



UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA
CORSO DI LAUREA MAGISTRALE IN BIOMEDICAL ENGINEERING

Kinematic models for human robot collaboration

Student:
Giorgia Chiriatti

Advisor:
Dr. Giacomo Palmieri

A.A 2018/2019



UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA
CORSO DI LAUREA MAGISTRALE IN BIOMEDICAL ENGINEERING

Kinematic models for human robot collaboration

Student:
Giorgia Chiriatti

Advisor:
Dr. Giacomo Palmieri

A.A 2018/2019

UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA

Via Brezze Bianche – 60131 Ancona (AN), Italy

Acknowledgments

First of all, I want to thank Mr. Palmieri, that with his perseverance has followed constantly this thesis project, showing me what beautiful is the research field. I would also to thank all the Meccanica delle Macchine team, which makes my workdays more enjoyable and pleasant.

I'd like to thank my parents that support me every day and my brother, that, even though we are far, he is beside me in my own path in life.

I want to thank my old friends, Benedetta, Carak, Federico, Francesco, Giacomo and Lorenzo that even are undergone to my little "life anxiety", they are always ready to help me. Thank you guys for the funny days that we spent together. Thank you Roby for these years that we have passed, sharing with me anxiety but above all amazing days.

And last but not least, Luigi, my rock, always on my side. Thank you for comforting me every single day and giving to me always your energy. You already know that without you it wouldn't be the same. This project is also yours

,

Giorgia Chiriatti

Abstract

The collaborative robots, also known as cobots, are nowadays involved in several industrial processes to speed up and streamline the process for greater productivity. The cobots and the operators are implicated in the cooperation of activities combining robotic and human skills: the robot's precision and repeatability is synchronous with the human intelligence and flexibility. Since these robotic systems are safe and reliable, they are adopted in the medical field, especially in the rehabilitation sector. Some traditional rehabilitation devices already exist on the market (MOTORE, PABLO), but the cobots application with handle systems have grown only over the last few years, as in the project of Universal RoboTrainer or ROBERT Life Science. Unlike traditional robots, the new generation of cobots is designed to be flexible, lightweight and easy to program. They typically have a rounded shape without any pinch points and internalized wires. Standards for collaborative operations are needed to ensure human safety and the proper machine operation.

In this thesis, mathematical models of human and robotic arm will be studied for the kinematic analysis of Human-Robot Collaboration (HRC). First, the motion planning is accomplished to generate a desired motion through a sequence of points. Then, the computation of the direct and inverse kinematics for a human and a robotic arm has been performed. In particular, the model chosen for the upper limb is the one with 7 degrees of freedom (DoF) represented by three rigid segments connected by frictionless joints. The robotic arm taken into account, instead, is the UR5 by Universal Robot company, which is characterized by 6 DoF. The aim of this project is to realize a framework for the human-robot motion planning. Starting from the anthropometric measurements and the joint excursions of the upper limb, the kinematic analysis allows the execution of robot joint motion perfectly in line with the human one. Therefore, the proposed algorithm is designed to be suitable for each patient, and in the meantime, the physiotherapist is responsible only for the assessment of the upper limb motion.

Simulations for HRC in rehabilitation processes will be executed with the aim of realizing a virtual prototype model for human and robotic arm. OpenSim software is adopted to model only the human system based on anthropometric measures. ADAMS[®] software, instead, is exploited for combining motions of human and robot models. Then, a control simulation has been performed in Simulink-ADAMS[®] environment.

Abstract

Three different human-robot motions are analyzed: one-way motion, round-trip motion and 3-points motion. In conclusion, the virtual model can execute the predefined movements without computational errors. The proposed methodology is a starting point of cobots integration into current rehabilitation practice improving the quality of physical rehabilitation. Future research could involve the analysis of different human-robot working modality.

Contents

Abstract	xi
1. Introduction	1
1.1. Robotic development	1
1.2. Collaborative robots for rehabilitation process	4
1.3. Robot structure	8
1.4. Standards for collaborative operations	11
2. Collaborative robots	17
2.1. Universal Robot UR5	19
2.2. KUKA LBR iiwa	25
3. Kinematic models	29
3.1. Motion Planning	29
3.2. Direct and Inverse Kinematic of human arm	34
3.3. Direct and Inverse Kinematic of robotic arm	45
4. Simulations	55
4.1. OpenSim software	55
4.2. Adams models	57
4.3. Simulink-Adams control simulation	60
4.4. Results	66
5. Conclusions	79
A. Denavit-Hartenberg convention	81
B. Matlab code	85
Bibliography	101

List of Figures

1.1. Human-robot cooperation.	2
1.2. MOTORE used for upper limb rehabilitation.	5
1.3. PABLO devices used in rehabilitation process.	5
1.4. Possible human-robot interfaces.	6
1.5. ROBERT Life Science.	7
1.6. Universal RoboTrainer.	7
1.7. The MitsubishiPA10-7 robot platform.	7
1.8. Example of the links of a robot.	9
1.9. Example of the joints of a robot.	9
1.10. Example of some anthropomorphic robots.	10
1.11. Example of anthropomorphic robot in cooperation activity.	11
1.12. Basic procedure of risk assessment.	13
1.13. Four basic safety procedures of ISO/TS 15066:2016.	14
2.1. Cobots: Fanuc CR-35iA, ABB YuMi, UR5, KUKA LBR iiwa [1].	18
2.2. UR Company production.	21
2.3. UR5 and teach pendant with Polyscope Robot User Interface.	23
2.4. Examples of grippers and vision system suitable for UR5. From the left: AG-95L gripper, ROBOTIQ AIRPICK gripper and 3D-TOF sensors	24
2.5. KUKA KMR iiwa	25
2.6. KUKA LBR iiwa: (1) In-line wrist, (2) Joint module, (3) Base frame, A1-A7 motors	27
2.7. Electrical and mechanical components of LBR iiwa.	28
3.1. Timing law with trapezoidal velocity profile.	31
3.2. Timing law of polynomial function of the 5th order.	33
3.3. Representation of upper limb joints.	35
3.4. Human arm axis convention	36
3.5. Example of human arm configuration.	41
3.6. Example of joint position configuration following a fixed trajectory.	46

