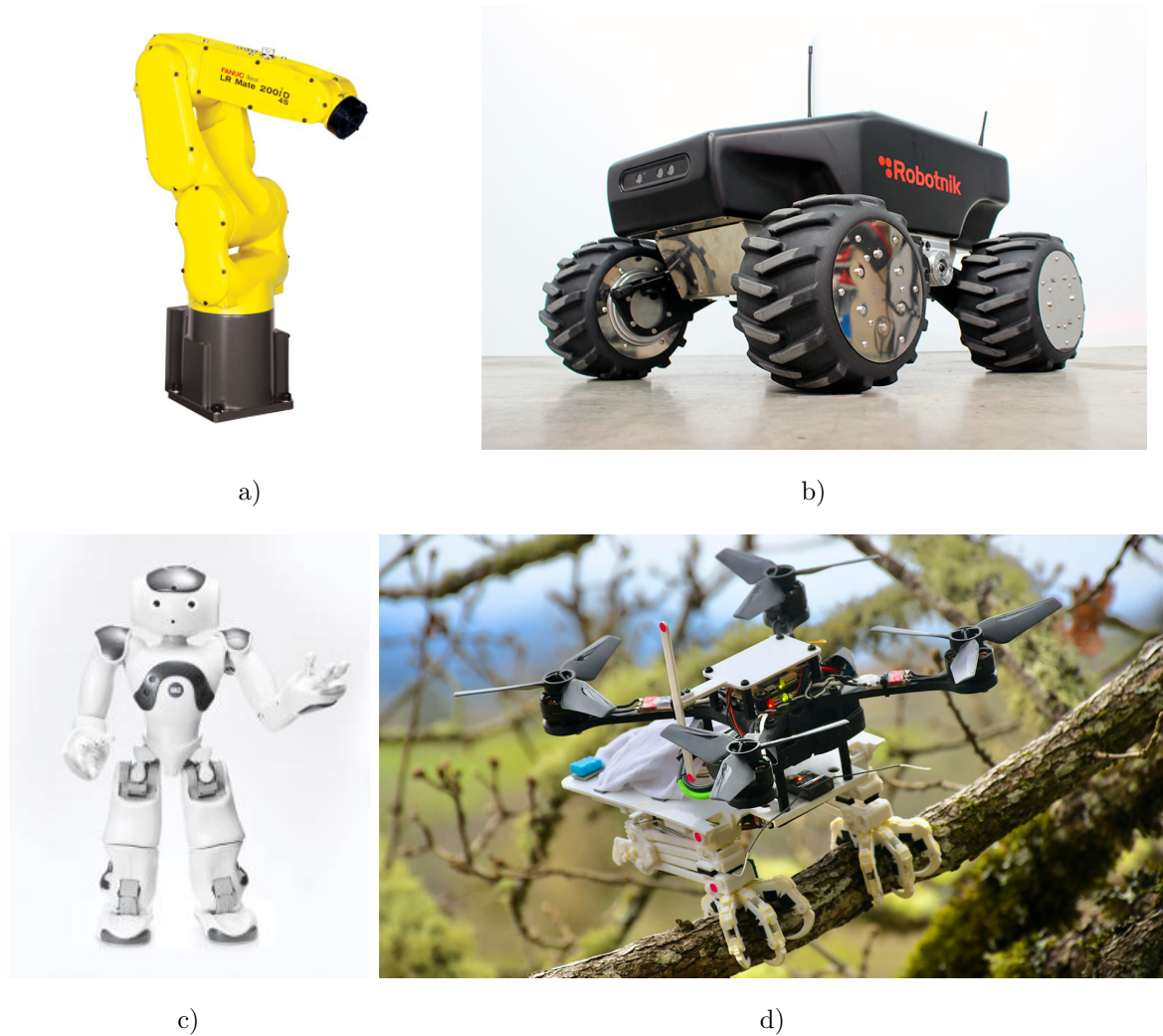


Figure 1.1: Types of robotic systems (a) Industrial robot [5] (b) Mobile robot [6] (c) Humanoid robot [7] (d) Robot-drone [8]

The interaction between robots and humans is an interdisciplinary field that is constantly expanding its range of applicable disciplines and industries. The primary goal of human-robot interaction is to keep humans safe at work, assuming that human and robot workspaces may overlap without resulting in human injury and that the human-robot relationship is cooperative. This discipline encompasses engineering, computer science, robotics, design, sociology, and psychology. The challenge in finding a solution comes from the fact that experts from various fields must collaborate to design an acceptable solution, with an emphasis on understanding the social aspects that occur and

human safety during interaction. Image 1.2 demonstrates five levels of human-robot collaboration. The first two views (Figures 1.2 a), 1.2 b)) depict a situation in which the robot and the human have no contact and their workspaces do not clash. The only difference is that in the first case, the robot's workspace is physically protected and is not accessible to humans. In the other three cases (Figures 1.2 c), 1.2 d), 1.2 e)), the human and robot workspaces overlap, and additional requirements must be met in the robot's construction, management, and the sensors that are integrated with the robot and its workspace to ensure human safety.

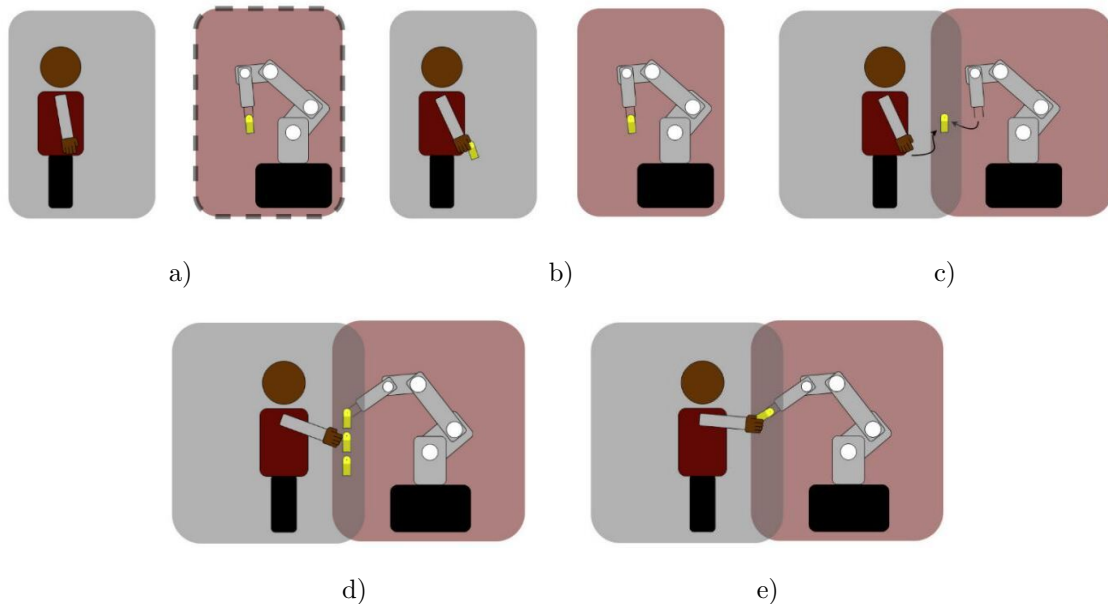


Figure 1.2: Five levels of human-robot interaction [9] (a) No collaboration, (b) Coexistence, (c) Synchronisation, (d) Cooperation, (e) Collaboration

Additionally, when the robot's movements closely mirror human actions, individuals can naturally assess the robot's workspace during task execution, thereby minimizing the risk of collisions between humans and robots. The popularity of imitation learning is on the rise because it allows robots to mimic human-like behavior. In this machine learning approach, robots learn movements by observing and imitating, often using expert demonstrations. This concept is emphasized in the research by Dzedzickis et al. [10]. The assumption is that the demonstrations are carried out optimally for the application and that the range of motions can be replicated using the chosen robotic

system. Once a model of the link between the demonstrator and the robotic system kinematics is created, new movements can be easily recorded and transferred to the robot. The study Maeda et al. [11] shows that this method is not trivial because the human arm has seven degrees of freedom of movement, whereas robot arms typically have six. They further claim that mapping cannot be performed without the so-called 'mapping error', which is caused by differences in the kinematics of humans and robots.

1.0.1. Approach to the task

The objective of master thesis was to address the challenge of mapping human movements onto a robot and constructing a model that establishes a connection between the kinematics of a human and the chosen humanoid robot. The choice of a humanoid robot for implementation results from the goal of using the robot to demonstrate exercises aimed at improving gross motor skills of children. This choice was made due to the humanoid robot's kinematics closely resembling that of a human, making it the most suitable robotic system for this purpose.

For this thesis, kinesiology experts created a set of five different exercises for the improvement of gross motor skills in children. Gross motor skills exercises focus on large muscle groups and are important for everyday human functioning because they play an important role in fundamental movements such as walking, running, lifting, and maintaining balance. Also, it is important that the chosen robot for implementation can effectively perform the designed exercises. They are all asymmetrical (when comparing the movements of the left and right arms during a specific exercise), and the hand rotations change during the exercises. As a result, the exercises are demanding to complete because they require both physical fitness and mental concentration.

The humanoid robot Pepper [12] was ultimately selected to carry out the mapped human movements. When the shoulder, elbow, and wrist joints are considered, the human arm has a total of seven degrees of freedom. Pepper, on the other hand, possesses a total of five joint movements, a factor considered during the formulation of the set of exercises. The defined exercises cover the complete range of motion of these five joint movements accessible to the robot, while the two extra degrees of freedom in the human arm remain unutilized.

For recording human movements, a suit to which retroreflective spheres are attached

was created. The sphere positions on the suit are determined so that they can be used to define the joint positions and segments of the arm. The OptiTrack motion capture system was used to record the exercises. The three-dimensional coordinates of the markers in time are obtained as the output and used as the input to the imitation algorithm.

The robot's kinematic model was created to describe the relationship between coordinates in the task and joint space. The kinematic model of the robot was extended with two more degrees of freedom of motion, and the lengths of the robot segments were scaled to the lengths of the volunteer segments to map the movements recorded by the volunteer to the robot.

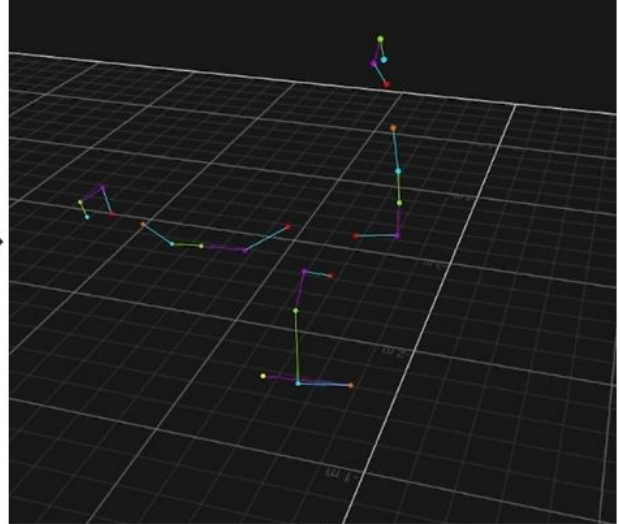
Finally, using the available API [13] and the Python programming language, the movement was successfully implemented on the humanoid robot Pepper.

Figure 1.3 demonstrates different phases of the thesis. First, motion capture data has to be obtained and processed. After that, an imitation algorithm must be utilized to generate trajectories described with generalized coordinates. Trajectories are tested in simulation, and if the behavior of the virtual robot is as expected, they are implemented on the robot.

Real world



Motion capture data



Robot implementation



Robot simulation

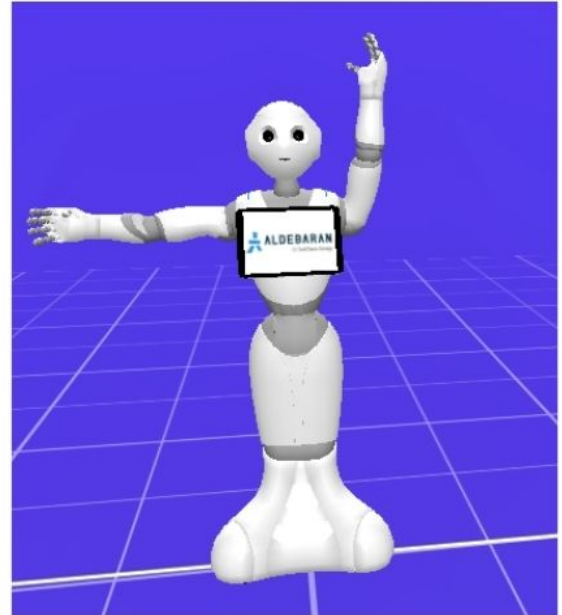


Figure 1.3: Phases of the thesis

2 Humanoid robot Pepper

This section will introduce Pepper, a humanoid robot designed to elicit positive responses by balancing human-like features with a non-threatening appearance, and discuss the importance of early exposure to robots in children’s environments for fostering future positive interactions. Robots that possess social interaction capabilities can engage with people most effectively. This is supported by research conducted by Kanda et al. [1], which indicates that humans are more encouraged and motivated when greeted by a friendly face capable of spoken interaction and human-like body gestures. If humans can identify with a robot to a certain extent, the opportunity to interact with it stimulates their interest, and they are more likely to obey and exercise with it. Pepper (Figure 2.1, Table 2.1) is a humanoid robot that meets the stated requirements. In accordance with the Uncanny Valley theory, humanoid robots that exhibit movement tend to evoke strong reactions from humans. The more these robots approach human appearance and behavior, the more pronounced the negative response from humans becomes. Pepper’s movement, behavior, and body structure are strikingly similar to those of humans. Despite this, Pepper’s appearance has been altered to resemble a toy or a general vision of what a robot should look like. That is why he produces positive responses from humans. The study Mahdi et al. [14] has demonstrated the significance of humanoid robot design and its evolution over time. They have also defined the various aspects of what a humanoid robot design should have. End-users should be involved in the design process, which is known as a co-design process, in which experts and non-experts both participate in a creative process, they have stated.

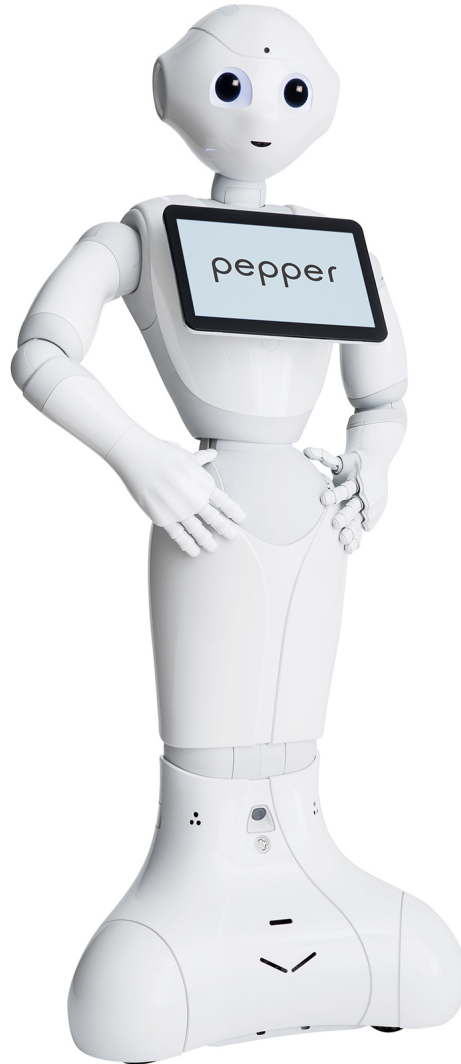


Figure 2.1: Humanoid robot Pepper [15]

Many job positions in the future are likely to include some form of human-robot interaction. As a result, incorporating robots into children's environments from an early age reduces the possibility of future animosity toward robots. The study Alam [16] talks about how demanding it is to incorporate robots into human lives from early on, but also that in the future, robots will assist teaching staff in their work with children.

Table 2.1: Technical specifications of a humanoid robot Pepper [15]

Size	Height: 120 cm, width 42,50 cm, length: 48,50 cm
Weight	128 kg
Autonomy	about 12 hours
Interaction sensors	4 microphones, 2 RGB HD cameras (mouth and forehead), 5 tactile sensors (3 in the head and 2 on the hands), touch screen on the breast
Motion sensors	1 3D camera behind the eyes for depth and detection of movements and obstacles. In the legs: 2 sonar, 6 lasers, one gyrosensor
Movement	3 omni-directional wheels
Maximum speed	5 km/h
Operating system	NAOqi OS

3 Motion Capture System

This section will introduce a motion capture system and its application for capturing human movement, specifically highlighting the OptiTrack system. It will also explain the process of capturing and post-processing data, providing insights into how the technology was utilized. Motion capture is the process of recording and tracking movement with the purpose of analyzing it. Data can be recorded in a variety of ways, but the two primary categories are visual and non-visual, Figure 3.1. Visual MCS rely on cameras and markers or computer vision techniques to precisely track and record the intricate movements of subjects or objects. Visual tracking systems can be classified into two categories: marker-based and markerless. Marker-based systems use cameras and retroreflective markers on the recording object for precise tracking, offering high accuracy. Active systems use special cameras that track light-emitting markers. Markerless systems, in contrast, use optical sensors or depth-sensing cameras to track body movements without attaching sensors. On the other hand, non-visual systems are classified into three categories: inertial, mechanical, and magnetic. Inertial sensors, including accelerometers and gyroscopes, provide orientation and position data suitable for home-based applications. Mechanical sensors, like goniometers, measure joint angles but require precise setup and expert alignment. Magnetic sensors use magnetic fields for tracking and offer good accuracy without line-of-sight issues but can be expensive and demand significant power. Each sensor technology has its advantages and disadvantages, making them suitable for different applications [17].