List of Figures

3.7. Difference between no-weighted and weighted joints.	47
3.8. Revolute joints of eUR5.	47
3.9. Kinematics of UR5.	48
3.10. UR5 dimensions.	48
3.11. DH parameters in the robot vertical position.	49
3.12. Geometry scheme for calculating q_1	50
3.13. Geometry scheme for calculating q_5	51
3.14. Geometry for q_6	52
3.15. Geometry for q_4	52
3.16. Example of the initial and final configuration of UR5.	53
4.1. Example of one right arm motion.	56
4.2. CAD systems imported in ADAMS [®]	58
4.3. Fixed reference frames in ADAMS [®] environment.	59
4.4. Initial pose of the system.	59
4.5. Co-simulation of mechatronic system.	60
4.6. Simulink-ADAMS control scheme.	61
4.7. Simulink-ADAMS control scheme with a round-trip polynomial motion. . .	64
4.8. Simulink-ADAMS control scheme with a polynomial motion through three fixed points.	65
4.9. Example of a one-way motion.	66
4.10. Example of a round-trip motion.	68
4.11. Example of a 3-points motion.	69
4.12. Human joint position during the one-way motion.	71
4.13. Robot joint position during the one-way motion.	72
4.14. Human joint position during the round-trip motion.	73
4.15. Robot joint position during the round trip motion.	74
4.16. Human joint position during the 3-points motion.	75
4.17. Robot joint position during the 3-points motion.	76
4.18. Plot of Scope block related to the Adams' output sub-system.	77
A.1. Denavit-Hartenberg kinematic parameters.	82

List of Tables

1.1. EU Directives [2].	15
1.2. General Standards [2].	15
1.3. Robot Standards [2].	16
2.1. Technical specifications [3].	22
2.2. Basic limiting safety-related functions in UR5 [3].	24
2.3. LBR iiwa: Motion range and Speed with rated payload for each axis.	28
3.1. Anthropometric values for human arm [4].	37
3.2. Admissible range for human arm joints [5].	37
3.3. DH parameters of UR5	48
4.1. Initial and final configuration of human joints.	56
4.2. Initial and final joint values for each motion.	67

Chapter 1.

Introduction

To bring the industrial production line towards high performance, a new concept of robotic is developing: the collaborative robots. These advanced robots are adopted not only in industrial applications, but also in the medical field. In particular, the reliability and security of cobots have applications in neuromuscular rehabilitation. In fact, due to the increasing scientific and technology potential and the rapid ageing of the society, many research groups have proposed robots to make the rehabilitation process easier. Consequently, this thesis project is focused on the development of a framework for human-robot motion planning, based on specifications adaptable for each different patient. In particular, from the anthropometric measurements and the joint excursions of the upper limb, is performed the analysis and the planning of the motion to realize kinematic simulations. From the kinematic study, is possible to extract the robot joint motions suitable for the specific analysis. In this way, the physiotherapist is the only one responsible for the assessment of the upper limb motion and the goal of the proposed algorithm is to design the robot joint trajectory perfectly in line with each human movement.

1.1. Robotic development

The concept of industrial robotics involves robot design, control and application in industry. The operation of a robot in industrial employments is generally included in a structured environment, whose geometrical or physical characteristics are mostly known a priori. In the 1960s were developed the first industrial robots characterized by versatility, adaptability to a priori unknown situations, positioning accuracy and execution adaptability. In the meantime, the automation concept, which joints the operation execution with intelligent information processing, becomes more and more spread. Automation is then the synthesis of industrial technologies typical of the manufacturing process and computer technology. To realize a flexible automated manufacturing system is essential to have a strong interaction of computer technology with the industrial technology. Indus-

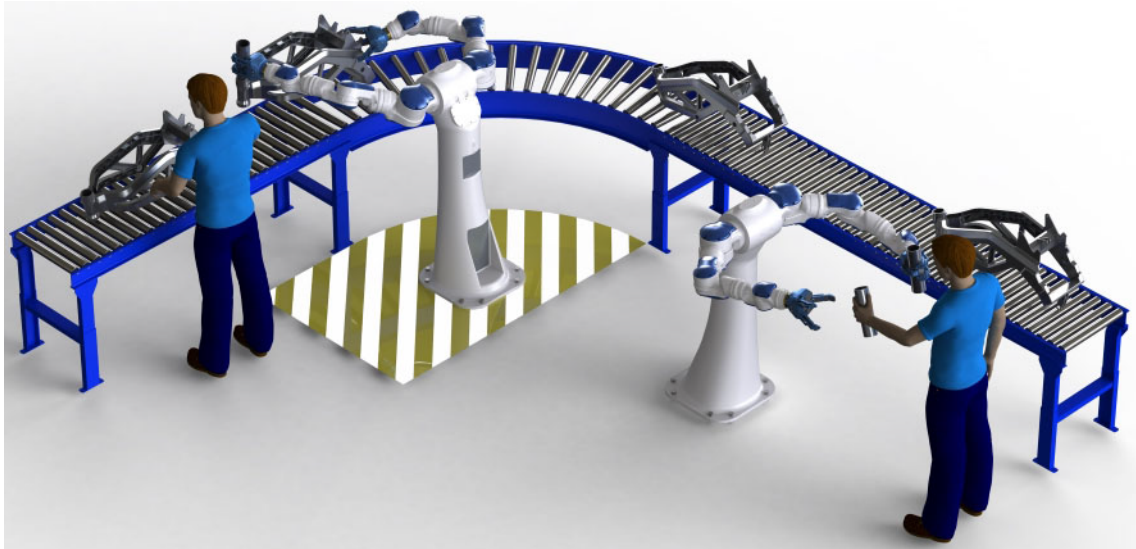


Figure 1.1.: Human-robot cooperation.

trial robots, which are one of the key drivers in Industry 4.0, are typical components of automated system that have evolved considerably since the last decades of the 20th century [6]. They are becoming more productive, flexible, versatile, safer and collaborative bringing the industrial automation towards a radical change in the production processes. It means that the machines have self-optimization, self-configuration and even artificial intelligence to complete tasks to achieve superior cost efficiencies and better quality services. The common strategy among industries has been to automate high volume products with low product variant. Instead of having big robot cells that are static, the trend is towards collaborative robots, small and flexible units. In this way, the industrial landscape is transformed, bringing out a new concept of robotics focused on cooperation activities with humans. The collaborative robots, also known as cobots, are expected to share the same environment with humans and to collaborate with them without requiring safety fences (1.1). The robot and the operator are involved in the cooperation of activities to improve the process productivity because of the combination of specific human and robotic skills. Human-robot collaboration lets several levels of automation (fully or partially) and, consequently, an opportune human intervention. The co-existence of humans and robots seems a promising solution for the execution of the tasks: robot's precision and repeatability is synchronous with human intelligence and flexibility. Robots and humans work hand by hand on interlinking tasks with the opportunity to speak using the dialog-controlled user interface. These new robots can be included in various operations, as logistics, production

1.1. Robotic development

and office management. They can be also controlled remotely by the operator that will be always informed through a message in presence of machine problems. The first cobots were designed and realized by the General Motors company. They were totally safe for humans because the driving force was not present in the machine. In 2004 KUKA company launched to the market LBR 3, the first lightweight cobot controlled by computer and designed for safe close cooperation between human and robot on deeply sensitive tasks. From then on, the companies have started to invest in the advanced technology of robots. The cobots are designed for different applications but unique safety standards must be considered for the collaborative operations (Section 1.4). In fact, every robotic company need to take measures to ensure human safety and the proper machine operation. First of all, severe restrictions are applied to the robot motion as speed reduction and bounds on driving forces. Indeed, if the distance between robot and human become smaller, the velocity of the robot will be reduced. This is guaranteed by control systems and multiple advanced sensors located on the base of the manipulator or on its wrist. A vision system is generally added to the robot through optical sensors. To follow the safety procedures, the collaborative robots are programmed to react immediately by stopping or inverting the position when they come into a human contact. Even in areas out of the reachable space of the manipulator where the operator does not get in contact with the robot but can be exposed with a dropped manipulated object, the robot is slowed down. There is usually an enable button that allows the robot motion only if it is pushed, otherwise the machine is stopped. Furthermore, robots are able to detect abnormal forces and torques applied to their joints by sensitive torque sensors to allow for safe collaboration. Hence, the robot is able to identify the impact into the obstacle, to analyse it in extremely short time and react. The speed and the position of the robots must be continually monitored. Since the cobots have lower payload capacity than traditionally robots and are typically lightweight with a rounded shape, the risk of injury is drastically reduced. Robot surfaces are softened with plastic materials or coated with a soft foamed material which is able to absorb energy of the impact. [1, 7].

These new generation of robots are designed to be flexible, portable and easy to program. They are user friendly because the operator can set simple motions with tablet or proper teach pedant based on a graphical interface. For complex motions, instead, it may be required the knowledge of some programming languages. It is usually possible to make the manipulator perform the self-learning modality, called hand guiding, in which the robot is driven by the operator along a path manually. In this case, the manipulator records each point of the track to define a trajectory that will be performed automatically. The collaborative robotics is constantly evolving also because the cobots have high sensitivity,

great accuracy and repeatability. All this makes it possible to perform complex works as assembly operations or machine enslavements [8, 9]. These new robots are able to place objects into precise slots where real-time corrections of position are needed because of their sensitivity. Moreover, they can follow the contour of the surface by applying specified force used for polished activity [1].

1.2. Collaborative robots for rehabilitation process

The reliability and security of collaborative robots gives rise to a new field of application: neuromuscular rehabilitation. These robotic systems can be used to a motor rehabilitation process of the upper limb in patients affected by neurological disease, spinal cord injury or ictus [10]. One of the main advantages of the robotic rehabilitation is to ensure the best reproducibility and a proper limb monitoring supervised by the physiotherapist. Therefore, robotic systems should be tested for efficacy and to identify the useful patient/specialist/robot interaction in the performing of individual exercises and creation of new rehabilitation protocols [11]. Robots, unlike humans, reach reliable measures of physical properties with levels of speed, accuracy, power and endurance over time. Reliability in the execution of repetitive tasks is high. In fact, robot systems compensate for the patient's inadequate strength and provide continuous feedback for the subjective perception of improvement. These characteristics make robotics a potential support in the rehabilitation domain for both trainers and patients. In this perspective, motor performance is expected to improve in speed and precision of movement thanks to the repetition of calibrated and replicable exercises in intensive training programmes [12]. Since the clinical picture of stroke patient is really complex, there are no generally guidelines on how to select the robotic intervention and how to customize it on the patient's residual functional abilities [13].

Today there are some rehabilitation devices on the market, as MOTORE and PABLO (1.2,1.3). The first device is used for the arm and shoulder rehabilitation, through which the patient can move the limb on a work plan getting the visual feedback. MOTORE is limited because of the only admissible movement, that is a planar motion. PABLO, on the contrary, is based on wireless sensors used only for specific applications as rehabilitation of the force hand grip without force feedback. There are other specific devices only for the rehabilitation of the fingers without involving the entire limb.

In recent years, the rehabilitation system is moving towards the collaborative robots with handle systems on their end-effector. This procedure promotes the study of human-robot interface, that can be a simple handle or a glove with the use of cameras (1.4). In this

1.2. Collaborative robots for rehabilitation process

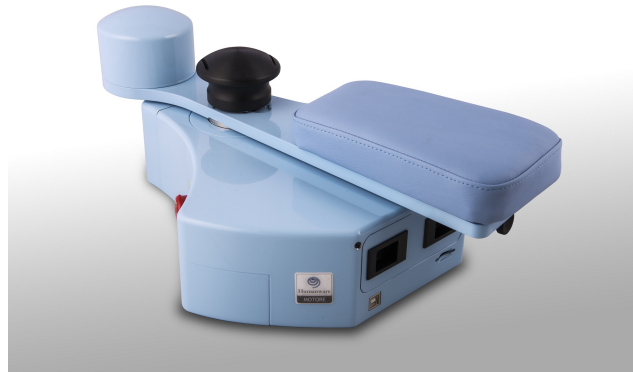


Figure 1.2.: MOTORE used for upper limb rehabilitation.



Figure 1.3.: PABLO devices used in rehabilitation process.

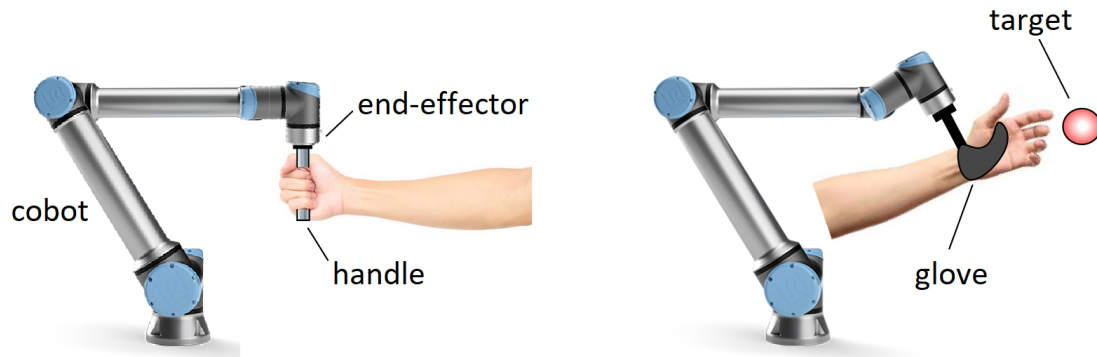


Figure 1.4.: Possible human-robot interfaces.

way it is possible to compute complex movements in the space [14–16]. An important advantage of end-effector based robots is that they are easy to adjust to different arm lengths [17]. Moreover, according to the severity of injury or to the rehabilitation process, the robots can work in three different modalities. The first is called passive modality, in which the patient is relaxed and the robot moves the limb through a predefined trajectory. In the active-assisted mode, instead, the patient triggers the initiation of the movement with his own force working with the robot as it moves the limb. In the last modality, the active one, the robot supplies a viscous resistance in the desired trajectory and in all other directions are blocked. The latest allows to the patient to reach toward the target with maximal effort [15]. ROBERT is an end-effector based robot already used in rehabilitation for the early mobilization of patients (1.5). This robot offers the active mobilization (assistive or resistive), in which patient’s function and strength determines the level of activity during exercise, and the passive mobilization for immobilized patients. ROBERT is very easy to set and use providing intensive treatment without increasing staff costs.