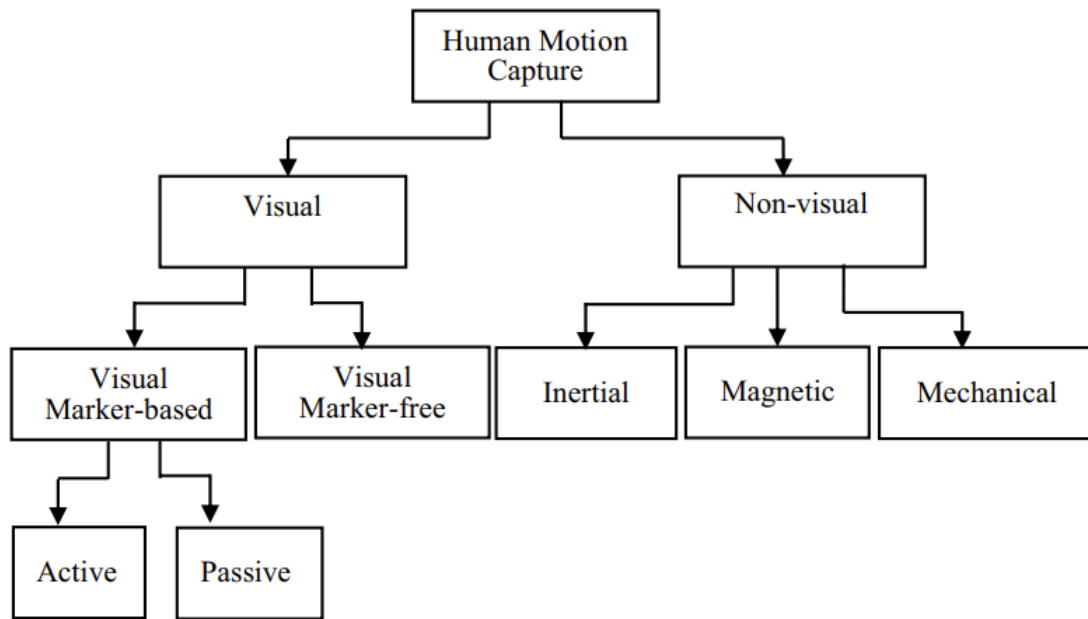


Figure 3.1: Classification of motion capture systems [17]

Applications for motion capture involve analyzing the motions of both living and non-living things, such as humans and animals [18], as well as mobile robots [19] and drones [20]. An overview of potential uses for optical motion capture systems of human movement is given below since the main focus of this thesis is the recording of human movements.

3.0.1. Applications of human motion capture

The study of human body movement captures the attention of professionals and experts from different fields due to its diverse applications. For instance, the study [21] focuses on the development of a system for recording movements of the upper body (arms) with the aim of improving the monitoring of rehabilitation results with precise recording of movements of the upper body and analysis of the results. On the other hand, motion capture technology has found commercial applications outside of the sphere of science, such as in the world of sports. The research discussed in [22], uses motion capture to analyze the movements of athletes during a game. An advantage of this approach is that, thanks to modern motion capture methods, the results can be monitored in real-time and analyzed quickly. On the other hand, a drawback of visual

markerless motion capture methods is their reliance solely on the quality of the cameras and the effectiveness of the computer vision algorithm, as no markers or devices are attached to the athletes. As previously stated, motion capture primarily serves the purpose of data acquisition. This function acts as the foundation for the creation of a database using marker-based motion capture systems. The study presented in [23] utilizes the mentioned premise. A database captured by motion capture systems is a valuable resource for developing and improving machine learning algorithms in the field of computer vision. Precisely trained models with valid data produce high levels of precision, potentially improving markerless techniques and related algorithms. Also, machine learning algorithms and motion capture devices have been integrated to map human movement into robot movement. The paper by Yang et al. [24] demonstrates how to use human arm motion data gathered by the VICON optical motion capture system and combine it with reinforcement learning techniques to produce more realistic robotic arm motions.

3.0.2. MCS in robotics

In terms of robotics in general, motion capture can be used as a data collection method. This strategy is becoming more popular as there is a greater requirement for human-robot interaction. It is important to note that collaborative robots and humanoids, which are frequently used in HRI, are designed in such a way that a person can work close to them without endangering their own health. Motion capture systems capture human movement, which is subsequently transferred to the robot via an imitation algorithm. There is a lot of research on the study of kinematics and the characteristics of human movements, including [21], [25]. Once there is a kinematic description of the human's motion, the next stage is to map that motion to the robot using an imitation algorithm. The imitation algorithm produces the joint space trajectories of the robot. The importance of mapping human movement into a robot's is that the robot can precisely portray captured human movement. Moreover, aligning robot movements with human behavior has security significance. When robots move in a way a person is accustomed to, a person can instinctively determine the robot's workspace, enhancing overall situational awareness. There has been research on the impact of the worker predictability of robot speeds and trajectories on worker productivity and sentiments,

such as the study [26]. In the mentioned study, they demonstrated that the low predictability of robot trajectories resulted in lower worker productivity and that the high speeds of these trajectories caused high levels of anxiety among workers. It confirms the hypothesis that people will accept a robot in their environment more readily if its motion resembles a human's. To summarize the issue of the intention of robot movements created by the example of human motions, they ensure the safety of humans at work and the cooperative interaction between robots and humans.

3.0.3. Description of the selected system

As previously noted, there are several motion capture methodologies. There are devices that offer direct information on an individual joint's motions, such as wireless devices for measuring movement tracking tracking [27]. Although no imitation technique is required when using these devices, the signal is not as precise as when employing a motion capture system with retroreflective spheres. When comparing any motion capture approach to visual marker-based motion capture, accuracy (in [mm]) is always lower, according to the study [28].

With that in mind, it was decided that OptiTrack [29] will be used for this thesis. It is a system comprised of 8 infrared Primex13 cameras (Figure 3.2, specifications given in Table 3.1), a suit, and retroreflective spheres for motion capture (referred to as 'markers' from now on).



Figure 3.2: Primex13 camera [30]

Table 3.1: Technical specifications of Primex13 camera [30]

Resolution	pixels	1280x1024
Frame rate	FPS	30-240
3D accuracy	mm	+/- 0,20
Range (passive markers)	m	16
Range (active markers)	m	25
Power	-	PoE

The suit was tailored to the volunteer who performed the movements. All of the movements were performed by a volunteer (age 25 years, height 178 cm, and weight 75.3 kg) and recorded at the Regional Centre of Excellence for Robotic Technology (CRTA), Zagreb. Cameras were positioned at a height of 3.5 meters in a rectangular pattern, as shown in Figures 3.3 and 3.4.



Figure 3.3: Camera setup view 1 (with cameras circled in red)



Figure 3.4: Camera setup view 2 (with cameras circled in red)

While the recording takes place, the 3D positions of the markers are logged by a program called Motive [2]. The positions of markers were recorded relative to a global frame that was placed on the ground, between the volunteer's legs. That way, the collected data is easy to examine because all of the values are logical for a human. Also, the collected data of all markers is calculated with respect to the same global frame. The data was collected at a rate of 120 Hz.

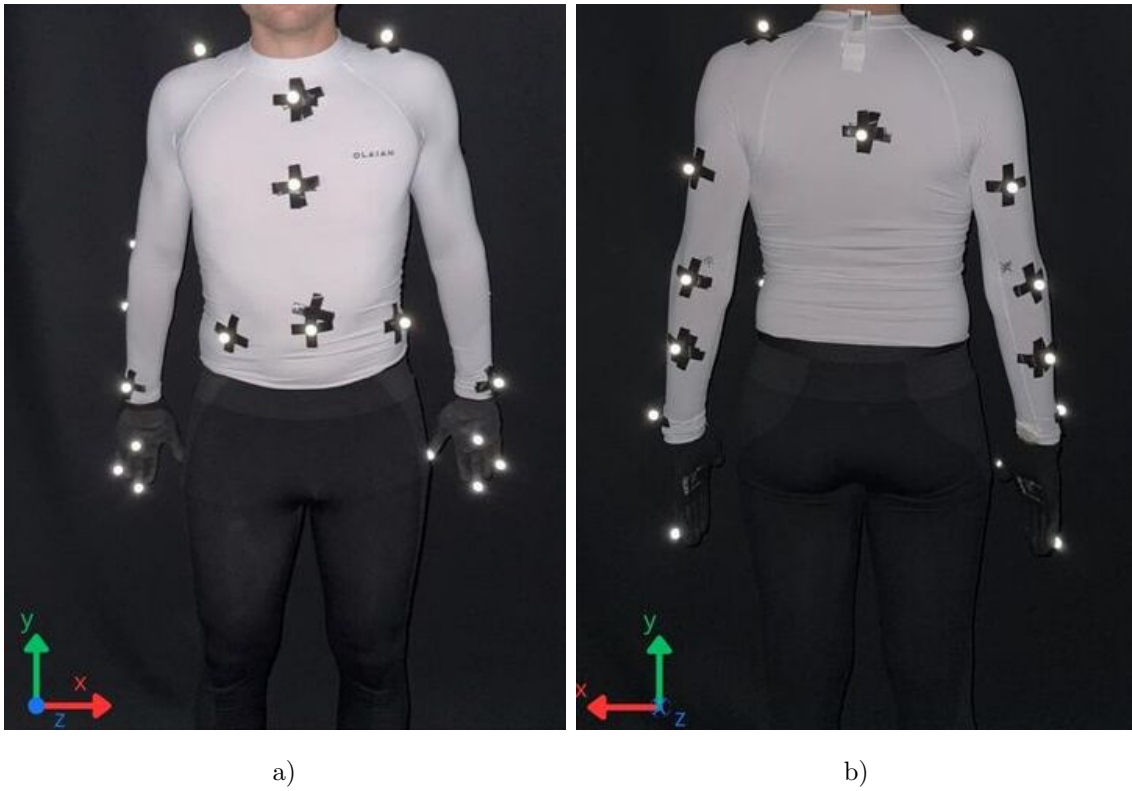


Figure 3.5: (a) Marker positions on the body's front (b) Marker positions on the body's back

The markers were positioned as shown in 3.5. They were placed on both joints and limb segments, and their locations were chosen such that each arm's seven degrees of freedom could be characterized. It is important to note that each recorded movement started and finished in a position from now on referred to as a 'zero position', Figure 3.6. A person is in a zero position when they extend their arms in front of them and match the zero position of the robot, where all joint values equal zero.



Figure 3.6: Zero position

3.0.4. Post-processing of the data

After recording the movements, post-processing should be performed to remove any signal defects [31]. Post-processing involves error elimination, labeling, interpolation for gap-filling, and additional steps such as handling occlusions and correcting signal peaks, ensuring a refined and accurate dataset for subsequent analysis and application. The data obtained by recording is, as previously mentioned, the 3D position of the marker at a particular moment in time (Figure 3.7 a)). For the benefit of an analysis, the markers are grouped into marker sets (3.7 b)).

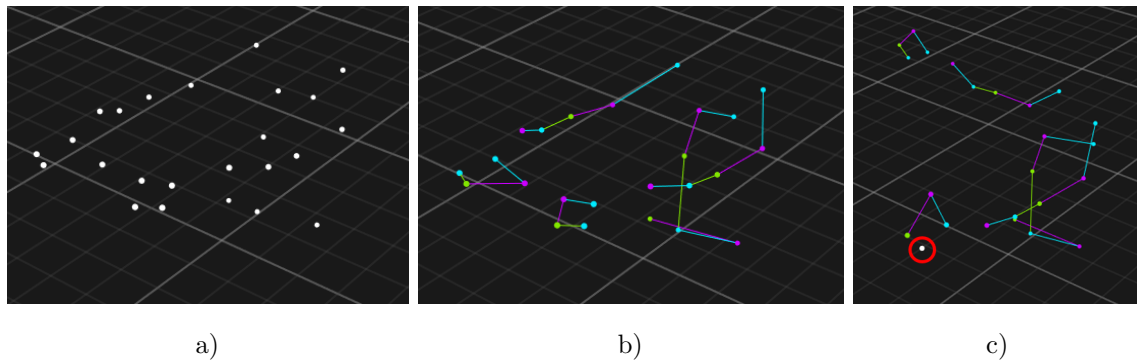


Figure 3.7: (a) Marker before editing (b) Labeled marker sets (c) Marker with a lost label

There were five marker sets overall in the recordings made for this thesis, one for each hand and arm and one for the torso. The hand marker set includes markers for the thumb, middle finger, and little finger, as well as the center of the hand. Shoulder, upper arm, elbow, forearm, and wrist joint markers are included in the arm marker set. The torso marker set includes markings for the back, neck, chest, and navel, as well as one marker for each hip. So, a total of 24 markers were placed on the individual, with the goal of describing the movements of the person's arms, hands, and torso. In order to reconstruct the 3D position of an individual marker, at least two cameras must see the marker at any moment. Figure (3.8) depicts a camera location relative to a human while recording movement.

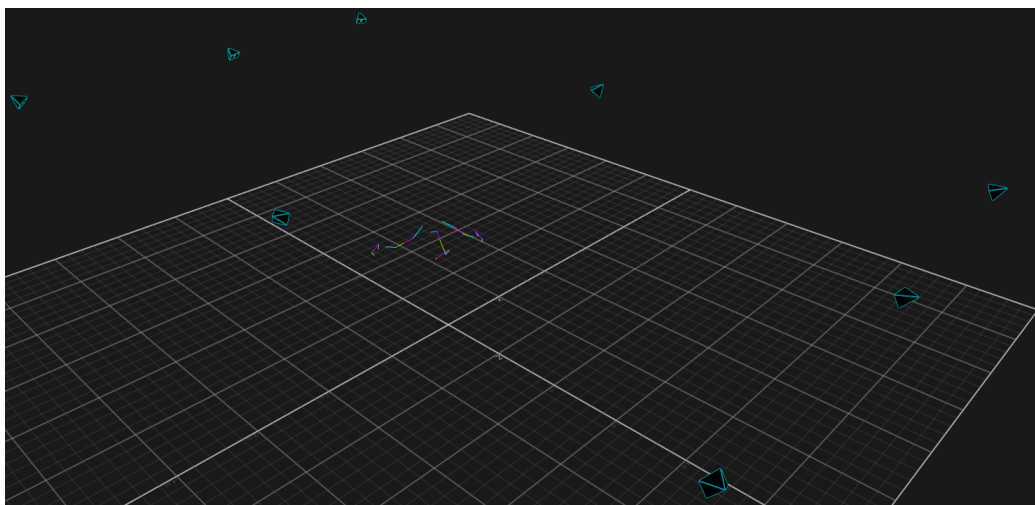


Figure 3.8: The position of eight cameras in relation to the human

To begin, all of the data is examined for potential errors (such as signal noise (Figure 3.11) or marker occlusion (Figure 3.7 c)). Following that, markers are labeled in order to link exported data to a specific body part. Then the data is processed once more by interpolating all of the gaps in the signal. Finally, the signal is exported to be applied (in this case, as an input to the imitation method). When recording the human body, the marker is often occluded due to large variations in the rotations of individual joints (for example, the palm facing the ground one moment and facing the sky the next). Also, due to the high rate of recording frequency, there is a loss within a couple of frames, which is less than a second in real time (3.7 c)).

However, there is an option to fill those gaps within the software itself. But first, you have to go throughout the entire take and re-assign marker IDs after each occlusion. Information about the marker's ID is usually defined at the first moment of recording, and when information about the marker's position is lost, it must be redefined by linking a label and marker. After all of the unlabeled markers have been re-assigned values, the gap-filling phase begins. The entire take is observed gap by gap, and a small part of the trajectory is trimmed at each gap. If the marker position information is lost, a small inaccuracy is introduced at the last moment when the marker existed, which is undesirable during data interpolation ((3.9)). It is important to note that in the graphs, the data in red represents the x-axis, the y-axis in green, and the z-axis in blue. After all of the gaps have been trimmed, it is time to fill them. If a small portion of the trajectory is missing, a cubic polynomial can be used to interpolate it. There is also the option of interpolation with a constant or a linear function; however, because human movements are sinusoidal in nature, such functions do not adequately depict the movement. If a significant part of the signal is missing, then it is reconstructed using other markers. For example, if the information regarding the little finger marker is lost, it will be reconstructed based on the movement of the center of the hand, thumb, and middle finger during that time period. This strategy has shown very good results because the sets of markers that are defined in these exercises often do not alter significantly in their mutual spatial relationship.

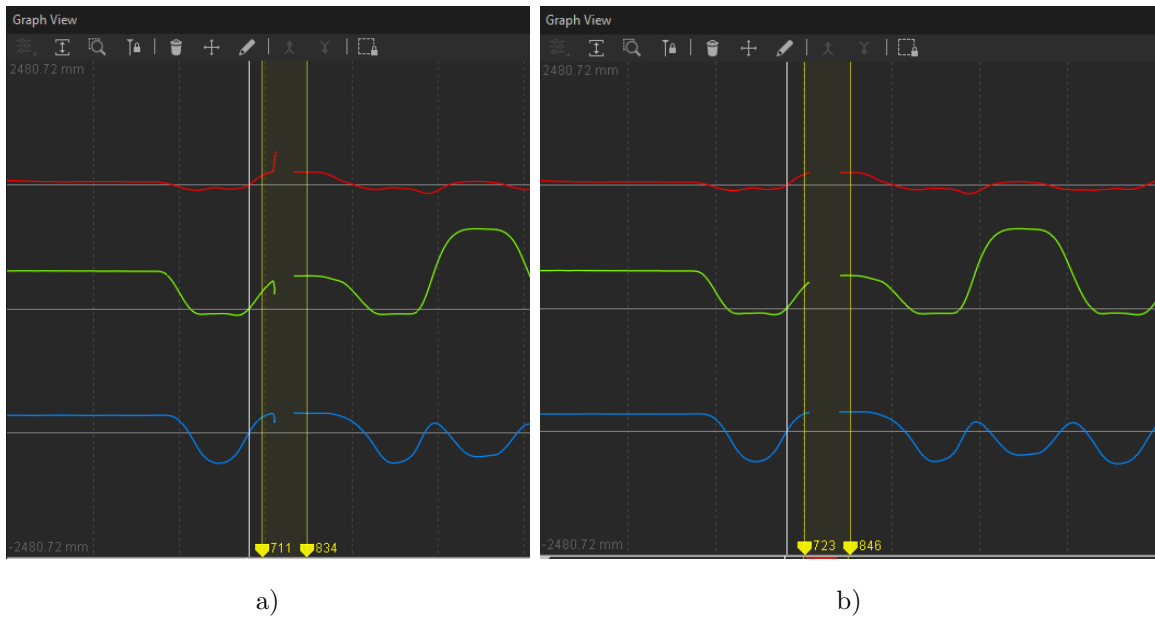


Figure 3.9: (a) Untrimmed data (b) Trimmed data

Figure 3.10 a) illustrates what happens when the incorrect type of interpolation is chosen. The gap is too large to use cubic interpolation here and the data was reconstructed with a large error. The desired interpolation was reached by selecting signal reconstruction using additional markers 3.10 b). Figure 3.7 c), 3.7 c), 3.7 c) demonstrates the process of interpolating with the incorrect model, removing that data, then interpolating with the correct model.