Universal RoboTrainer from Universal Robots company, instead, is another robot that has great potential as a training partner for patients with injury caused by stroke (1.6). In fact, RoboTrainer is used as a training partner since studies have shown that numerous repetitions of training exercises are an efficient way of helping the brain to learn to control paralysed muscles so that functional ability gradually returns. The robot’s sensors evaluate how much help the patient needs, ensuring that the exercise is to the appropriate level required by the patient. Another end-effector industrial based robot customized for rehabilitation purpose is Mitsubishi PA-10. It is generally used for the rehabilitation of the upper limb in patients affected by neurological disease. Optical systems are integrated in Mitsubishi PA-10 to record and acquire the arm movement with the aim of analyse

1.2. Collaborative robots for rehabilitation process



Figure 1.5.: ROBERT Life Science.



Figure 1.6.: Universal RoboTrainer.



Figure 1.7.: The MitsubishiPA10-7 robot platform.

Chapter 1. Introduction

patient deficit and define personal rehabilitation protocols (1.7).

The progress of technology allows to adapt the robot in response to patient's behaviour dynamically. From many studies and trials is deduced that the collaborative robots in the rehabilitation process can improve the quality of the physical exercise. Moreover, thanks to the three-dimensional motion, due to the subject reachable space, all the joints of the upper limb can be treated. In particular, the recovery time with the robot is faster than the traditional therapy, especially in movements as the forearm flexion or the shoulder adduction. Furthermore, the economic impact is moderate, because there are available cobots at a relatively low cost (about 30k€).

Automating rehabilitation training for the affected upper limb by robotic systems has been proposed to increase the number of repetitions of exercise without overburdening therapists and to maximize the patients attention and effort as well. In most of the cases, the robotic therapy can be associated with computer gaming environment to provide more motivating training context. This approach facilitates the subject learning and make the improvement faster. Indeed, this new therapy can possibly decreases the recovery of the patient, which means lower cost for healthcare systems. The use of robots in the medical field ensures an high intensity of treatment and the monitoring, or changing, of rehabilitation processes at distance, promoting so patient independence.

1.3. Robot structure

According to the scientific definition, the robot is seen as a machine that is able to modify the environment in which it operates. This is achieved by the actions that are conditioned by certain roles of behaviour intrinsic in the machine as well as by some data the robot acquires on its status and on the environment [18]. Robotic system is a complex system characterized by multiple subsystems: the mechanical system, the drive system, the control system and the sensing system. The mechanical system is the essential component that involves the manipulation apparatus (mechanical arms, end-effectors and artificial hands). The structure of a robot manipulator consists of a sequence of rigid bodies, known as links (1.8) that are interconnected by articulations (joints). The articulation between two consecutive links produces a rotational movement if the joint is a revolute joint or a translational movement if the joint is a prismatic one (3.8). Robots have typically anthropomorphic shape like human arm with their structure generally fixed (Figure 1.10) and used mostly in cooperation activities (Figure 1.11). These machines, which include an arm section, wrist section and hand section, are able to execute complex and agile movements. In fact, the robotic arm ensures mobility to the machine, the wrist confers

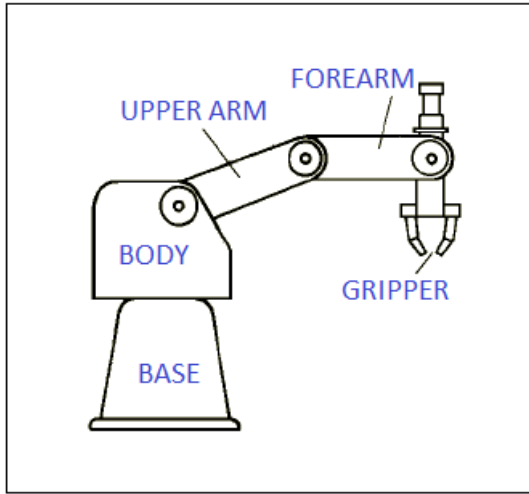


Figure 1.8.: Example of the links of a robot.

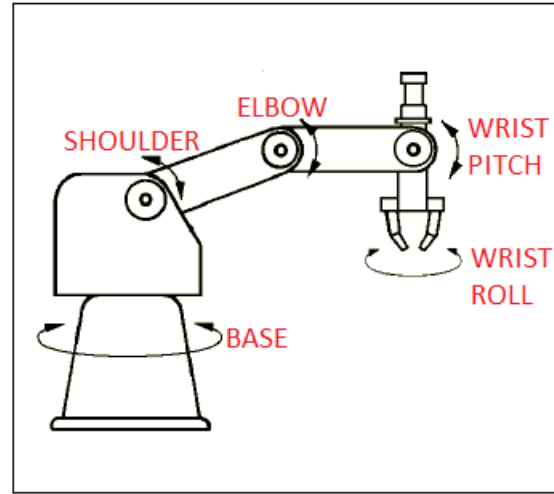


Figure 1.9.: Example of the joints of a robot.

dexterity and the end-effector performs a desired task [18]. Typically the robot manipulators are those with a fixed base, around which the robot movement occurs.

The execution of a robotic action is given by the actuation system, which animates the mechanical components of the robots converting energy into mechanical motions. In particular, the actuating system includes:

- a power supply
- a power amplifier
- a servomotor
- a transmission

The power supply is provided by the primary source and the power amplifier modulates the power flow, which is delivered for the desired motion performance. Actuation of joint motion is assigned to motors (servomotors) which allow the realization of a desired motion for the mechanical system. Servomotors can be pneumatic, hydraulic and electric. Since the robot motors need to have good trajectory tracking and positioning accuracy, they should have low inertia, high power-to-weight ratio, wide velocity range, high positioning accuracy and low torque ripple so as to attest continuous rotation even at low speed. The transmission, instead, transfers mechanical power from the motor to the joint. It is influenced by the power requirements, the kind of desired motion and the location of the



Figure 1.10.: Example of some anthropomorphic robots.

motor with respect to the joint. In fact, the transmission can convert the output of the motors in a quantitative way (velocity and torque) and in a qualitative way (a rotational motion about the motor axis into a translational motion of the joint) [18]. Therefore, all the actuating system is related to the motion control system associated with the drive system (servomotors, drives and transmissions).

The control system is the system that command the execution of actions in respect to the goals set by the task planning. The control action is generally executed by feedback loops to attain the desired position.

The sensing system, instead, is able to perceive the internal status of the robot and the external status of the environment. The internal status is captured by proprioceptive sensors (e.g. position transducers), while the external one is obtained by exteroceptive sensors (e.g. force sensors and cameras). The position transducers provide an electric signal proportional to the linear or angular displacement of the mechanical system related to a given reference position. The most common of this type are encoders and resolvers. The force sensors, instead, measure the effort induced by the force or torque applied to an extensible object. It is also possible to obtain indirect measurements of force with the estimation of small displacements. The vision sensors (cameras) measure the intensity of the light reflected by an object. The goal of all these sensors is to extract the features characterizing the interaction of the robot with the objects in the environment to enhance the degree of autonomy of the system.

Robots are often categorized by the number of degrees-of-freedom (DOFs) that ensure



Figure 1.11.: Example of anthropomorphic robot in cooperation activity.

the task execution. Since the programmed task usually involves the object location and orientation in three dimensional space, the robot is characterized by six DOFs: three for the position and three for the orientation. The robot complexity escalates as the number of DOFs increases. If more DOFs are available respect the minimum number required to execute a given task, the robot is redundant and become more flexible and versatile. In fact, manipulators with six DOFs can have some difficulties as singularities or obstructed workspace. Therefore, the seven DOFs robot arm, as human arm, constitute an excellent structure [19]. One of the main advantages of a redundant structure is that collisions with obstacles closer to the manipulator can be easily avoided. Moreover, its joint position and velocity limits are prevented and usually the robot motion performance is improved. On the other hand, the analysis of a redundant manipulator is more complex respect a non-redundant system.

1.4. Standards for collaborative operations

Since humans and robots interact in close proximity to each other, safety procedures and reliability standards must be complied by companies and operators. The Machinery Directive (2006/42/CE) issued by the European Parliament supplies a uniform protection level of safety and health protection for machines. The Directive explains first the economic and social impact of machines. Then, it is more focused on the risk assessment and on the

Chapter 1. Introduction

implementation of the resulting safety measures. EN ISO 12100 is the general standard for machinery safety [20], instead, a more specific standard for industrial robots is EN ISO 10218-1 and EN ISO 10218-2 developed by a committee of industry experts [21]. The first part of this ISO standard regarding the robot itself, while the EN ISO 10218-2 specifies all safety prerequisites for robot integration depicting dangerous situations and providing requirements to reduce risks. This standard, with reference to possible robot movement and space, explains the following definitions:

- Maximum space: is the space which can be swept by the moving parts of the robot plus the space which can be swept by the end-effector.
- Safeguarded space: is the space defined by the perimeter safeguarding devices.
- Restricted space: is a portion of the maximum space restricted by devices that establish limits.
- Operating space: is a portion of the restricted space that is actually used while performing all motions commanded by the task.

These standards are the basic guidelines for detecting and evaluating possible hazards to reduce the risk of accidents. Basically the risk identification, known as risk analysis, is performed combining operation modes with all possible activities, even unpredictable robot and operator behaviours. Then, the risk evaluation is conducted based on the probability of dangerous failure. Ultimately, the possible risks are reduced by different protective measures. All the procedure is repeated iteratively until the residual risk is minimal or belongs to an acceptable level, as shown in the Figure 1.12 [22]. Since the standard EN ISO 10218 lacks of threshold values for human-robot contact, the International Organization of Standardization published the ISO/TS 15066:2016 (Robots and robotic devices - Collaborative Robots), a technical specification able to update the existing robot safety standards [23]. This is not a standard, so it provides guidance not requirements. The specification first presents a general risk identification and assessment about human-robot collaboration. Then, many conditions for collaborative robot system applications are shown. This cooperation is also limited to a defined collaborative workspace, defined as the space where robot and human can perform tasks simultaneously. Four different basic safety procedures are highlighted in human-robot collaboration (Figure 1.13):

- Safety-rated monitored stop: robot stops when the operator is in the collaborative workspace and continues its process when the operator has left the collaborative workspace.

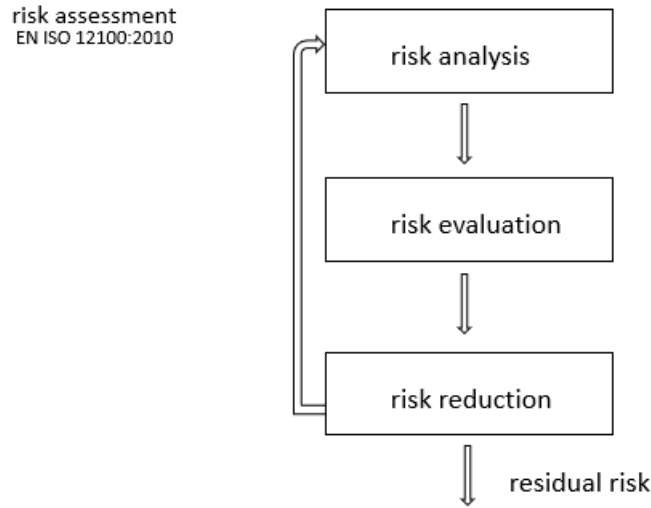


Figure 1.12.: Basic procedure of risk assessment.

- Hand guiding: robot movements are controlled by operator manually within the collaborative workspace, e.g. by an hand guiding device.
- Speed and separation monitoring: robot motion speed can increase or decrease depending on the distance between operator and robot. Moreover, if the distance gets below a minimum protection separation distance the robot stops.
- Power and force limiting: contact forces between operator and robot are technically limited to a safe level. Maximum pressure and forces for quasi-static and transient contact depend on different body parts. Quasi-static contact is a sustained trapping contact, instead transient contact is the most frequent type of collision with a short duration (<50 ms).

These four basic principles of protection in human-robot collaboration are used in many application areas, e.g. in manufacturing process where power and force usually need to be fixed. There could be different solutions within standard requirements with a resulting difference in safety level. For example, multiple small robots could be more appropriate than a large and heavy robot because of their kinetic energy is less than the large one. Hence, when both robots are moving with the same speed, the small robots are less harmful to a human if collision occurs [24]. There is also EN ISO 9283 standard, that is not directly related to safety procedures, but consists of performance criteria and related test methods in manipulating industrial robots [25]. Therefore, for every operations standards and laws

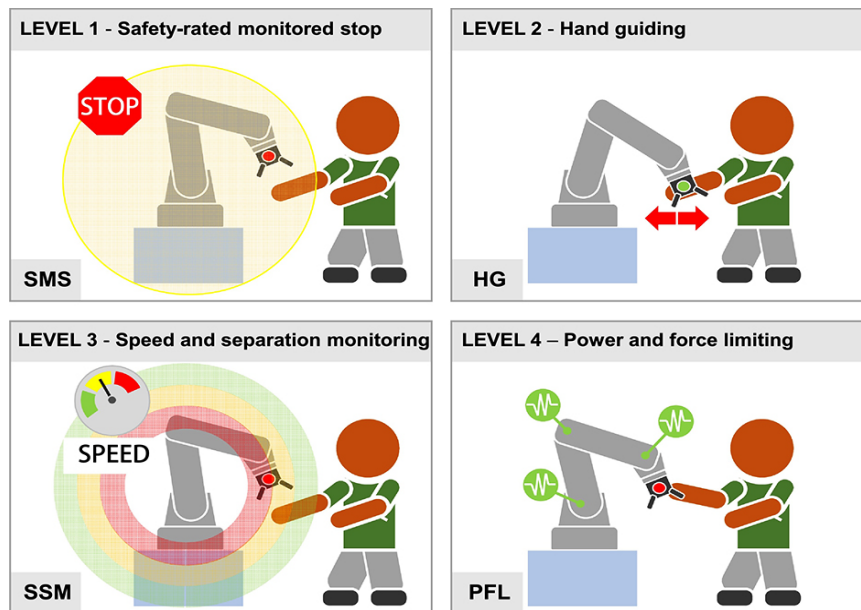


Figure 1.13.: Four basic safety procedures of ISO/TS 15066:2016.

should be respected. In the following tables (1.1, 1.2, 1.3) are summarized examples of the most know laws and general standards in EU [2].