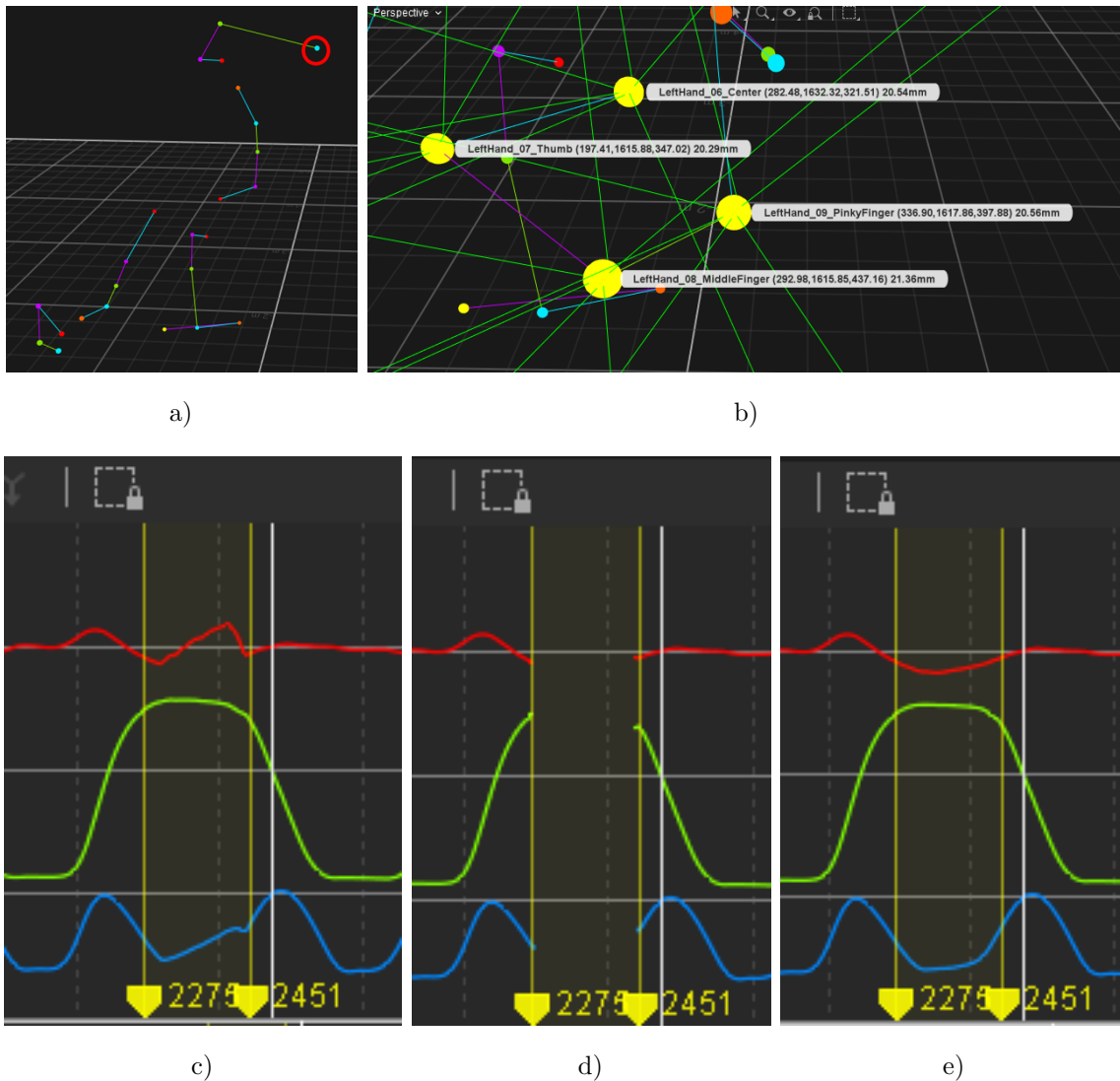


Figure 3.10: (a) Error due to the wrong interpolation method (b) Hand marker set (c) Error in the data due to the wrong interpolation method (d) Deleted part of the data (e) Correctly reconstructed data

Another error that can occur when recording data is the appearance of a peak in a signal, and this error can be attributed to noise and a high recording frequency. In this case, the locations of the errors must be identified, and that portion of the recording is deleted, with the trajectory of the human movement interpolated on this section. This has no effect on the movement because such errors result in the deletion of a small amount of data, and the movement, by its nature, still accurately represents human

movement ((3.11)). It is important to eliminate large peaks at this stage, and later the signal will be additionally smoothed using the Savitzky-Golay filter, [32]. It is a digital filter that removes the noise from the signal and is used to avoid incorrectly recorded data.

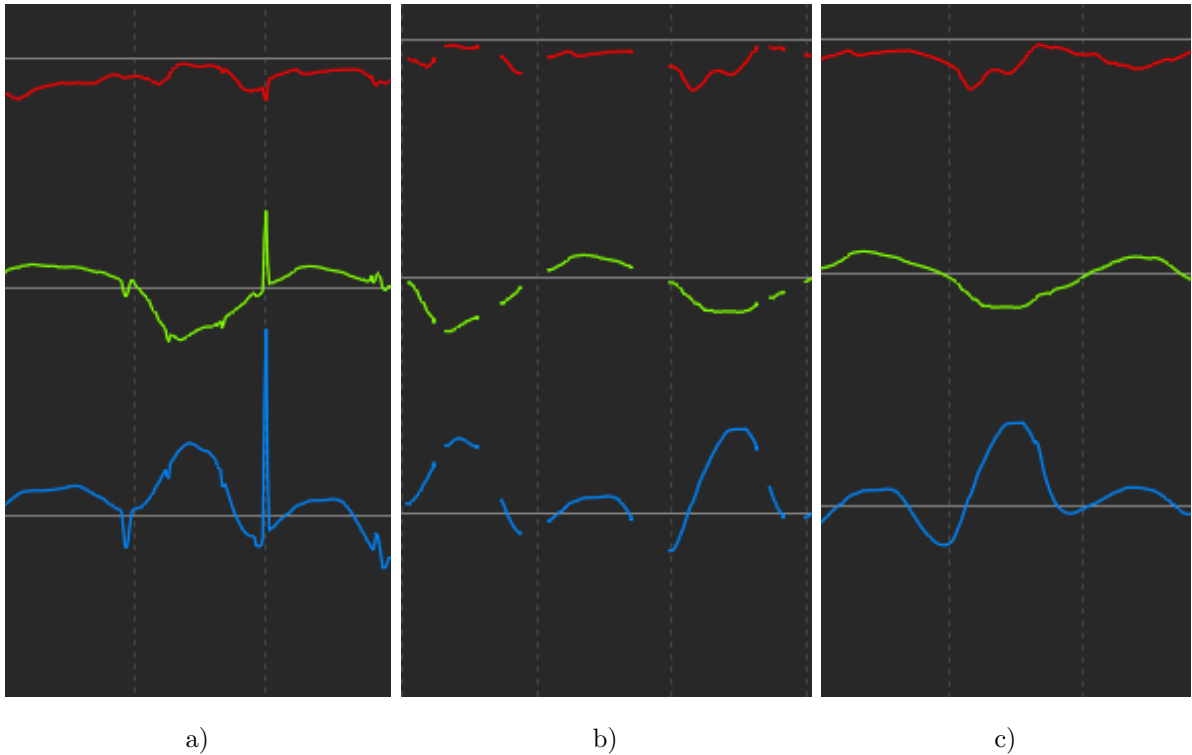


Figure 3.11: (a) Error due to the peak of a signal (b) Deleted part of the data (c) Correctly reconstructed data

3.0.5. Selected exercises

A set of five exercises (Figure 3.12) was recorded with the intention of being replicated by the robot. They focused on the upper body, with an emphasis on the arms and hands. All of the five exercises were executed with the exchange of the pointing direction of each arm alternately. The orientation of the hand is the focus of the exercises and requires the greatest attention when performing them. The movements are asymmetric, and the hands move in different directions at the same time, but the orientation of the hand, which also changes during the execution of the exercise, brings additional weight to the exercises.

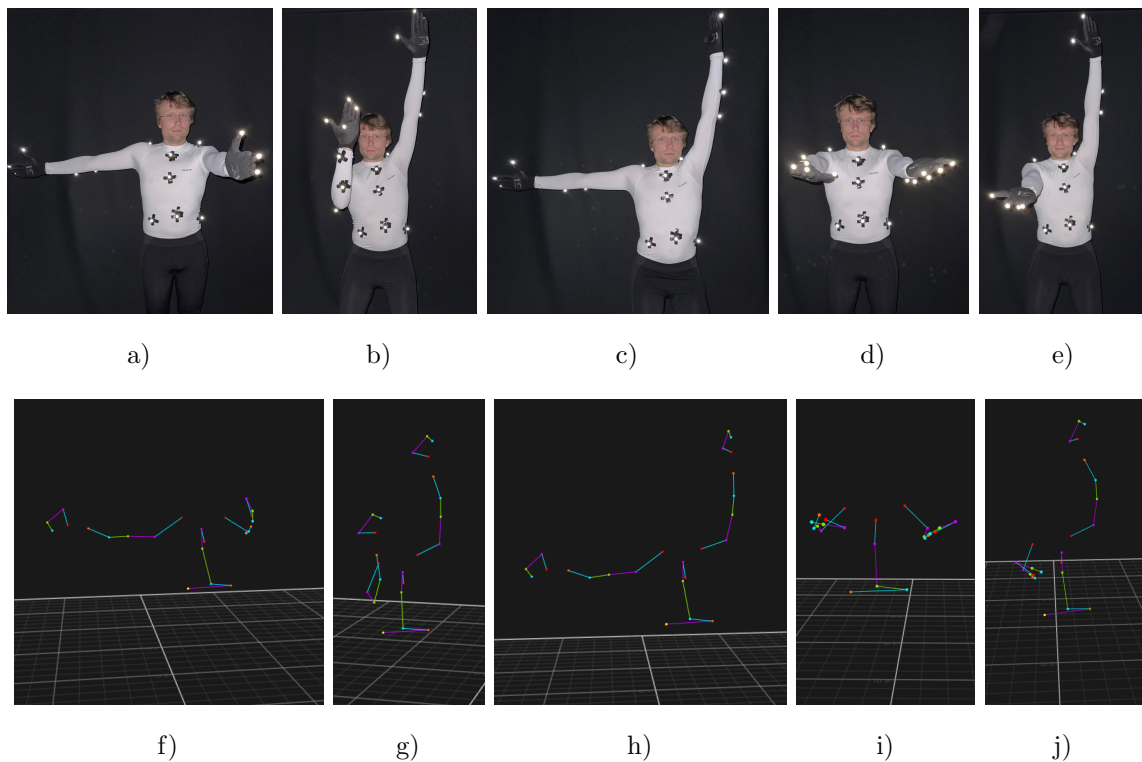


Figure 3.12: Exercises captured using a smartphone camera (top row) and the OptiTrack system (bottom row). (a) The first exercise was a motion where one arm is reaching in front of the person and the other is reaching to the side. (b) The second exercise was a motion where a person starts with both arms reaching to the side, after which the person reaches to the sky with one arm while flexing the other arm at the elbow, creating a motion of 90 degrees in front of the person. (c) The third exercise consisted of one arm reaching in front of the person and the other reaching towards the sky. (d) The fourth exercise starts in the initial position. After the initial position, the arms move downwards until they point a bit behind the person, and then they go to the front, but the rotation of hands is opposite (one hand points towards the sky and the other towards the ground). (e) In the last exercise, one arm points towards the sky while the other points in front of the person. (f), (g), (h), (i), (j) Analog poses from recorded motion displayed in Motive software

Each arm's pointing orientation was alternatively switched throughout the execution of each of the five exercises. What was really important while coming up with the exercises to be recorded was to consider whether a robot would be able to replicate

them. The chosen robot was a humanoid robot, Pepper. Figure Degrees of freedom in Pepper's body are depicted by Figure 3.13, and the ones relevant to this thesis are the 2 DOF in the neck, the 5 DOF in each arm, and the 2 DOF in the hip.

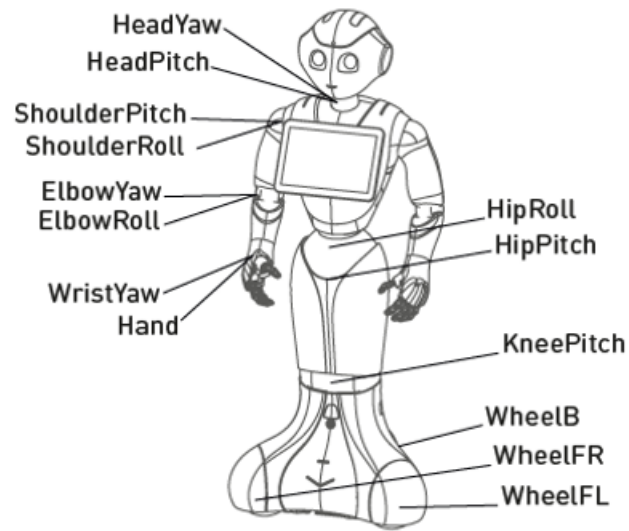


Figure 3.13: Pepper's degrees of freedom [33]

4 | Kinematic Model of a Robot

4.0.1. Robot kinematics

This section will explain the importance of describing a robot's kinematic model using Modified Denavit-Hartenberg (MDH) parameters. It will then introduce a kinematic model employed to describe Pepper, a humanoid robot. The purpose of kinematics in robotics is to use mathematical laws to describe the structure of the robot so that it may be driven using control algorithms. Before moving on to the topic of kinematics, the most important terminology will be explained. The first term is link, which is a rigid component that makes up the robot's structure. A joint connects two links and restricts the degrees of freedom between them. Joints are classified as revolute or prismatic. The motion between the two links is limited to a rotation along the common axis by a revolute joint. A prismatic joint, on the other hand, limits the motion between the two links to translation along the common axis. Finally, joint and task space are essential terms for enabling accurate control and planning of robot motions. The location of all of the robot links is defined in joint space using joint angles, which is represented using joint coordinates (individual motor rotation). The position and direction of the robot's end-effector are specified in task space.

In robotics, the link between joint space and task space is essential. Direct and inverse geometric models define this relationship; the output of a direct geometric model is the position of the robot in task space as a function of its position variables in joint space. The inverse geometric model is the inverse of the known position of the end-effector in the task space, and it outputs the vector of the robot's coordinates in the joint space for the known location of the end-effector in the task space.

To transfer recorded human motion to the robot, a model for the transformation between the joint and the task space is required. The structure of a robot must be determined before building a mathematical model of it. Khalil et al. [34] propose categorizing robotic structures as serial link, tree-structured, and close-looped structures. After deciding on the robot's structure, the next stage is to decide on the model representation that will be used to describe it. The most common is a model defined using Denavit-Hartenberg parameters (DH in future references), as described in the study [35]. This model has some faults, and scientists have proposed revisions and enhancements to address them over the years. Hayati et al. [36], for example, describe modified Denavit-Hartenberg parameters (MDH in future references). The DH model has four parameters, and the MDH model requires two more. By adding two more parameters, MDH model overcomes the problem of singularity, which occurs when two adjacent joints are close to a parallel state or are parallel to each other. According to the study [37], DH and MDH models are equivalent in the case of a serial link robot structure, with the main variation being the assignment of coordinate systems.