Standards for collaborative operations are needed to ensure human safety and the proper machine operation. To obtain the minimal risk, not only simple human-robot collisions are analysed but all possible ways in which a robotic arm could hit a person is examined. Standards activities, including adoption of risk assessments and controls systems validation are necessary conditions where collaborative applications can be performed safely.

1.4. Standards for collaborative operations

Table 1.1.: EU Directives [2].

89/654/EC	Workplace Directive
2001/95/EC	Product Safety Directive
2004/108/EC	Electromagnetic compatibility Directive (EMC)
2006/42/EC	Machinery Directive (MD)
2006/95/EC	Low Voltage Directive (LVD)
2009/104/EC	Use of Work equipment Directive

Table 1.2.: General Standards [2].

EN ISO 12100	Safety of machinery - General principles for design - Risk assessment and risk reduction
EN ISO 13849-1/2	Safety of machinery - Safety-related parts of control systems - Part 1: General principles for design, Part 2: Validation
EN 60204-1	Safety of machinery - Electrical equipment of machines - Part 1: General requirements
EC 62061	Safety of machinery – Functional safety of safety-related electrical, electronic and programmable electronic control systems

Table 1.3.: Robot Standards [2].

EN ISO 10218-1	Robots and robotic devices - Safety requirements for industrial robots - Part 1: Robots
EN ISO 10218-2	Robots and robotic devices - Safety requirements for industrial robots - Part 2: Robot systems and integration
ISO/PDTS 15066	Robots and robotic Devices – Collaborative Robots

Chapter 2.

Collaborative robots

Cobots can work side-by-side with humans without safety fences, sharing the same workspace to perform manufacturing process. Indeed, in human-robot collaboration (HRC) is the human that determines a goal, while the robot work is to assist the operator and accomplish the collaborative task. Cobots are designed to improve particular functions like visual perception, action recognition, intent prediction, safe on-line motion planning, etc [26]. In the human-robot communication, these new robots are more intuitive, allowing non-expert operators to create and alter robot programs quickly and intuitively. Cobots are able to understand the state of the environment and perform actions based also on stochastic and unpredictable situations. In this way, the operator programs the machine during run-time or on-line.

The ability to be flexible is given by some programming features. First of all, there is the communication human-robot that can be verbal (with speech) or non-verbal (with gesture, gaze or head pose). The optimization is fundamental to generate desirable performance and important aspects of a cobot's surroundings (obstacles and tools positions) are mathematically modelled as a function of cobot's action [26]. Moreover, these robots are designed to minimize the human workload, energy consumption and time wasted or maximize the product quality and physical comfort. Then, a cobot learns skills similar to the human one with demonstrations, trials and errors. The user interface, with the proper teach pendant, is generally intuitive and user-friendly due to symbolic programming. A visual library of robot functions can be also utilized to create the program.

Cobots are lightweight, have lower payload and velocity respect the traditional one. They typically have a rounded shape without any pinch points and internalized wires. In particular, their load capacity is around 10 Kg and maximal motion velocity is limited to 250 mm per second. For this reason the robot is lightweight and it does not cause much damage after a human impact as standard robots. However, the robot is upgraded with sensors to prevent any collisions [1]. These robots can also feel abnormal forces on their path and, consequently, are programmed to stop when they perceive force overload or



Figure 2.1.: Cobots: Fanuc CR-35iA, ABB YuMi, UR5, KUKA LBR iiwa [1].

come into a human contact. In fact, they are designed to dissipate forces in case of impact on a wide surface. In any case, the most important feature for a correct HRC is to ensure safety for the cooperation of activities. Danger might arise when a robot loses balance and falls, this has to be avoided through control and suitable hardware design. Depending on the purpose of the robot and the desired ability, software and hardware have to be carefully chosen. Robot's sensor may vary widely and have to be customized according to the area of applications [27].

Collaborative robots are not designed to fully substitute current technologies. From the socio-economic aspect, these robots produce high competitiveness companies and are able to speed up some operations to increase the production process. Every dangerous situations can be easily avoided due to the many safety rules. Manufactures have adopted different strategies to guarantee safety of their products. In the Figure 2.1 are shown few examples of collaborative robots. From the left side, the first robot is the oldest type of manipulators. The second is ABB YuMi robot that is able to work as an operator. The last two are light weighted cobots with embedded system sensor [1]. Moreover, some cobots have sensor cover able to detect the obstacle before the collision itself due to capacitive sensors. Collision can be also predicted with vision systems that are usually joined with the manipulator. The safety of the robot is strictly related to safety of the technology placed on the end of it and everything attached at the end can decrease its safety. Moreover, the dangerousness increases when the robot is manipulated with an object with pointy part or when a drilling or welding head fixed to the robotic wrist. Therefore, to achieve successful collaborative activity is necessary to elaborate a detailed risk analysis. The cobot trajectory needs to have the following constraint:

$$distance \leq velocity \times Tb \tag{2.1}$$

where T_b is the braking time that depends on the robot payload. Indeed, robot motion should result from a compromise between production (maximum or nominal speed) and safety (guarded speed) [8].

Cobots can be used in a wide range of application as healthcare, construction, space applications, search and rescue. All of these areas have different requirements and the degree to which they are integrated varies [27]. The simplest way to use cobots is to consider them as tools that have to aid in the achievement of physical tasks, improving the process productivity. Under normal operations, cobots are able to work autonomously, however in any abnormal situations the human must intervene to make decisions on the robot behaviour. In this way, the system has the ability to deal with unexpected events [28]. Moreover, a proper HRC can be provided by force feedback system. In this case, the robot uses its force information (non-verbal communication) to generate the motion planning to collaborate in the execution of the task. HRC system takes advantage of varying levels of autonomy and multi-modal communication allowing the robotic system to work independently and ask its human counterpart for assistance when a problem is encountered. Cobots, with adjustable autonomy, enabling the system to vary the level of robotic system autonomy, increases productivity. Then, knowing what is happening in the robot's workspace is essential in a collaboration system.

2.1. Universal Robot UR5

Unlike traditional robots, which are big and expensive, this new generation of cobots is designed to be flexible, easy to program and safe for every cooperation activities, in accordance with national laws and standards. The Universal Robot (UR) company, founded in 2005, has developed three type of machines: UR3, UR5 and UR10. The first cobot sold by distributors in Denmark and Germany was the UR5 in 2008. In 2012, UR10 is launched and the UR3 appeared only in 2015 (Figure 2.2 a). These cobots can be used in various industrial activities and the choice of the best suited UR model depends on the kind of work and on the possible load to be moved.

The UR company has recently made a new series of robots by improving the previous one. The e-series cobots (eUR3, eUR5, eUR10) are equipped with more intuitive programming and additional components to be employed in multiple industry production. They are more versatile with an increase number of safety functions. However, being the e-series an upgrade of the UR versions, the technical specifications, the mechanical and electrical structures are the same. Each robot of UR is named after their payload capacity (kilograms). In fact, the UR3 is the smallest one and is perfect for light assembly tasks and

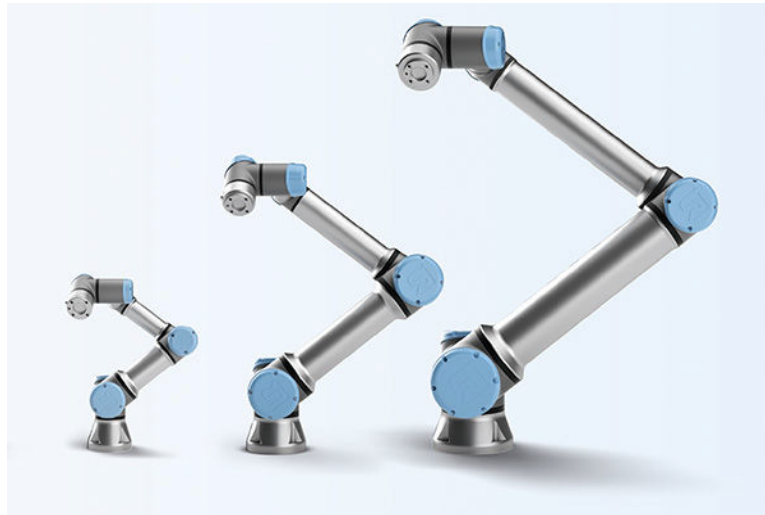
jobs for absolute precision. The largest robot arm (UR10), instead, is the one with most power capable to reach radius of up to 1300 mm.

However, the one used in rehabilitation process and the most widespread is the UR5, slightly bigger than UR3 and strikes the perfect balance between size and power. In particular, it has internal controller safety mechanism that stops all movements of the manipulator within 500 ms if the force action on the tool-center-point (TPC), i.e. the center point of the output flange, exceeds 25 Kgm/s. Thus, the force control assures that the robot arm will still approach the specified force. UR5 is ideal for automating low-weight processing tasks (up to 5 Kg) like picking, placing and testing. It is capable to reach radius up to 850 mm from the base joint (robot workspace) and combine compact measures (about 20 Kg with a footprint of 149 mm). The robot, as shown in Figure 2.2 b, is an arm with 6 DoF composed extruded aluminium tubes and six joints (base, shoulder, elbow and three wrist joints). Therefore, all its six joints contribute to the transformational and rotational movement of its end-effector [3]. The UR5 weight is 18.4 Kg with each joint ranges of $\pm 360^\circ$. The speed limits of all joints correspond to $\pm 180^\circ/\text{s}$, while the tool speed is typical 1 m/s. The repeatability is $\pm 0.1\text{mm}/\pm 0.0039$ and the robot can work in a temperature range of 0-50°C [3]. An infinite-spin lets the robot to perform an infinite rotation around its axis for generating spin works without other devices. The electrical equipment of UR5 is composed by a controller, for digital and analog control signals; a small connector at the tool end that provides power and control signals for gripper and sensors; an Ethernet connection is provided at the bottom of the control box. To energize the robot, the control box needs to be connected to the mains with an input voltage range of 100-240 VAC and input frequency range of 47-63 Hz [3]. Some of the main technical specifications of the UR robots are reported in the Table 2.1.

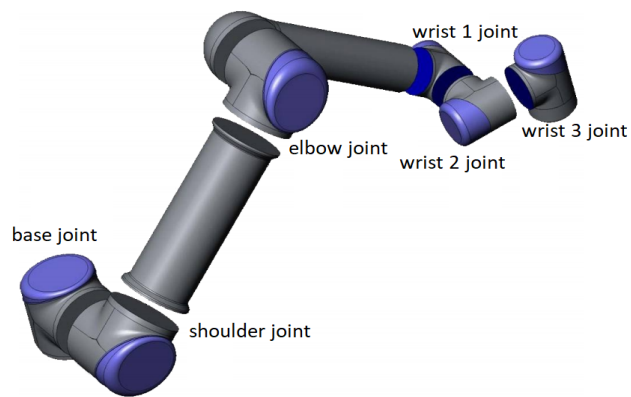
To initialize the UR5, a sequence of movements is required to sort out the various part. Programming is quick and easy thanks to the innovative touchscreen. The robot can be programmed with UR script language or directly through the interface using the teach pendant (Figure 2.3). One advantage of using remote script commands is that the program can be further developed and edited while the robot is running production. In fact, the differences in using this method and the teach pendant is that in the second one the program execution has to be stopped while editing, and the actual robot work needs to be interrupted. The choice of programming environment is up to the programmer and typically programming languages are C++, Java, Visual Basic or Phyton. Using the teach pendant, instead, the robot program can be easily executed with Polyscope interface. In this manner, four different methods can be adopted to move the machine:

- Move command: corresponds to simple movements from one point to another (in

2.1. Universal Robot UR5



(a) Universal Robot: UR3, UR5, UR10.



(b) UR5 joints.

Figure 2.2.: UR Company production.

Table 2.1.: Technical specifications [3].

		UR3	UR5	UR10
Weight		11 kg	18.4 kg	28.9 kg
Payload		3 kg	5 kg	10 kg
Reach		500 mm	850 mm	1300 mm
Joint ranges		$\pm 360^\circ$	$\pm 360^\circ$	$\pm 360^\circ$
Speed	base joint	$\pm 180^\circ/\text{s}$	$\pm 180^\circ/\text{s}$	$\pm 120^\circ/\text{s}$
	shoulder joint	$\pm 180^\circ/\text{s}$	$\pm 180^\circ/\text{s}$	$\pm 120^\circ/\text{s}$
	elbow joint	$\pm 180^\circ/\text{s}$	$\pm 180^\circ/\text{s}$	$\pm 180^\circ/\text{s}$
	1° wrist joint	$\pm 180^\circ/\text{s}$	$\pm 180^\circ/\text{s}$	$\pm 180^\circ/\text{s}$
	2° wrist joint	$\pm 180^\circ/\text{s}$	$\pm 180^\circ/\text{s}$	$\pm 180^\circ/\text{s}$
	3° wrist joint	$\pm 180^\circ/\text{s}$	$\pm 180^\circ/\text{s}$	$\pm 180^\circ/\text{s}$
	Tool	1 m/s	1 m/s	1 m/s

Cartesian or Joint space).