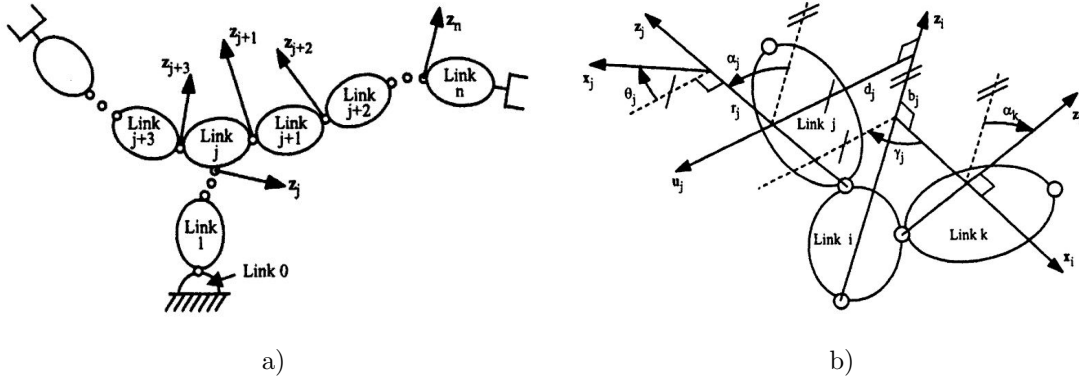


Figure 4.1: (a) Tree-structured robot notations [34] (b) Tree-structured robot MDH parameters definition [34]

There are four stages to describing a robot's kinematic model using MDH parameters. First, each joint's frame is defined using a set of rules. The z_j is an axis of rotation along the joint j , and the x_j axis is aligned with the common normal between z_j and z_{j+1} . Finally, the right-hand rule determines the y_j axis. The coordinate system's origin is defined as the intersection of the x_j and z_j axes. The values of the parameters are then determined. There are six parameters in the case of a tree-structure robot (which

is a case in this thesis because a humanoid robot is used). When x_{j-1} is positioned along the common normal between z_{j-1} and z_j , four parameters (α , d , θ , and r) are sufficient to define ${}^i\mathbf{T}_j$. To calculate the transformation from frame i to frame j , α_j , d_j , θ_j , and r_j have to be determined. α_j is the angle between z_{j-1} and z_j about x_{j-1} , d_j is the distance between z_{j-1} and z_j along x_{j-1} . θ_j is the angle between x_{j-1} and x_j about z_j , r_j is the distance between x_{j-1} and x_j along z_j . In another case, when x_i is along the common normal between z_i and z_j , two additional parameters (γ , b) are required. To calculate the transformation from frame i to frame j (${}^i\mathbf{T}_j$), α_j , d_j , θ_j , r_j , γ_j , and b_j have to be determined. α_j , d_j , θ_j , and r_j are determined as they were in the previous case, but to determine γ_j and b_j , a vector u_j has to be defined first. It is defined as a common normal between z_i and z_j . γ_j is the angle between x_j and u_j about z_j and b_j is the distance between x_j and u_j along z_j . Figure 4.1 a) shows how joints are notated when describing the kinematics of a tree-structured robot, and figure 4.1 b) shows how parameters are determined.

After filling out the entire table with parameters, it is time to acquire the transformation (translation and rotation) of each frame. It is represented by a homogeneous transformation matrix (Equation (4.1)), where vectors ${}^i\mathbf{s}_j$, ${}^i\mathbf{n}_j$, and ${}^i\mathbf{a}_j$ represent rotation and vector ${}^i\mathbf{P}_j$, represents translation.

$${}^i\mathbf{T}_j = \begin{bmatrix} \mathbf{s}_x & \mathbf{n}_x & \mathbf{a}_x & \mathbf{P}_x \\ \mathbf{s}_y & \mathbf{n}_y & \mathbf{a}_y & \mathbf{P}_y \\ \mathbf{s}_z & \mathbf{n}_z & \mathbf{a}_z & \mathbf{P}_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

Equation (4.2) is used to obtain the transformation, with the remark that in a case where only four parameters are used to define the frame position and orientation, variables γ_j and b_j must be set to zero. It should be noted that in this notation, S stands for sinus function and C stands for cosinus function. $C\theta_j$, for example, denotes the cosine of θ_j .

$$\begin{aligned}
{}^i\mathbf{T}_j &= \mathbf{Rot}(z, \gamma_j) \mathbf{Trans}(z, b_j) \mathbf{Rot}(x, \alpha_j) \mathbf{Trans}(x, d_j) \mathbf{Rot}(z, \theta_j) \mathbf{Trans}(z, r_j) = \\
&= \begin{bmatrix} C\gamma_j C\theta_j - S\gamma_j C\alpha_j S\theta_j & -C\gamma_j S\theta_j - S\gamma_j C\alpha_j C\theta_j & S\gamma_j S\alpha_j & d_j C\gamma_j + r_j S\gamma_j S\alpha_j \\ S\gamma_j C\theta_j + C\gamma_j C\alpha_j S\theta_j & -S\gamma_j S\theta_j + C\gamma_j C\alpha_j C\theta_j & -C\gamma_j S\alpha_j & d_j S\alpha_j - r_j C\gamma_j S\alpha_j \\ S\alpha_j S\theta_j & S\alpha_j C\theta_j & C\alpha_j & r_j C\alpha_j + b_j \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{4.2}$$

Finally, transformation matrices are multiplied to obtain a link between the robot's base (labeled with 0) and end-effector (labeled with n) in Equation (4.3).

$${}^0\mathbf{T}_n = {}^0\mathbf{T}_1 \cdot {}^1\mathbf{T}_2 \cdot \dots \cdot {}^{n-1}\mathbf{T}_n \tag{4.3}$$

4.0.2. Kinematics of humanoid robot Pepper

A humanoid robot Pepper has 20 DOF (Figure 3.13). For the purpose of this thesis (upper body imitation algorithm and body pose estimation), 15 rotational joints were used, as presented in Figure 4.2. Each degree of freedom was represented with a revolute joint. When analyzing frames, Pepper's body was considered to have three separate kinematic chains. The start of each kinematic chain is at the base; in this case, that is the torso of the robot. So, the first kinematic chain starts in the torso and ends with Pepper's head. It contains five rotational joints: three in the torso and two in the neck. The second and third also start in the torso and end with the wrist of each hand. Each one of them contains eight rotational joints, three in the torso and five per arm. The main distinction between studying a tree-structured robot and analyzing a serial link structure is the analysis of the link that connects all three chains (link j depicted in Figure 4.1 a)).

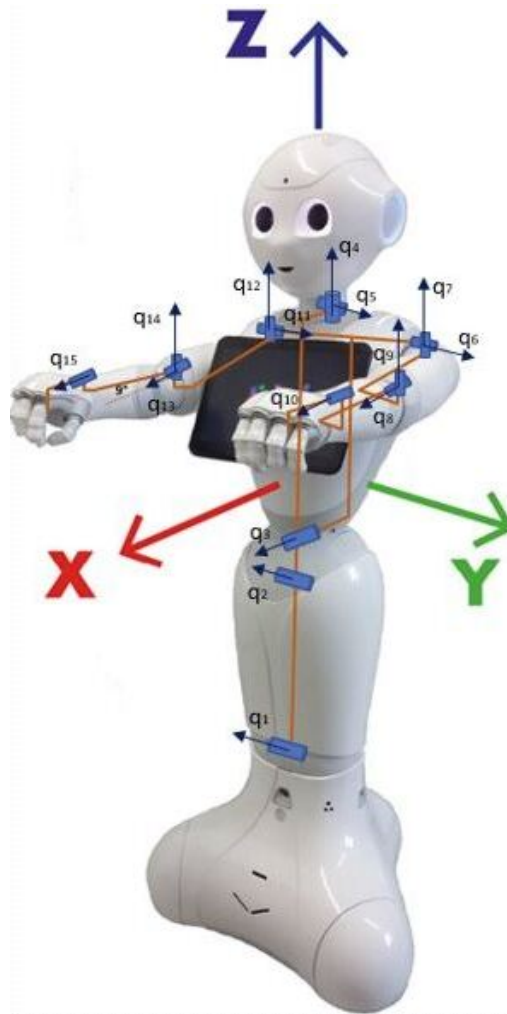


Figure 4.2: Pepper's kinematic configuration [33]

Because the robot's design causes a shift in q_6 and q_{11} (Figure 4.2), MDH parameters were used to create Pepper's kinematic model. That shift creates a case of parallelism between two adjacent axes, and two more parameters (γ , b) are required to define the kinematics of the robot so that there is no instance of singularity in the movement. Table 4.1 displays six MDH parameters for Pepper, a humanoid robot, and also two additional parameters: 'joint' is used to numerate each joint, and 'ans' is used to show which joint was before the numerated one. For example, to calculate the transformation matrix of the first joint, it was done with respect to the base of the robot (${}^0\mathbf{T}_1$). But the transformation matrix of the sixth joint is calculated with respect to the third joint (${}^3\mathbf{T}_6$) because Pepper is a tree-structured robot.

Table 4.1: MDH parameters of a humanoid robot Pepper

joint	ans	α [rad]	d[mm]	θ [rad]	r[mm]	γ [rad]	b[mm]
1	0	$\pi/2$	0	$\pi/2 + q_1$	0	0	0
2	1	0	268	$0 + q_2$	0	0	0
3	2	$\pi/2$	79	$\pi/2 + q_3$	0	0	0
4	3	$\pi/2$	-38	$-\pi/2 + q_4$	308,9	0	0
5	4	$-\pi/2$	0	$-\pi/2 + q_5$	0	0	0
6	3	$-\pi/2$	-57	$0 + q_6$	149,76	0	86,82
7	6	$\pi/2$	0	$\pi/2 + q_7$	0	0	0
8	7	$\pi/2$	15	$0 + q_8$	181,2	0	0
9	8	$-\pi/2$	0	$0 + q_9$	22,97	0	0
10	8	$\pi/2-0,157$	23,6	$\pi/2 + q_{10}$	150	0	0
10'	10	0	-30,3	0	69,5	0	0
11	3	$-\pi/2$	-57	$0 + q_{11}$	-149,74	0	86,82
12	11	$\pi/2$	0	$\pi/2 + q_{12}$	0	0	0
13	12	$\pi/2$	-15	$0 + q_{13}$	181,2	0	0
14	13	$-\pi/2$	0	$0 + q_{14}$	22,97	0	0
15	14	$\pi/2-0,157$	-23,6	$\pi/2 + q_{15}$	150	0	0
15'	15	0	-30,3	0	69,5	0	0

After obtaining a robot's MDH parameters, a homogenous transformation matrix for each DOF is determined. That takes a long time because it is a repetitive task that must be repeated for every joint; in the case of this thesis, it adds up to 15 times. As a result of Khalil et al.'s research, a software package called SYMORO+ was created [38]. It generates nearly all of the symbolic models required for robot simulation, control, identification, and design. SYMORO+ can also create symbolic models for serial link, tree-structured, and closed-loop robot systems. In addition to MDH settings, it requires the additional parameters listed in Table 4.2. The parameter notation is the same as that defined in [34]. Parameter μ is equal to 1 if joint is actuated, and equal to 0 if it isn't. Parameter σ is equal to 1 if joint is prismatic, and equal to 0 if it is revolute. We used SYMORO+ to generate transformation matrices.

Table 4.2: Additional parameters for SYMORO+ software input

joint	ans	μ	σ
1	0	1	0
2	1	1	0
3	2	1	0
4	3	1	0
5	4	1	0
6	3	1	0
7	6	1	0
8	7	1	0
9	8	1	0
10	8	1	0
10'	10	0	-
11	3	1	0
12	11	1	0
13	12	1	0
14	13	1	0
15	14	1	0
15'	15	0	-

5 | The Imitation Algorithm

This section will describe the algorithms used to perform a conversion process for a humanoid robot that mimics human dual arm motion. The data gathered from recording a human with a motion capture system is used to define the characteristics of the human body and its natural movements. Movement can be defined in task or joint space. The recorded movements are in task space, but we wish to retrieve them in joint space so that they may be performed by a humanoid robot. The kinematics of a human body have to be determined. In the presented thesis, it was accomplished by adding two more degrees of freedom in each wrist to a kinematic model of the robot Pepper. Furthermore, in transformation matrices that describe an extended kinematic model of a robot, the size of the robot's segments was adjusted to match the size of the actor's limbs by setting them to a value of the actor's segment dimensions. Figure 5.1 shows the kinematic model of a human, derived by extending a robot's kinematic model. In this thesis, the primary focus of an imitation algorithm is on trajectory representation using generalized coordinates while adhering to the robot joint boundaries. Generalized coordinates from a human kinematic model were employed directly to describe the movement of a robot.

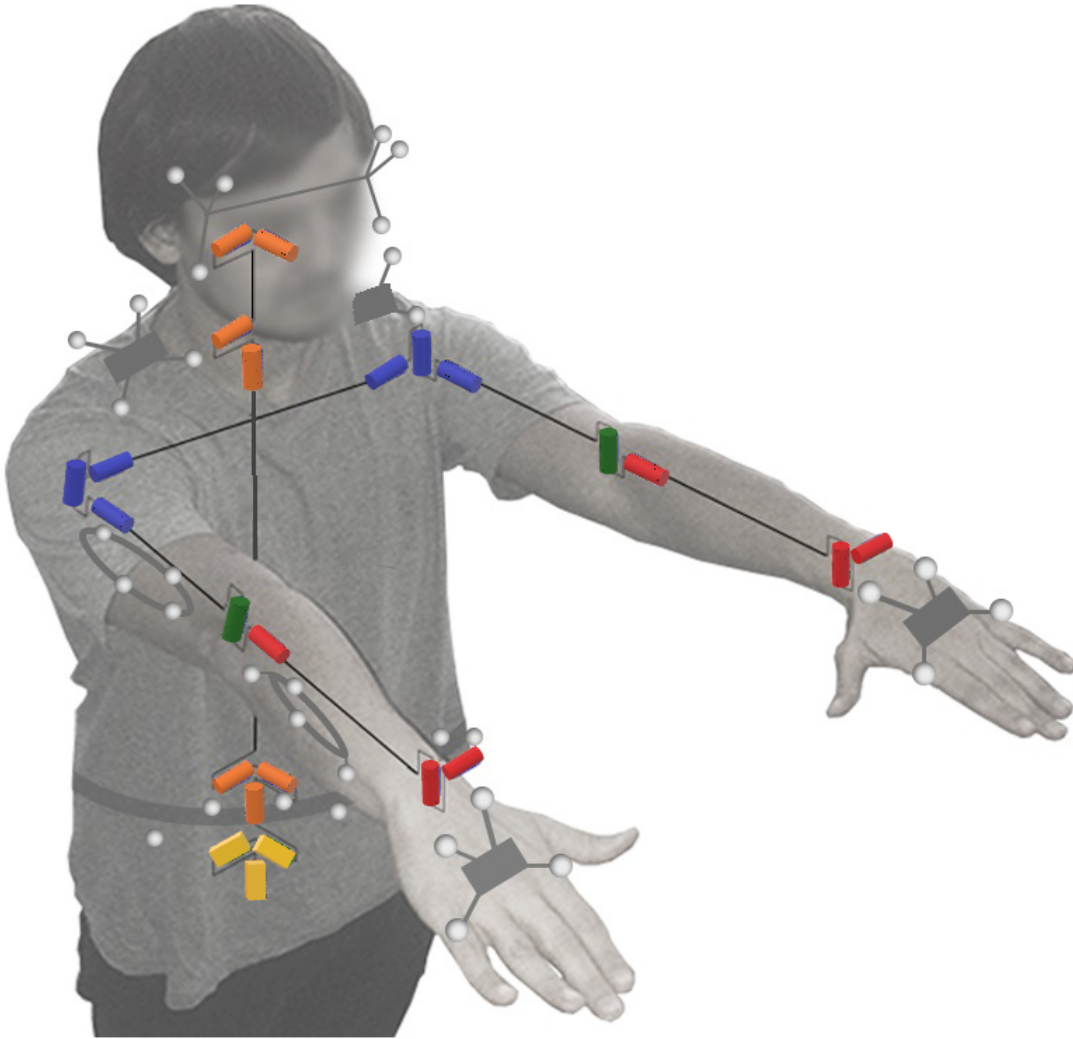


Figure 5.1: Kinematic model of an upper body of a human [39]

5.0.1. Initialization algorithm

An initialization algorithm specifies the basic structure of a robot's scaled model and is used to attach virtual frames to it, which will mimic the motion of real marker frames in imitation process. In this algorithm, the initial position of an actor is transformed from task to joint space. The reason for the initialization was to avoid discrepancy between the robot's zero position and the initial location of a movement. It also prevents any jerk motions that may occur. As previously stated, the markers were placed on the volunteer while capturing a human movement. From here on, these will be referred to as

real markers because they record information regarding the actual positions of markers while performing movements. Virtual markers, on the other hand, are those defined in a human's kinematic model. They are introduced later in the imitation process to enable the minimization algorithm.

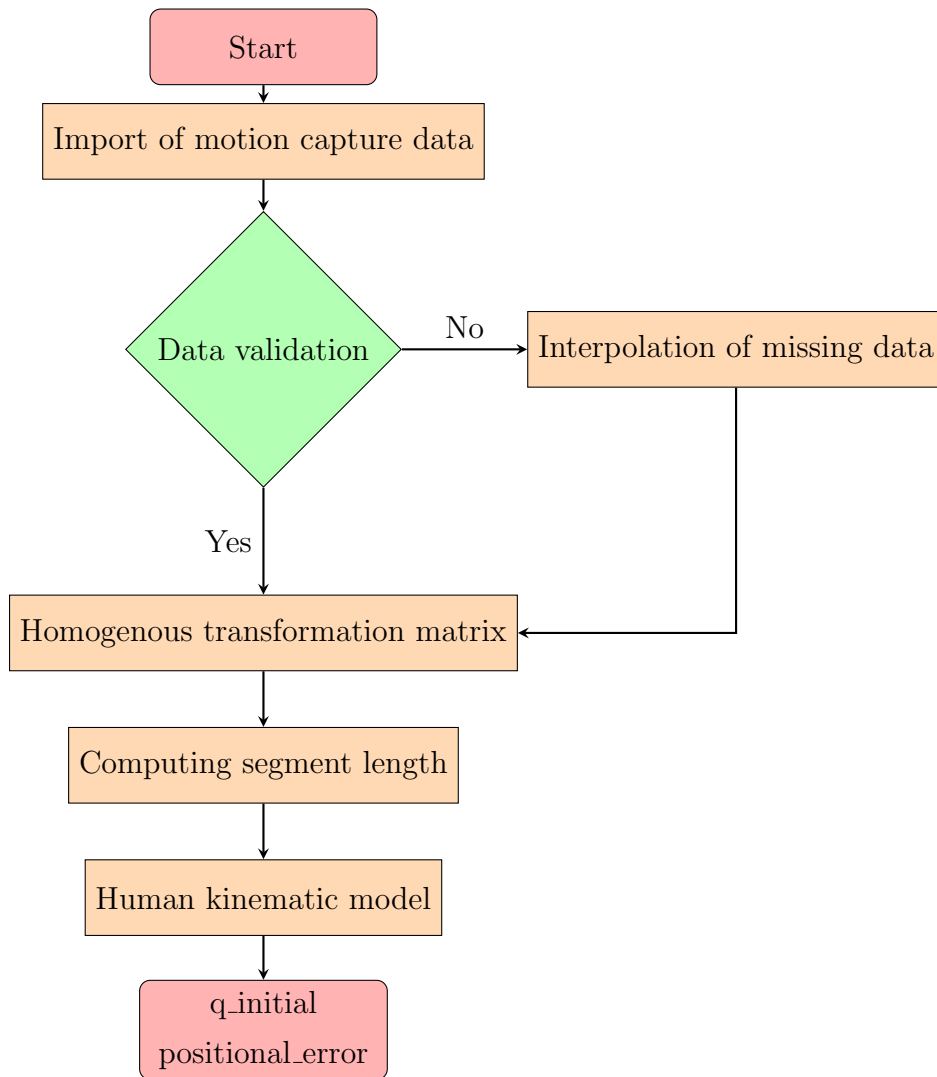


Figure 5.2: Initialization algorithm flowchart

An initialization process flow diagram is shown in Figure 5.2. Data from recorded motion capture is imported and verified at the beginning of the algorithm. If gaps remain in the motion capture data postprocessing procedure, this indicates an error. If that is the case, the algorithm will not function appropriately since each marker will have a different total number of data points characterizing the movement. The choice

must be made; if the data is accurate, the algorithm will proceed; if not, interpolation must be completed first. In this instance, there were no data mistakes because Motive software was used to postprocess the data. If errors occurred, they would only affect a few samples. Because of this, interpolation could be carried out without worrying about losing data regarding the characteristics of human movement (for instance, using a cubic polynomial). The initial step in performing computational operations on the data is to write down the positions of the markers in time as homogenous transformation matrices. The rotational and translational components make up the information contained within the transformation matrices. Because there is no information about the rotation (because we are discussing the position of the marker's center, which is geometrically defined as a point), the rotation part is given as a unit matrix. The marker's position along the x, y, and z axes is represented by a spatial vector (the last row of the homogeneous matrix of transformations). Because the recorded data contains a significant number of sequences (keep in mind, it was originally recorded at 120 Hz), the data was saved in a 3D matrix via a loop. The notion of three-dimensional matrices is illustrated in Figure 5.3, where the first information is the number of rows, the second information is the number of columns, and the third information is the number of matrices in that record.

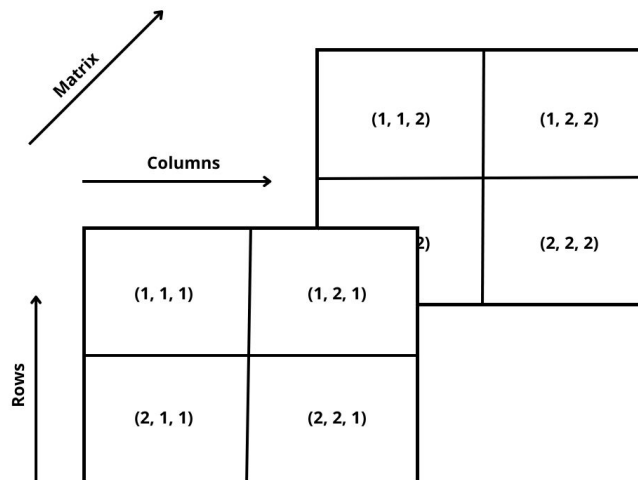


Figure 5.3: Illustration of 3D matrix

To define a scaled model of the robot, i.e., a kinematic model of a human based on a humanoid model, the lengths of the segments must be calculated from the marker positions. This was done at the beginning of each motion, when the actor remained in

the starting position for several seconds without moving. The table 5.3 illustrates which markers were used to determine the length of each section.

Table 5.1: Overview of the markers utilized to measure the length of each section

Segment	Markers used (name)	Markers used (number, Figure 5.4)
Shoulder width	Left Shoulder, Right Shoulder	1, 2
Upper arm	Right Shoulder, Right Elbow	1, 3
Forearm	Right Elbow, Right Wrist	3, 4
Hand	Right Wrist, Right Middle Finger	4, 5
Trunk	Navel, Neck	6, 7



Figure 5.4: Markers used in defining the segment lengths of a human

It is also worth noting that the length was determined using the Euclidean norm (5.1). The straight-line distance between two locations in a space of Euclidean geometry,

such as a two-dimensional or three-dimensional space, is calculated using the Euclidean distance formula. For three-dimensional space with points (x_1, y_1, z_1) and (x_2, y_2, z_2) , the Euclidean distance is represented by the equation:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (5.1)$$

Then, using the segment lengths obtained in the previous phase, the extended kinematic model of the robot, which represents the kinematic model of the human, is defined. It is also worth noting that the data was converted from milliliters to meters so that the findings could be interpreted more easily later in the simulation. The Matlab function 'fmincon' is utilized for the algorithm's optimization. 'fmincon' is an abbreviation for 'function minimization with constraints'. Its main goal is to determine the minimum of a constrained nonlinear multi-variable function. The following data was utilized as input to the function: the criterion function, variable boundaries (upper and lower), and the initial assumption. The error between the positions of real and virtual markers is defined as the criteria function, and the purpose of this approach is to minimize that error to zero. The positions of the real and virtual markers are thus equalized, and the generalized coordinates are derived with the inverse kinematic algorithm using the previously specified geometric kinematic model of a human. Upper and lower boundaries are defined as a range of angles describing the robot's movement restrictions, defined in Table 5.2.

Table 5.2: Boundaries of Pepper's joints (left and right arm) [40]

Joint	Boundaries [°]	Boundaries[rad]
Left Shoulder Pitch	-119,5 ... 119,5	-2,0857 ... 2,0857
Left Shoulder Roll	0,5 ... 89,5	0,0087 ... 1,562
Left Elbow Yaw	-119,5 ... 119,5	-2,0857 ... 2,0857
Left Elbow Roll	-89,5 ... -0,5	-1,562 ... -0,0087
Left Wrist Yaw	-104,5 ... 104,5	-1,8239 ... 1,8239
Right Shoulder Pitch	-119,5 ... 119,5	-2,0857 ... 2,0857
Right Shoulder Roll	-89,5 ... -0,5	-1,562 ... -0,0087
Right Elbow Yaw	-119,5 ... 119,5	-2,0857 ... 2,0857
Right Elbow Roll	0,5 ... 89,5	0,0087 ... 1,56
Right Wrist Yaw	-104,5 ... 104,5	-1,8239 ... 1,8239

The robot's range of motion is limited in comparison to that of a human, which must also be considered. Because all motions are recorded starting from the 'zero position' (Figure 3.6), which corresponds to all the robot's rotations being equal to zero, the first estimate of the initialization process is set to a vector with all terms equal to zero. The initialization algorithm produces a vector that defines the generalized coordinates of the first moment of recording, which is later used as input for the imitation method, which is executed for each recorded movement. It is also crucial to note that the initialization algorithm only looked at markers placed in joints, whereas the imitation algorithm looked at all markers, including joint and segment markers.

5.0.2. Imitation algorithm

Using inverse kinematics, an imitation algorithm is used to transfer the entire recorded movement from the human to the robot. Figure 5.5 demonstrates imitation algorithm flowchart. The initial task is to run the loaded marker positions through the Savitzky-Golay filter, as previously described in the thesis. The trajectory is smooth because of this. The simulation process continues following the initialization procedure and human kinematic model, with the segment lengths remaining constant. There is, however, some variation in the markers whose positions are analyzed. To better reflect the nature of

human movements, markers placed on segments when recording movements with the motion capture system are also considered in imitation algorithm. The relationship between joint and segment markers should be determined and kept fixed during the motion in order to obtain and follow the motion of the real markers by virtual markers while computing the optimal values of the generalized coordinates. Figure 5.6 illustrates an example of calculating the transformation matrix that describes the marker on the forearm segment. Which joint markers were utilized to calculate which segment markers are shown in Table 5.3. Because the markers are always in the same place on the body, this transition is determined in the first moment and remains constant throughout the exercise. In the mentioned Table there are two different matrices, \mathbf{T}_{rj_x} (where 'rj' stands for robot (virtual), joint marker) and \mathbf{T}_{rm_x} (where 'rm' stands for robot (virtual), segment marker). In both matrices, 'x' stands for different body joint or segment. This relation is given by Equation (5.2).

$${}^0\mathbf{T}_x \cdot {}^x\mathbf{T}_y = {}^0\mathbf{T}_y \rightarrow {}^x\mathbf{T}_y = ({}^0\mathbf{T}_x)^{-1} \cdot {}^0\mathbf{T}_y \quad (5.2)$$

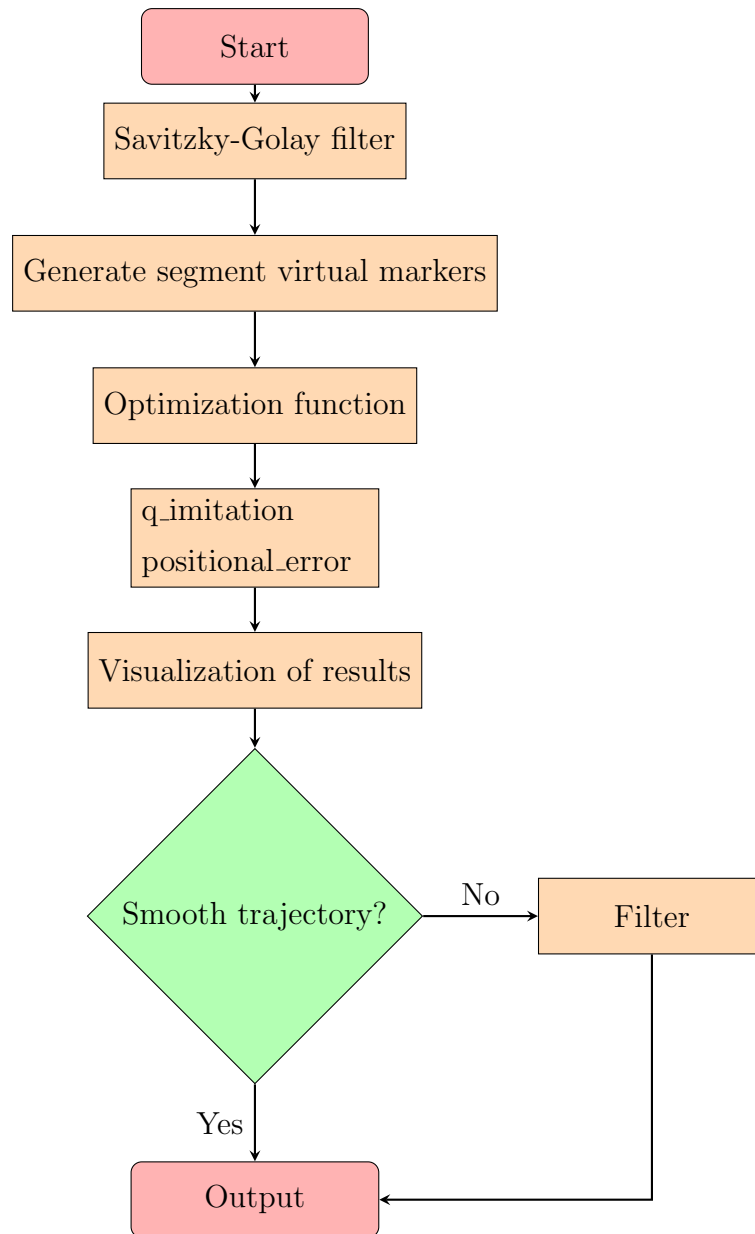


Figure 5.5: Imitation algorithm flowchart

Table 5.3: The pairing of the transformation matrices of joint markers and corresponding segment markers

Right arm	Left arm
$\mathbf{T}^{rjRightShoulder}, \mathbf{T}^{rmRightUpperArm}$	$\mathbf{T}^{rjLeftShoulder}, \mathbf{T}^{rmLeftUpperArm}$
$\mathbf{T}^{rjRightElbow}, \mathbf{T}^{rmRightForearm}$	$\mathbf{T}^{rjLeftElbow}, \mathbf{T}^{rmLeftForearm}$
$\mathbf{T}^{rjRightWrist}, \mathbf{T}^{rmRightHand}$	$\mathbf{T}^{rjLeftWrist}, \mathbf{T}^{rmLeftHand}$

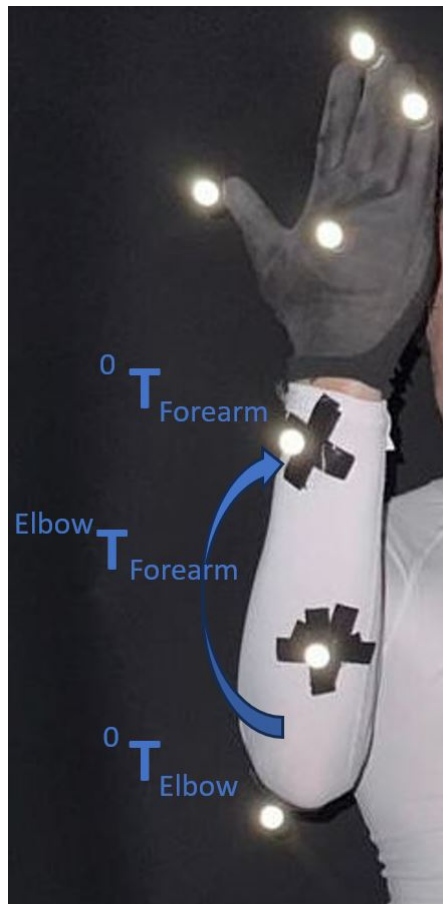


Figure 5.6: Example of calculating the transformation matrix that describes the relation between the marker on the elbow and forearm segment

The optimization algorithm is repeated after determining the placements of the virtual markers of the segments and their relations with the nearest joint markers. The first distinction in the algorithm is that, as previously stated, both segment and joint

markers are observed. Because the skin on the segments moves more than the skin on the joints, and segment markers move related to the muscular action (e.g. flexion), weight factors (α and β) are introduced. They are defined in such a way that markers located on joints are given greater significance in the optimization, in this case, α was set to 1 and β was set to 1,3.

$$q_{imitation}(ti) = \min_q(\zeta) \quad (5.3)$$

$$\zeta = \left\| \begin{bmatrix} \alpha(\overrightarrow{\mathbf{P}}_{am}(ti) - \overrightarrow{\mathbf{P}}_{rm}(q)) \\ \beta(\overrightarrow{\mathbf{P}}_{aj}(ti) - \overrightarrow{\mathbf{P}}_{rj}(q)) \end{bmatrix} \right\|^2 \quad (5.4)$$

The criteria function is shown by the Equations (5.3) and (5.4).

$$\overrightarrow{\mathbf{P}}_{am} = \begin{bmatrix} \mathbf{P}_{amLeftUpperArm} \\ \mathbf{P}_{amRightUpperArm} \\ \mathbf{P}_{amLeftForearm} \\ \mathbf{P}_{amRightForearm} \\ \mathbf{P}_{amLeftHand} \\ \mathbf{P}_{amRightHand} \end{bmatrix} \quad (5.5)$$

$$\overrightarrow{\mathbf{P}}_{rm} = \begin{bmatrix} \mathbf{P}_{rmLeftUpperArm} \\ \mathbf{P}_{rmRightUpperArm} \\ \mathbf{P}_{rmLeftForearm} \\ \mathbf{P}_{rmRightForearm} \\ \mathbf{P}_{rmLeftHand} \\ \mathbf{P}_{rmRightHand} \end{bmatrix} \quad (5.6)$$

$$\overrightarrow{\mathbf{P}}_{aj} = \begin{bmatrix} \mathbf{P}_{ajLeftShoulder} \\ \mathbf{P}_{ajRightShoulder} \\ \mathbf{P}_{ajLeftElbow} \\ \mathbf{P}_{ajRightElbow} \\ \mathbf{P}_{ajLeftWrist} \\ \mathbf{P}_{ajRightWrist} \end{bmatrix} \quad (5.7)$$

$$\vec{\mathbf{P}}_{\text{rj}} = \begin{bmatrix} \mathbf{P}_{rj\text{LeftShoulder}} \\ \mathbf{P}_{rj\text{RightShoulder}} \\ \mathbf{P}_{rj\text{LeftElbow}} \\ \mathbf{P}_{rj\text{RightElbow}} \\ \mathbf{P}_{rj\text{LeftWrist}} \\ \mathbf{P}_{rj\text{RightWrist}} \end{bmatrix} \quad (5.8)$$

$\vec{\mathbf{P}}_{\text{am}}$ (Equation (5.5)) is a vector of actor (real) segment markers, $\vec{\mathbf{P}}_{\text{rm}}$ (Equation (5.6)) is a vector of robot (virtual) segment markers, $\vec{\mathbf{P}}_{\text{aj}}$ (Equation (5.7)) is a vector of real joint markers, and $\vec{\mathbf{P}}_{\text{rj}}$ (Equation (5.8)) (Equation (5.7)) is a vector of virtual joint markers.

The initial value is the vector of generalized coordinates obtained by the initialization process, marked as $\mathbf{q}_{\text{initial}}$. Because all recorded samples are run through a loop in the imitation algorithm, the initial value in the next iteration is equal to the $\mathbf{q}_{\text{initial}}$ from the current iteration. It is also worth noting that for one minute of data acquired at a frequency of 120 Hz, about 7000 different three-dimensional positions for 24 separate markers were gathered. This takes a certain amount of processing power and computer memory, and the method must be run offline.

After the optimization function's loop is completed, the data about the robot's positions in the joint space that characterize a specific movement is collected, marked as $\mathbf{q}_{\text{imitation}}$. Then, as a representation of the coordinate systems and hand trajectory (Figure 5.7), a simulation is run to roughly visually determine if the algorithm effectively mapped the movement from the task space to the joint space.

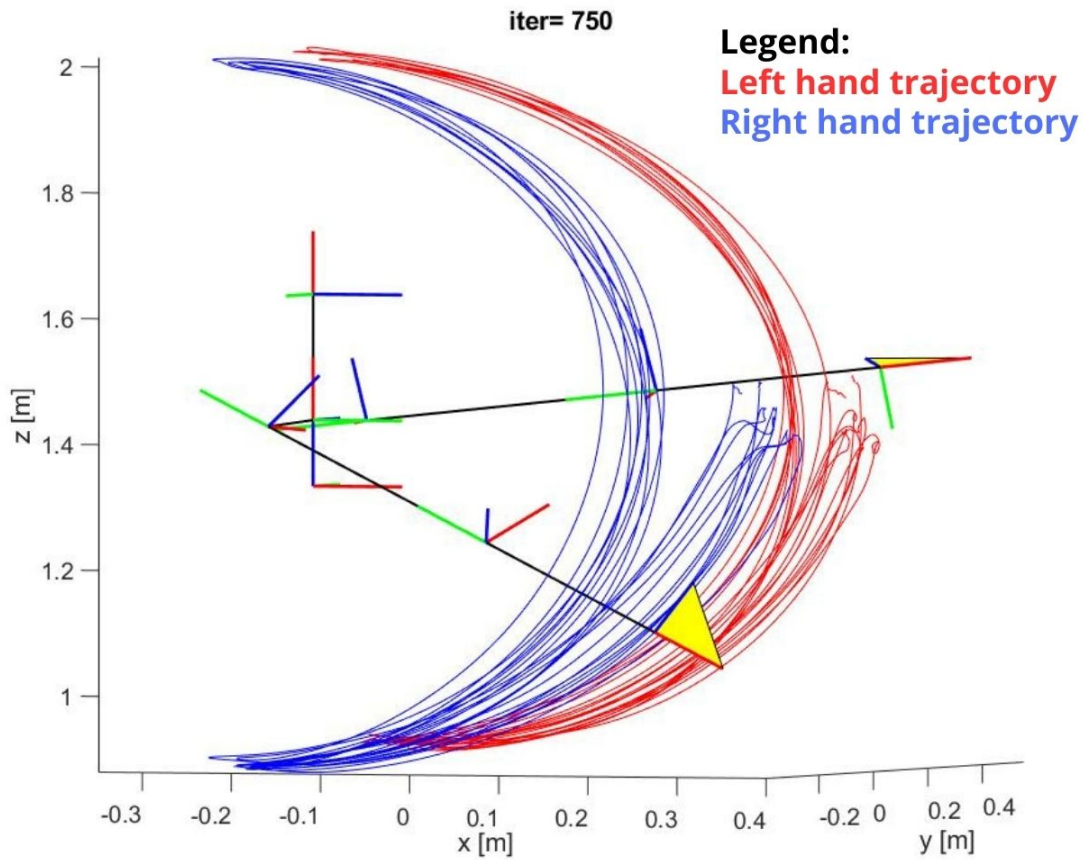
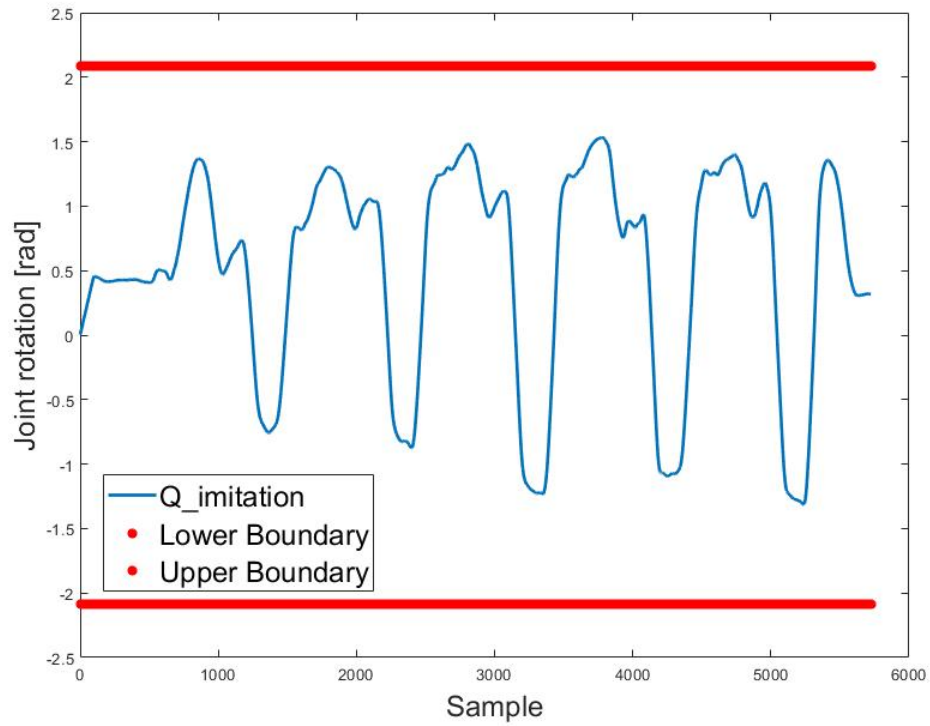
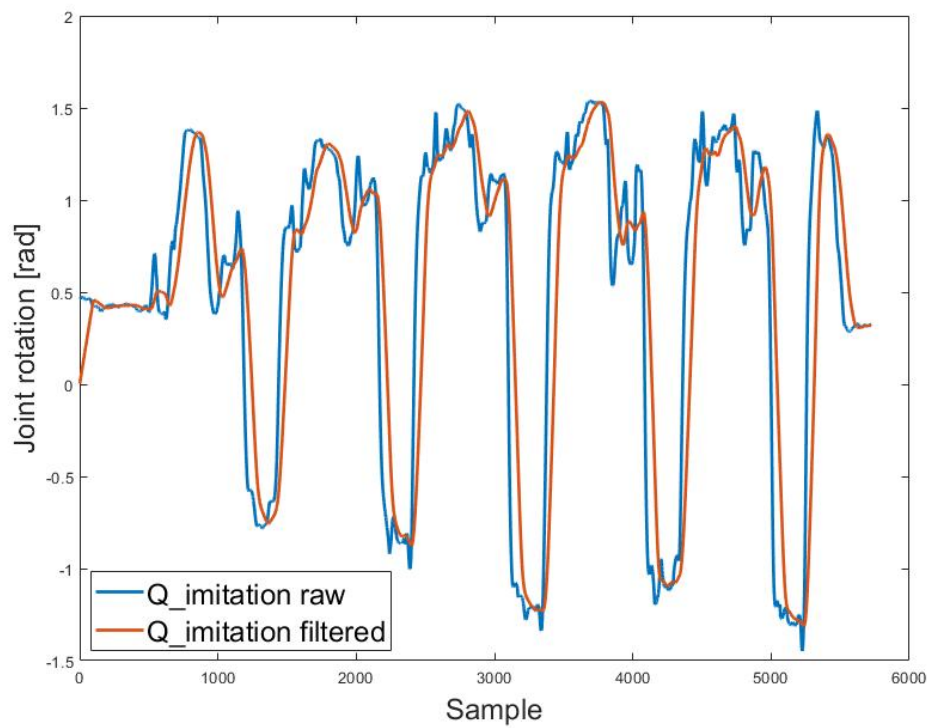


Figure 5.7: 3D data visualization

In addition, the trajectory of each joint, as well as the limits of the robot for that joint, are visualized to ensure that all of the robot's limits in the algorithm are respected (Figure 5.8 a). This visualization provides a closer look at how these trajectories appear. Because the movement was repeated five times in each recording, the trajectories are predicted to be sinusoidal. There should also be no noticeable signal peaks. They must be filtered if they occur. A filter function is applied in this situation to smooth the time series data of signals using a moving average filter specified by the two coefficients. Peaks appear because, when the optimization approach is implemented, each sample is examined independently, and some scores result without errors, while others result with error. In this case, filtering is used to smooth and highlight specific patterns in the data, (Figure 5.8 b). A moving average filter averages the signal, which can aid in reducing high-frequency noise and identifying underlying trends in the data. The final step is to save the data as a '.txt' file.



a)



b)

Figure 5.8: (a) Boundaries and trajectory (b) Raw and filtered trajectory

6 Exercises performed by the robot

This section will outline the methodology employed for executing generated trajectories by the robot. After generating the trajectories for all of the exercises, a Python script was written to allow communication between the computer and the robot. Furthermore, the robot API [13] was used to control the robot's movements.

A form of communication based on the TCP/IP (Transmission Control Protocol/Internet Protocol) set of protocols was used to connect the robot to the computer. These communication protocols are built on the server-client model. In the context of this thesis, the computer from which the data is sent serves as the server, and the robot serves as the client. The IP address and port number of the robot are used to establish a connection, and commands are then sent over this connection.

The Python script first enables communication between the robot and the computer, and then loads the 'naoqi' library. Following that, a loop is initiated that reads data from the '.txt' file, which is the output of the Matlab script and contains the generated trajectory of the robot for recreating the desired movement. Motion and posture classes of the mentioned library were used. Using them, the robot's motors are turned on at the start, and it rises to an equilibrium state before receiving the generated trajectories. After completing the trajectories, the robot returns to the 'rest' position.

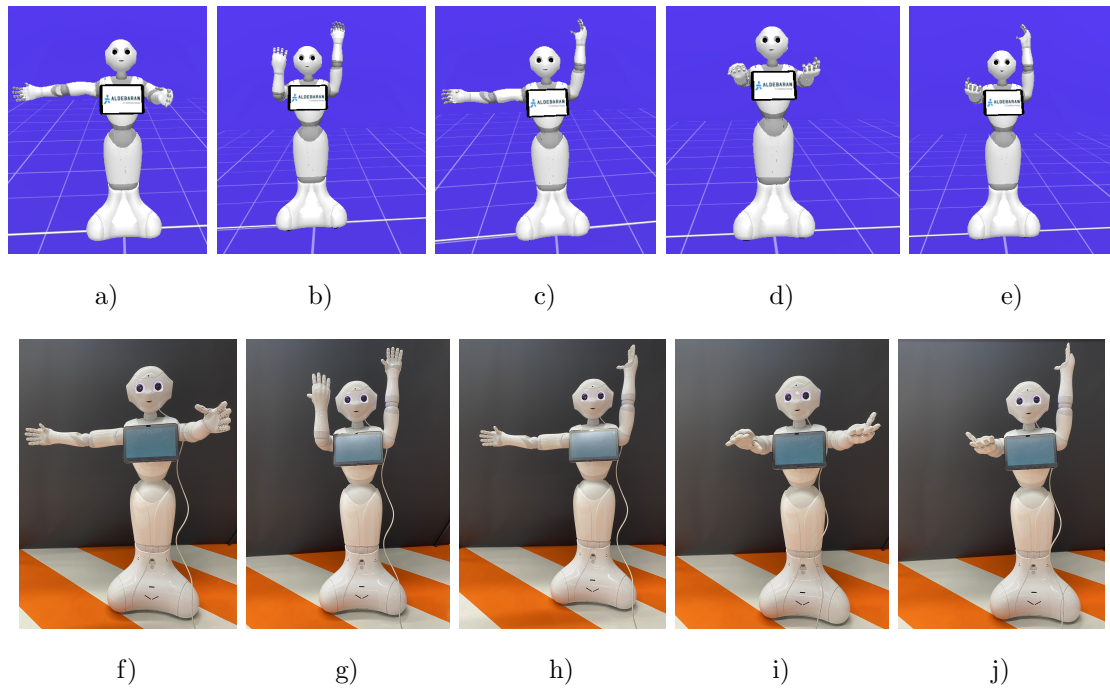


Figure 6.1: Exercises in a simulator (top row) and implemented on the robot (bottom row).

Figure 6.1 depicts the final positions of the movements, both in the simulator and on the robot. The display is analog to the one shown in the image 3.12.

7 Results

This section will showcase the outcomes achieved through the imitation algorithm, which relies on a minimization function and inverse kinematics. The input for this algorithm consists of 3D coordinates of human exercises captured using a motion capture system, while the output consists of the generalized coordinates representing the robot's corresponding movements. It is important to note that only one exercise's results will be visualized, while the results of all five exercises were numerically processed.

The chosen exercise for graphical representation is exercise number 5, as depicted in Figures 3.12 e), 3.12 j), 6.1 e) and 6.1 j). In this exercise, as in the others, recording begins with the reference 'zero position' (Figure 3.6). The sequence involves lowering the arms along the body 7.1 a), followed by raising one arm in the air beside the body while extending the other in front of the body 7.1 b). The arms then return to the position where arms are lowered along the body, repeating the before described movement where the arms interchange roles 7.1 c).

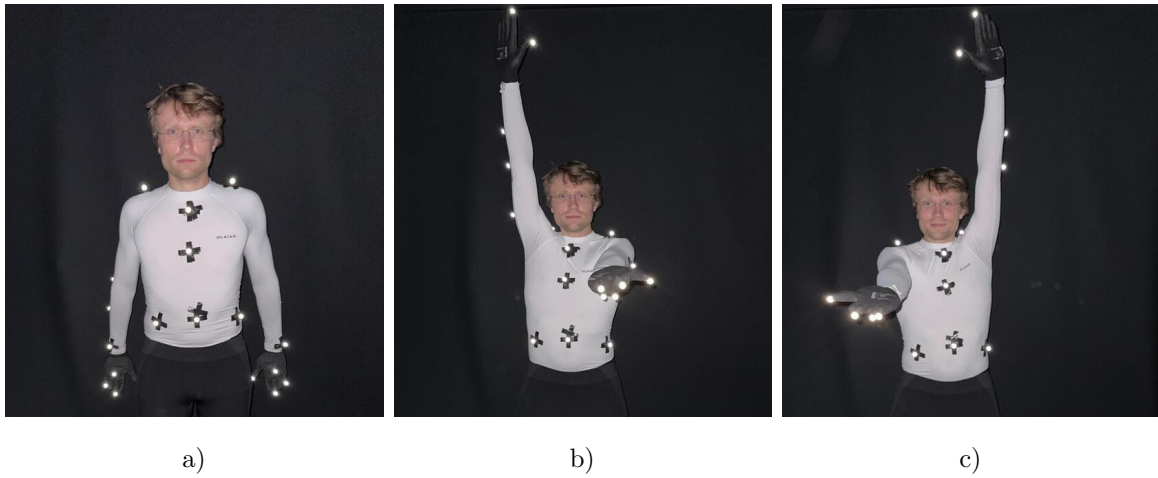


Figure 7.1: Three positions that alternate during the exercise

The kinematics described are evident in the displays illustrating the movements of all five joints of the robot arm. Figure 7.2 presents the results for the left arm, while Figure 7.3 illustrates the right arm. The graphical representation of the angle change is depicted using sinusoids, given that each exercise was repeated five times. Consequently, it is anticipated that the range of motion for the 'Shoulder Pitch' degree of freedom will be between $-1,5709$ and $1,5709$ radians (between -90 and 90 degrees). Considering the palm's orientation change during the exercise, it is also expected that the 'Elbow Yaw' and 'Wrist Yaw' degrees of freedom will have values up to $1,5709$ radians (90 degrees). In contrast, 'Shoulder Roll' and 'Elbow Roll' have a minimal impact on this exercise, resulting in trajectories of smaller amplitude. Insights from previous recordings in a study Knežević et al. [41] suggest that repeating the same exercise enhances performance through iterations, allowing the selection of the best iteration for implementation on the robot. The obtained generalized coordinates presented in Figures 7.2 and 7.3 were first tested on a robot simulator 6.1 e), and then on the real robot 6.1 j).

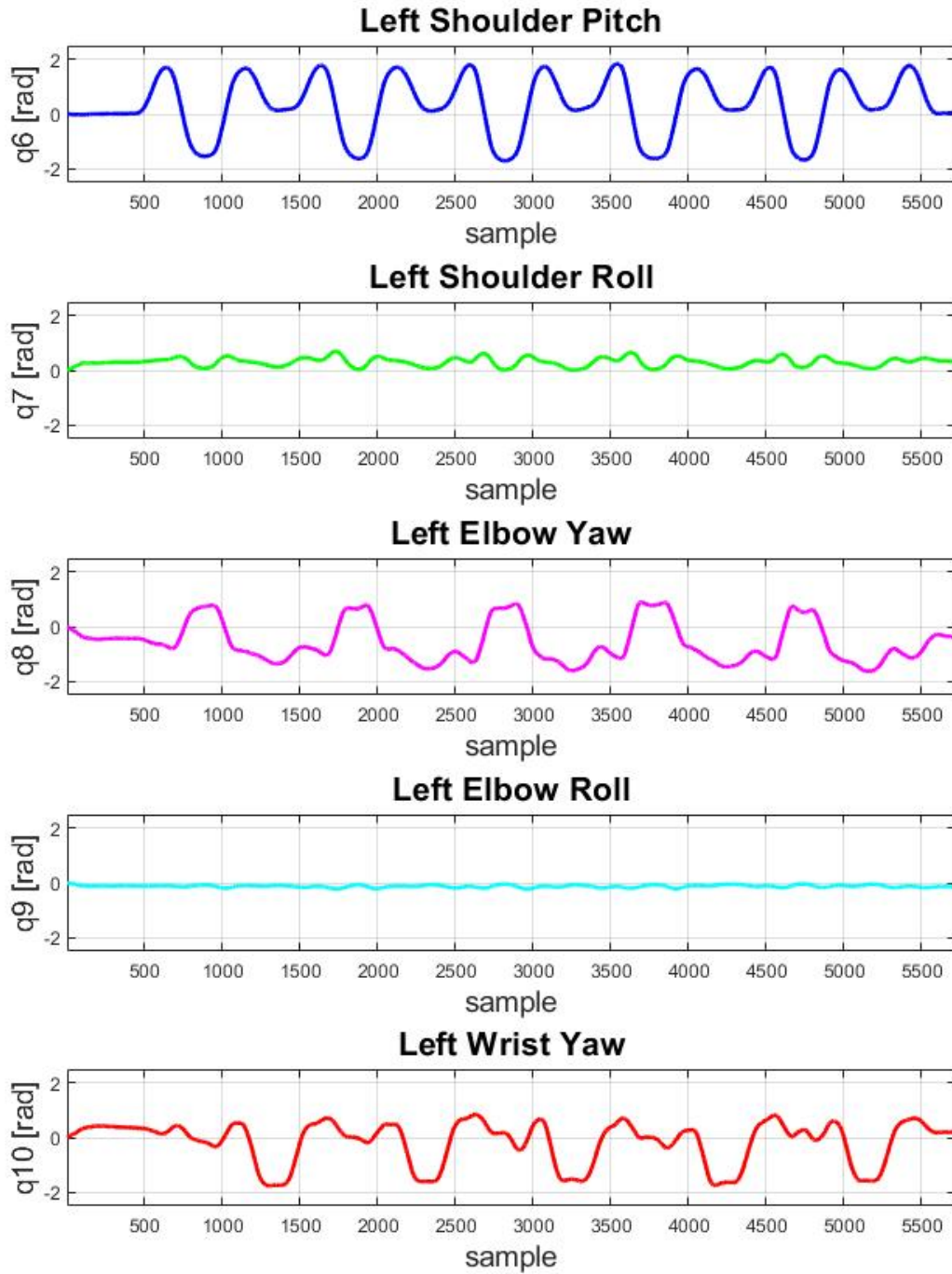


Figure 7.2: Generalized coordinates for the left arm in exercise number 5

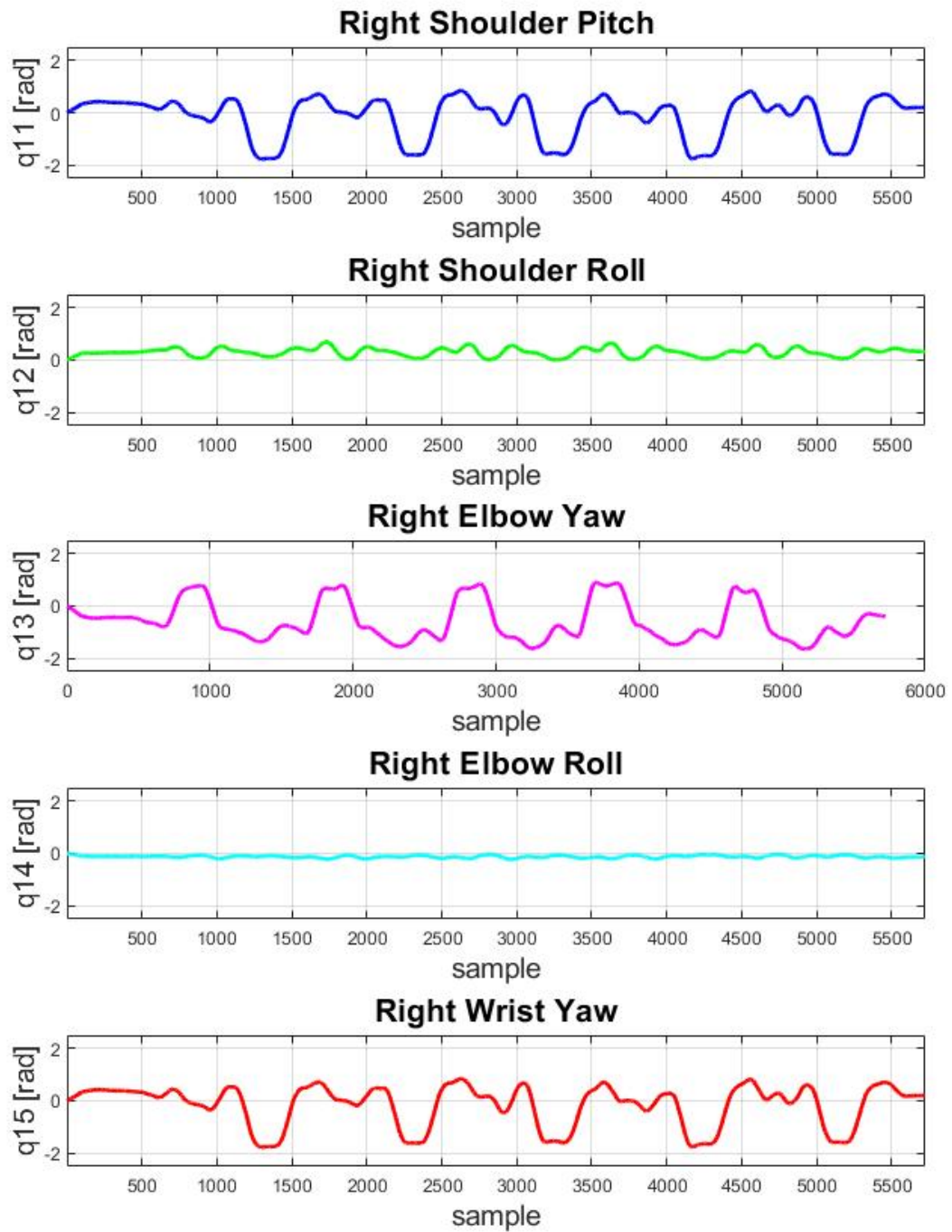


Figure 7.3: Generalized coordinates for the right arm in exercise number 5

In addition to showcasing the generated trajectories, it is important to examine how closely the obtained trajectory aligns with the desired one. Figure 7.4 displays the disparities between real and virtual markers positioned on segments (upper arm, forearm, hand).

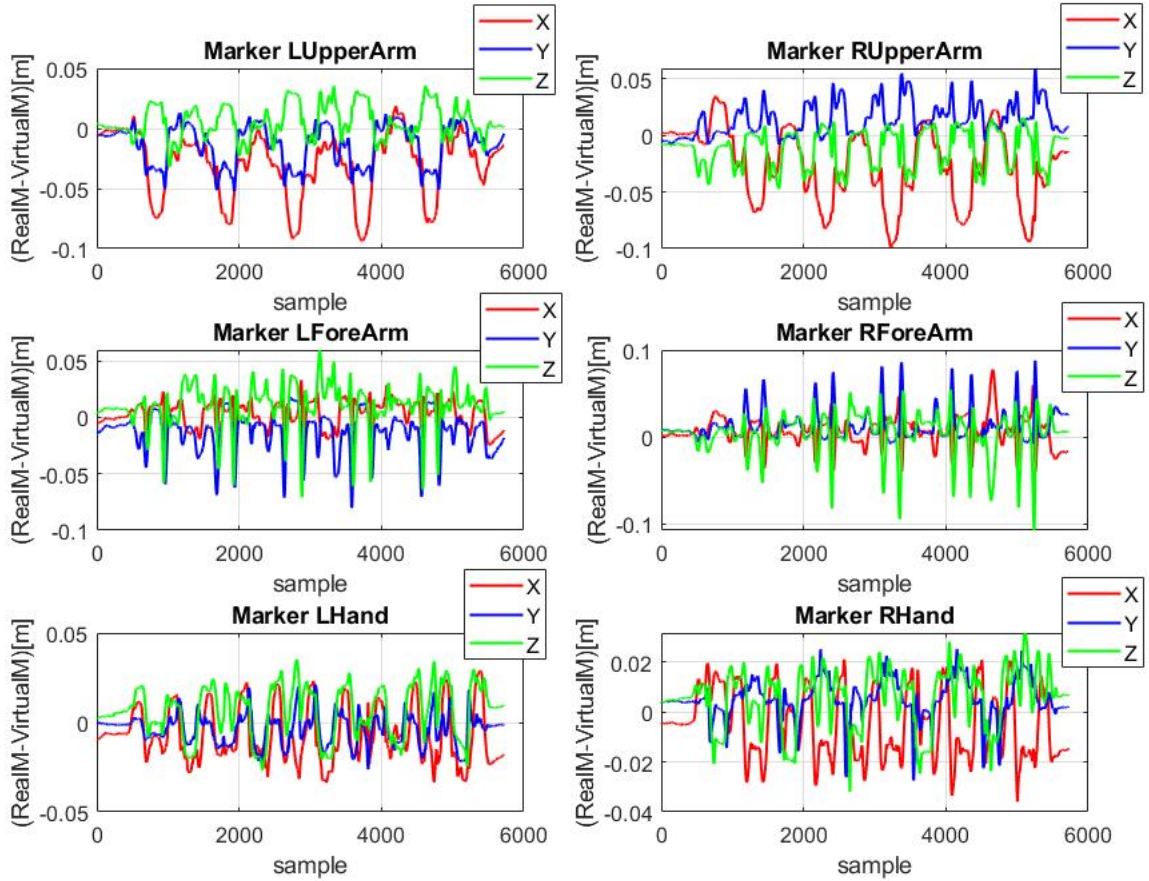


Figure 7.4: Error in following virtual markers with real markers on the upper arm, forearm, and hand during the exercise

Figure 7.5 provides a spatial representation of the desired and achieved trajectories for segment markers.

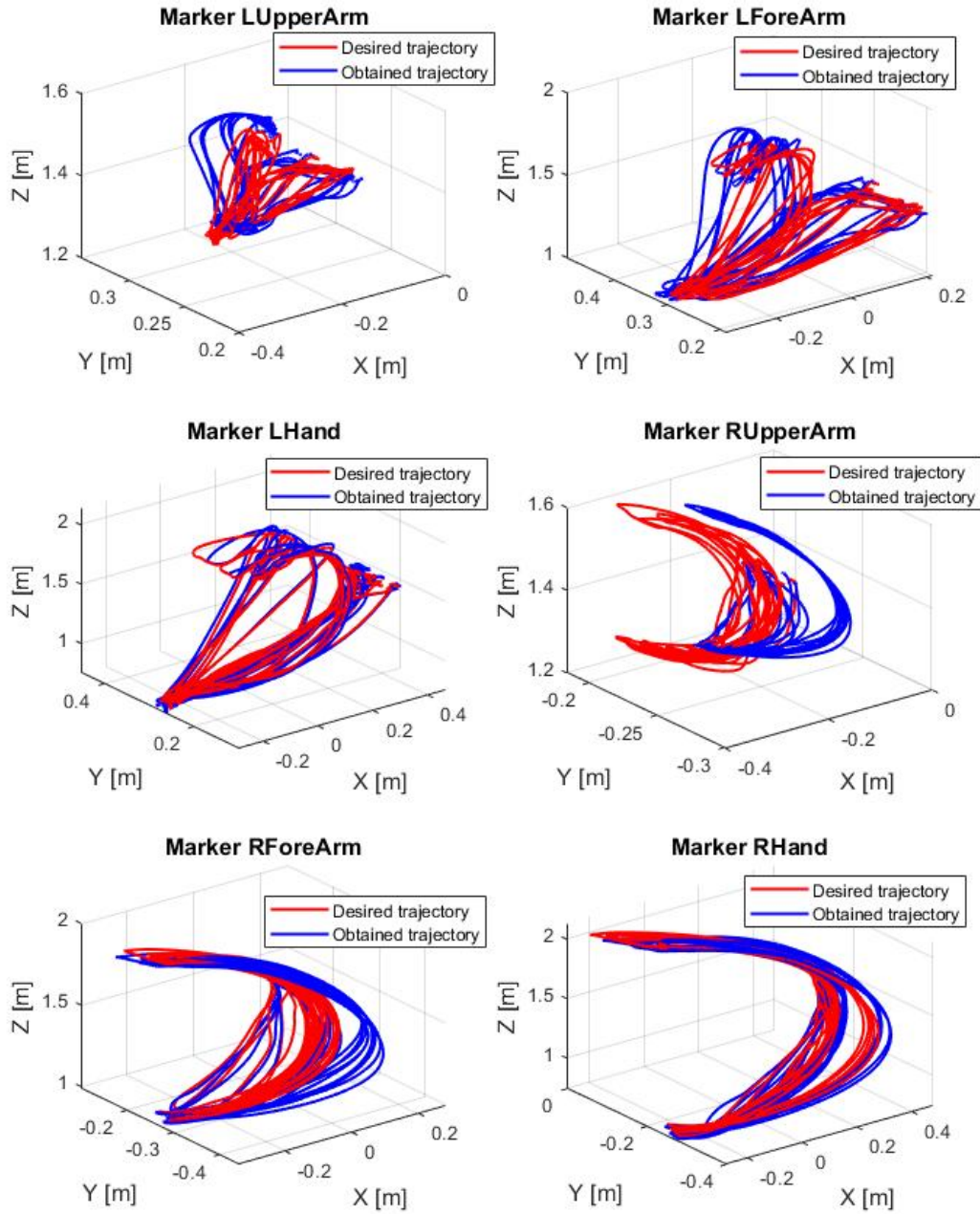


Figure 7.5: Spatial representation of desired and achieved trajectories for segment markers

It's important to highlight that the disparity between the positions of real and virtual markers is more prominent for the shoulder and upper arm markers, gradually diminishing towards the hand and wrist markers. This discrepancy arises from the exclusion of certain aspects of human kinematics in the model. The human kinematic model was devised as an extension of the robot model, representing a simplified version of the human kinematic structure. Human body kinematics are notably more intricate than portrayed, featuring not only three rotations but also three translations in the shoulder. Given that the selected exercises involve expansive ranges of movement, these translations become particularly pronounced. An illustration of shoulder translation along the z-axis is provided in Figure 7.6.

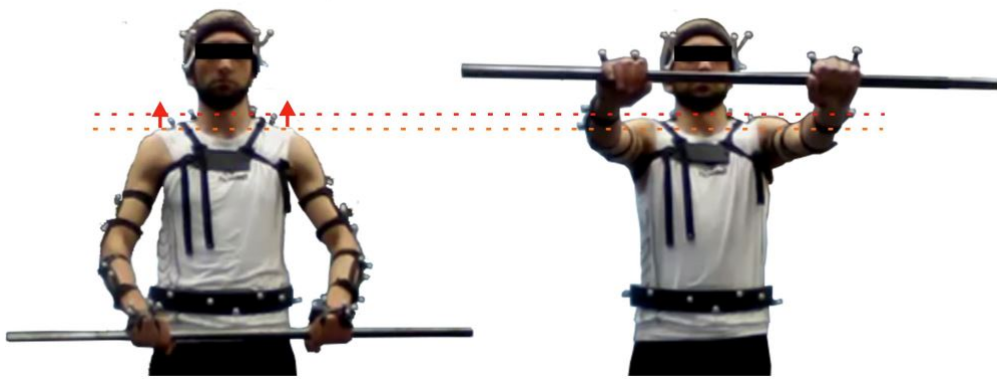


Figure 7.6: Shoulder translation along the z-axis [39]

Figures 7.7, 7.8, and 7.9 depict the differences in positions between recorded joint markers (shoulder, elbow, wrist joint) and the 3D positions of joints obtained by human kinematic model. According to the findings, the errors in following the positions of the shoulder joints are the greatest when compared to the elbow and wrist joints. The 8 cm error in the x and y axes for the shoulder joints is due to unmodeled kinematics in the shoulder. The large error in tracking the upper arm marker (8 cm) also reflects the influence of the unmodeled shoulder in the human kinematic model.

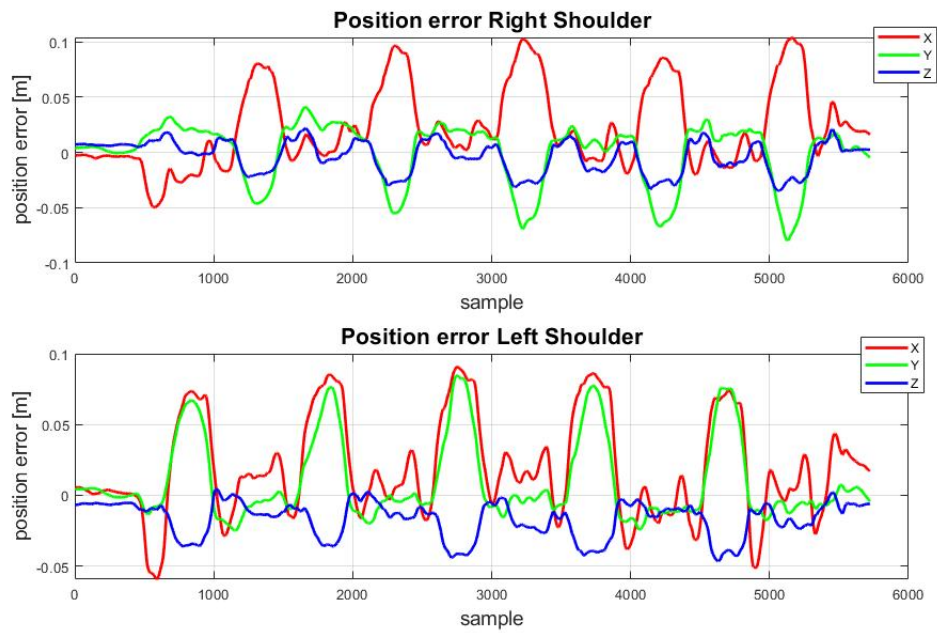


Figure 7.7: Error in following the virtual marker with the real marker of the shoulder during the exercise

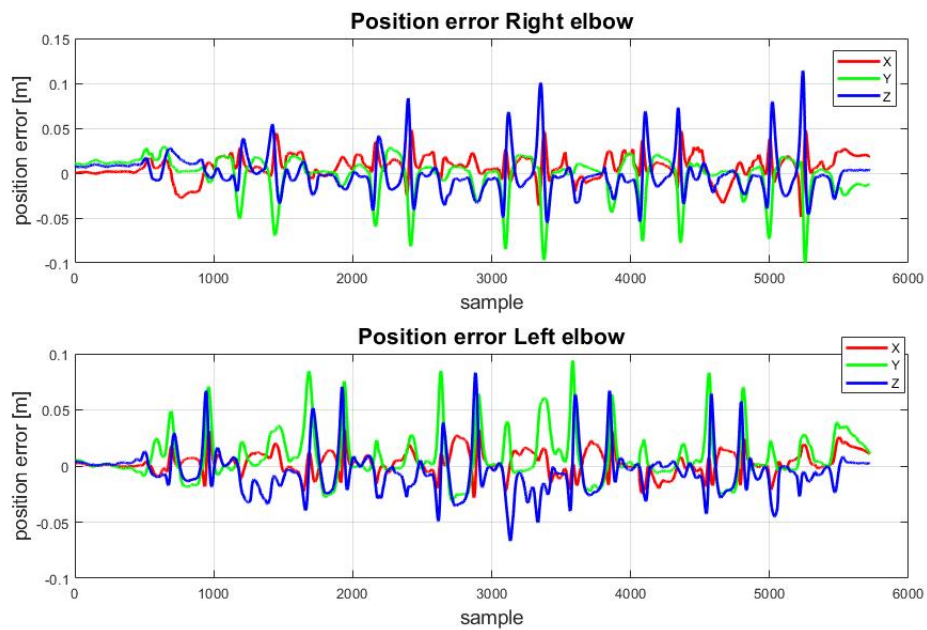


Figure 7.8: Error in following the virtual marker with the real marker of the elbow during the exercise

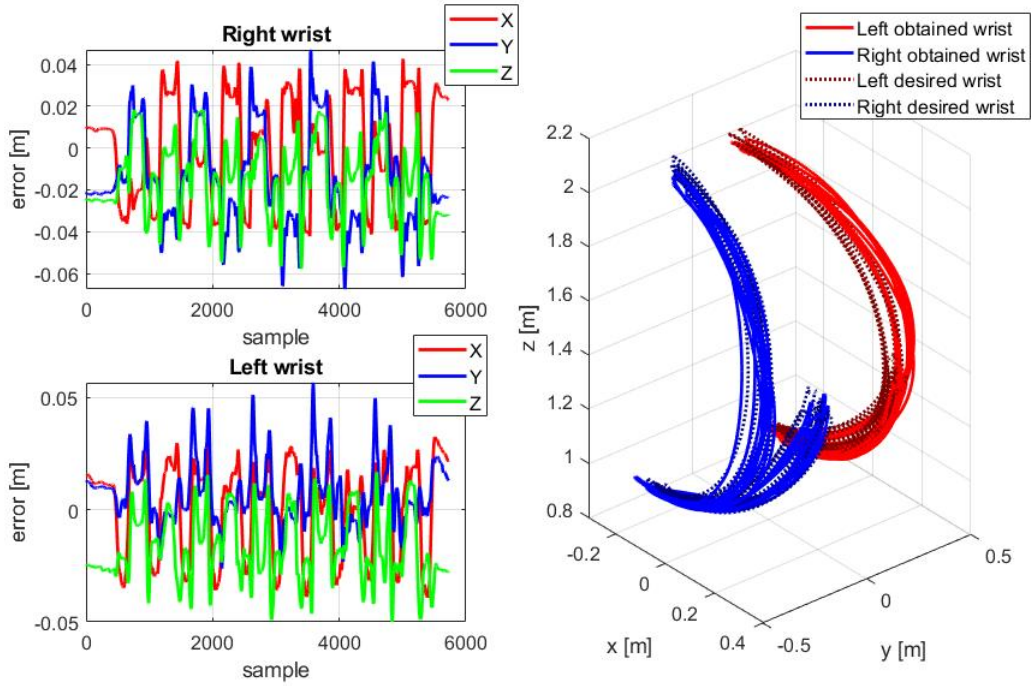


Figure 7.9: Error in following the virtual marker with the real marker of the wrist and spatial representation of desired and achieved trajectories during the exercise

Also, the size of the error in the elbow joints is evident but smaller compared to the error in the shoulder (5cm), which has the effect of defining the fixed length of the segments of the rigid kinematic model of humans, the lengths of which change due to the movement of the joint markers due to the wide range of the arms, which conditions stretching the muscles and skin and moving the marker. A similar conclusion can be drawn by observing the error in tracking the forearm marker, which is about 6cm. On the other hand, the error in following the movement of the joints of the wrist and the hand marker is the smallest, about 2–3 cm. This error is acceptable for contact and non-contact tasks [39]. These results are expected because the kinematic model of the extended robot model below the shoulder is close to the human model. Additionally, Figure 7.9, illustrates errors along the x, y, and z axes and presents the spatial arrangement of the desired and obtained trajectories for the hand joint.

Figure 7.10 illustrates the discrepancies in tracking the virtual marker with the real markers on the fingers (thumb, middle finger, pinky finger) of both hands. The orientation of the palm is an important piece of information, and it is imperative for the

generated robot trajectories to accurately capture this aspect as well. The magnitude of the error is comparable to the hand markers error (3cm), suggesting that the imitation algorithm effectively maps the orientation of the hand.

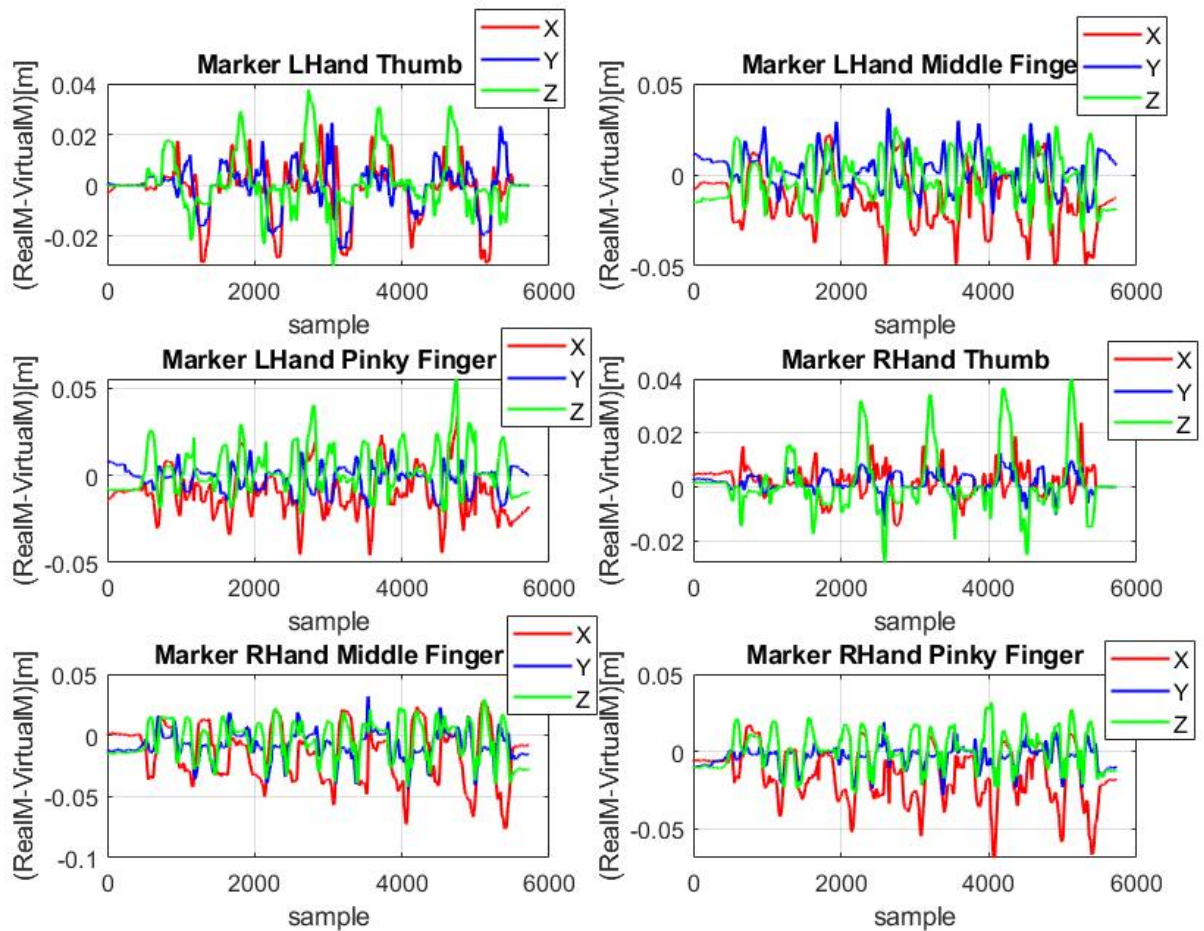


Figure 7.10: Error in following virtual markers with real markers on the thumb, middle finger and pinky finger during the exercise

8 Conclusion

The mapping of human movements to robot motions is presented in the context of the master's thesis. Human movements were recorded using the OptiTrack motion capture system to accomplish this. Following that, a kinematic model of the humanoid robot Pepper was created, as well as a kinematic model of a human (with two extra degrees of freedom) that was integrated into the robot's kinematic model. An imitation algorithm was then used to generate motion described with generalized coordinates. The imitation algorithm is formed by the minimization of positions between real markers (recorded data) and virtual markers (located on the extended kinematic model of the robot) and the utilization of inverse kinematics. Inverse kinematics involves calculating the joint rotation in radians for specific joint positions in meters. The generated trajectories were then implemented on the humanoid robot Pepper.

The analysis of results revealed that error in marker positions is a direct result of the unmodeled translational degrees of freedom in the human kinematic model, defined as the robot's extended kinematic model. The exercises chosen involve a wide range of shoulder movements, making the differences more noticeable when compared to mapping movements focused on elbow or hand actions. The simplified kinematic model of the human, which excludes shoulder translations, is justified because the robot used for the exercises lacks these degrees of freedom and cannot execute the mentioned translations. The rotational degrees of freedom have been successfully mapped from humans to robots, enabling the mapping of selected exercises to improve the gross motor skills of children. The unmodeled translational degrees of freedom resulted in errors displayed in tracking recorded data, as presented and explained in the 'Results' chapter. Regardless of that,

due to human physiology, translations will occur during the performance of exercises by children.

This master's thesis builds upon the preliminary research conducted by Knežević et al. [41], carried out in collaboration with the Mihajlo Pupin Institute, Belgrade, and the Faculty of Technical Sciences, University of Novi Sad. The thesis continues the work initiated in that research, focusing on mapping movements from robots to humans. The mapped motions, suggested by experts in the field of kinesiology, aim to enhance the gross motor skills of children aged 7 to 9. Future efforts will be directed toward developing human pose estimation algorithms and a validation algorithm to assess how well children follow movements demonstrated by the robot. Additionally, the progress of children's results over a specific period will be monitored. To conclude, future efforts will be focused on motivating children to exercise and thus improving their gross motor skills through the incorporation of innovative technologies and algorithms. These include the utilization of humanoid robots and computer vision to create engaging and effective approaches for promoting physical activity among children.

Bibliography

- [1] T. Kanda, T. Miyashita, T. Osada, Y. Haikawa, and H. Ishiguro. Analysis of humanoid appearances in human–robot interaction. *IEEE transactions on robotics*, 24(3):725–735, 2008.
- [2] <https://optitrack.com/software/motive/>, August 2023.
- [3] I. Zamalloa, R. Kojcev, A. Hernández, I. Muguruza, L. Usategui, A. Bilbao, and V. Mayoral. Dissecting robotics-historical overview and future perspectives. *arXiv preprint arXiv:1704.08617*, 2017.
- [4] Communication Team. Types of robots: Classification, applications and examples. <https://www.telefonica.com/en/communication-room/blog/types-of-robots-classification-applications-and-examples/>, May 2023.
- [5] Fanuc products. <https://www.fanucamerica.com/product/robots/productsbyseries/default.aspx?seriesId=3&robotseries=LR%20Mate%20Series>.
- [6] Mobile robotics applications: more safety and productivity for your plant. <https://robotnik.eu/mobile-robotics-applications-more-safety-and-productivity-for-your-plant/>.
- [7] Nao6. <https://www.aldebaran.com/en/nao>.
- [8] Robotic drones can now fly, stop and perch just like birds. <https://science.howstuffworks.com/perchiing-drones-news.htm>.

- [9] J. Dzedzickis, A. Subaciute-Žemaitiene, E. Šutinys, U. Samukaite-Bubniene, and V. Bučinskas. Advanced applications of industrial robotics: New trends and possibilities. *Applied Sciences*, 12(1):135, 2021.
- [10] M. Zare, P. M. Kebria, A. Khosravi, and S. Nahavandi. A survey of imitation learning: Algorithms, recent developments, and challenges. *arXiv preprint arXiv:2309.02473*, 2023.
- [11] G. Maeda, M. Ewerton, D. Koert, and J. Peters. Acquiring and generalizing the embodiment mapping from human observations to robot skills. *IEEE Robotics and Automation Letters*, 1(2):784–791, 2016.
- [12] Pepper- technical specifications. <https://support.aldebaran.com/support/solutions/articles/80000958735-pepper-technical-specifications>.
- [13] Naoqi apis. <http://doc.aldebaran.com/2-4/naoqi/index.html>.
- [14] H. Mahdi, S. A. Akgun, S. Saleh, and K. Dautenhahn. A survey on the design and evolution of social robots—past, present and future. *Robotics and Autonomous Systems*, page 104193, 2022.
- [15] Pepper robot - discover its technical specifications. <https://www.generationrobots.com/pepper/technical-specifications.html?lang=en>.
- [16] A. Alam. Social robots in education for long-term human-robot interaction: socially supportive behaviour of robotic tutor for creating robo-tangible learning environment in a guided discovery learning interaction. *ECS Transactions*, 107(1):12389, 2022.
- [17] R. Haratian. Motion capture sensing technologies and techniques: A sensor agnostic approach to address wearability challenges. *Sensing and Imaging*, 23(1), 2022.
- [18] X. Bin Peng, T. Coumans, E. Zhang, T.-W. Lee, J. Tan, and S. Levine. Learning agile robotic locomotion skills by imitating animals. *Robotics: Science and Systems XVI*, 2020.
- [19] M. F. R. Lee and K. H. E. Lee. Autonomous target tracking and following mobile robot. *Journal of the Chinese Institute of Engineers*, 36(4):502–529, 2013.

- [20] Ma. Pavliv, F. Schiano, C. Reardon, D. Floreano, and G. Loianno. Tracking and relative localization of drone swarms with a vision-based headset. *IEEE Robotics and Automation Letters*, 6(2):1455–1462, 2021.
- [21] O. Tsilomitrou, K. Gkountas, N. Evangeliou, and E. Dermatas. Wireless motion capture system for upper limb rehabilitation. *Applied System Innovation*, 4(1):14, 2021.
- [22] S. Barris and C. Button. A review of vision-based motion analysis in sport. *Sports Medicine*, 38(12):1025–1043, 2008.
- [23] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. Black. Amass: Archive of motion capture as surface shapes. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.
- [24] A. Yang, Y. Chen, W. Naeem, M. Fei, and L. Chen. Humanoid motion planning of robotic arm based on human arm action feature and reinforcement learning. *Mechatronics*, 78:102630, 2021.
- [25] J. Lobo-Prat, J. M. Font-Llagunes, C. Gómez-Pérez, J. Medina-Casanovas, and R. M. Angulo-Barroso. New biomechanical model for clinical evaluation of the upper extremity motion in subjects with neurological disorders: An application case. *Computer Methods in Biomechanics and Biomedical Engineering*, 17(10):1144–1156, 2012.
- [26] M. Koppenborg, P. Nickel, B. Naber, A. Lungfiel, and M. Huelke. Effects of movement speed and predictability in human–robot collaboration. *Human Factors and Ergonomics in Manufacturing and Service Industries*, 27(4):197–209, 2017.
- [27] <https://www.movella.com/products/wearables/xsens-mtw-awinda>, August 2023.
- [28] E. Van der Kruk and M. M. Reijne. Accuracy of human motion capture systems for sport applications; state-of-the-art review. *European journal of sport science*, 18(6):806–819, 2018.
- [29] <https://optitrack.com/>, August 2023.

- [30] <https://optitrack.com/cameras/prime-13/>, August 2023.
- [31] <https://docs.optitrack.com/motive/data-editing>, August 2023.
- [32] A. Savitzky and M. J. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, 1964.
- [33] <https://support.aldebaran.com/support/solutions/articles/80000958735-pepper-technical-specifications>, August 2023.
- [34] W. Khalil and E. Dombre. *Modeling, Identification amp; Control of Robots*. Kogan Page Science, 2004.
- [35] J. Denavit and R. S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *Journal of Applied Mechanics*, 22(2):215–221, 1955.
- [36] S. Hayati. Robot arm geometric link parameter estimation. *The 22nd IEEE Conference on Decision and Control*, 1983.
- [37] M. Granja, N. Chang, V. Granja, M. Duque, and F. Llulluna. Comparison between standard and modified denavit-hartenberg methods in robotics modelling. *World Congress on Mechanical, Chemical, and Material Engineering*, 2016.
- [38] W. Khalil and D. Creusot. Symoro+: A system for the symbolic modelling of robots. *Robotica*, 15(2):153–161, 1997.
- [39] M. Tomić. *DUAL-ARM ROBOTIC MANIPULATION INSPIRED BY HUMAN SKILLS*. Phd thesis, University of Belgrade, 2018.
- [40] Aldebaran 2.0.6.8 documentation. http://doc.aldebaran.com/20/family/juliette_technical/joints_juliette.html.
- [41] T. Knežević, M. Radmilović, J. Borojević, J. Šumarac, M. Švaco, and M. Raković. Physical education exercises validation through child-humanoid robot interaction. *Advances in Service and Industrial Robotics*, page 132–140, 2023.