- Servo command: used for much tighter robot control, in which the machine is moved toward the desired position and then, all motion are blocked.
- Speed command: makes the robot move with constant velocity.
- Force command: makes the robot compliant along one or more axes while is still moving along the other axes.

Moreover, the UR5 can be moved directly by the operator manually, adopting the self-learning modality, in which the robot records each point of the track to define a motion. In manual hand-guiding applications with linear movements, is suggested to put the joint speed limit of the base and shoulder at $40^\circ/\text{s}$ to prevent fast movements.

The UR5 complies with EN ISO 10218, implying that the robot can operate in close proximity to humans and have several built-in safety mechanisms such as protective stop that triggers if an external force over 150 N is applied to the robot [29]. The UR5 has a number of safety-related functions that can be used to limit the movement of its joints and of the robot TCP. The basic limiting safety-related functions are reported in Table 2.2, designed basically for collaborative robot applications. But, meanwhile, advanced limiting safety functions have been reached. Indeed, when the robot is no more in normal conditions (i.g. in human collision), the reduced configuration is adopted, in which the machine has



Figure 2.3.: UR5 and teach pendant with Polyscope Robot User Interface.

new and more restrictive safety limits. Instead, when safety limits are violated, the robot stops and the safety system must be restarted by the recovery mode. In this modality, is not possible to program any motions, but the robot arm needs to be manually moved back within limits. In extreme cases it could be necessary to stop immediately the robot through an emergency stop button (the red button) located on the teach pendant.

The robot can be adapted to various industry processes with different accessories: from grippers for end-effector to vision system and software. For example, for assembly operations with UR5, the AG-95L adaptive gripper or ROBOTIQ AIRPICK (a vacuum gripper) are suitable (Figure 2.4). The vision system can be also added for having a guidance 3D (or 2D) or a vision process. Moreover, the robot can be easily assembled and installed on the floor, on the wall, on the ceiling and even on a moving platform. Hence, these cobots can be easily re-deployed to multiple applications without changing the production layout. Easy programming, fast set-up, flexible deployment and safe collaborative applications are the main advantages of the UR production.



Figure 2.4.: Examples of grippers and vision system suitable for UR5. From the left: AG-95L gripper, ROBOTIQ AIRPICK gripper and 3D-TOF sensors

Table 2.2.: Basic limiting safety-related functions in UR5 [3].

Limiting Safety Function	Description
Joint position	Min. and max. angular joint position
Joint speed	Max. angular joint speed
TCP position	Planes in Cartesian space limiting robot TCP position
TCP speed	Max. speed of the robot TCP
TCP force	Max. pushing force of the robot TCP
Momentum	Max. momentum of the robot arm
Power	Max. applied robot arm power



Figure 2.5.: KUKA KMR iiwa

2.2. KUKA LBR iiwa

KUKA company, founded in 1898, has developed in 2004 the first cobot: KUKA LBR 3. Today, this company has until 25 variety of robots, which differ one from the other for their own technical specifications (i.g. maximum payload or radius of action) and their field of applications. Only in the 2014 the KUKA LBR iiwa is launched on the market. The LBR abbreviation stands for "lightweight robot", while iiwa is for "intelligent industrial work assistant". In fact, it is a lightweight robot, generally applied for delicate assembly work, used without safety fences for HRC. The KMR iiwa, instead, is an autonomous LBR iiwa robot, because of the strength combinations of the sensitive LBR iiwa robot with those of a mobile, autonomous platform (Figure 2.5). Since the KMR iiwa is location-independent and high flexible, the industrial production is greatly increased. The cobot LBR iiwa is available in two versions with two different payload capacity (7 kg and 14 kg) and, consequently two radius of action (800 mm and 820 mm), respectively LBR iiwa 7 R800 and LBR iiwa 14 R820. There is another model on the same production line (LBR iiwa 7 R800 CR), with the same characteristics of the 7 R800, but specifically suitable for the installation on the floor. However, the KUKA LBR iiwa models have all the same mechanical and electrical structure and their systems typically comprise the manipulator, the controller, connecting cables and the end-effector (tool). The reaction time is closed to zero thanks to their joint torque sensors, which can detect immediately any sort of

contact reducing their level of force and speed. The KUKA LBR iiwa is a 7-axis arm and each axis is composed by torque, range and temperature sensors. They are able to provide signals for robot control, as position control and impedance control. All drive units and current-carrying cables, instead, are located inside the robot. The robot is redundant for the 7-axis allowing high versatility and accuracy. Its main components are shown in the Figure 2.6, where is also reported the location of motors (from A1 to A7). The weight of the LBR iiwa 7 R800 is approximately 23.9 kg, with a volume of working envelope of $1.7 m^2$. The pose repeatability is ± 0.1 mm and the temperature range available of robot working is 5-45°C. The robot can be installed on the floor, on the wall and on the ceiling. The KUKA LBR iiwa 14 R820, instead, has slightly different technical specifications, as the weight, that is about 29.9 kg and the pose repeatability (± 0.15 mm). In Table 2.3 are reported for each axis the motion range and the speed with rated payload. Moreover, KUKA LBR iiwa system has a rounded shape in order to minimize dangerous collisions. The electrical installation includes all the supply and control cables for the motor axis A1 to A7. All the connections on the motors are plug-and-socket connections. The entire cabling is routed internally in the robot and the connecting cable is connected to the robot controller. In these models the electronics have a maximum packing density and thanks to the joint torque sensors and safety-oriented position sensors is possible to ensure the safety technology in a minimal space 2.7. The robot control is conducted by KUKA Sunrise Cabinet, an high-speed servo control with the smartPAD, an user-friendly interface. Thanks to the LBR iiwa operating system (KUKA Sunrise.OS) it is possible to combine motion and sensor systems for a proper dynamic interaction of kinematic systems with multiple arms. The graphical interface has an intuitive operation with self-explanatory symbols and many powerful user-friendly functions. In fact, the program is automatically illustrated in a structured manner characterized by block diagram, which can be used for the entire design process (planning, programming and maintenance). The KUKA Sunrise.OS toolboxes can be interfaced with Matlab toolbox allowing the reproduction of programs immediately and, thanks to the integrated Java libraries, is possible to compiled process expertise. Out-of-the-box, the robot can only be controlled through a Java based program and all the sensory data or information related to the current task is only available locally on the robot control system or in the KUKA smartPAD. [30,31].

KUKA LBR iiwa is built in accordance with safety rules and used only by safety-conscious users that are aware of the risks involved in its operations. All safety-relevant mechanical and electromechanical components of the industrial robot are tested for correct functioning and a local emergency stop button is located on the teach pendant. There are some safety stop functions (Safety Stop 0 and Safety Stop 1) that interrupt the normal robot

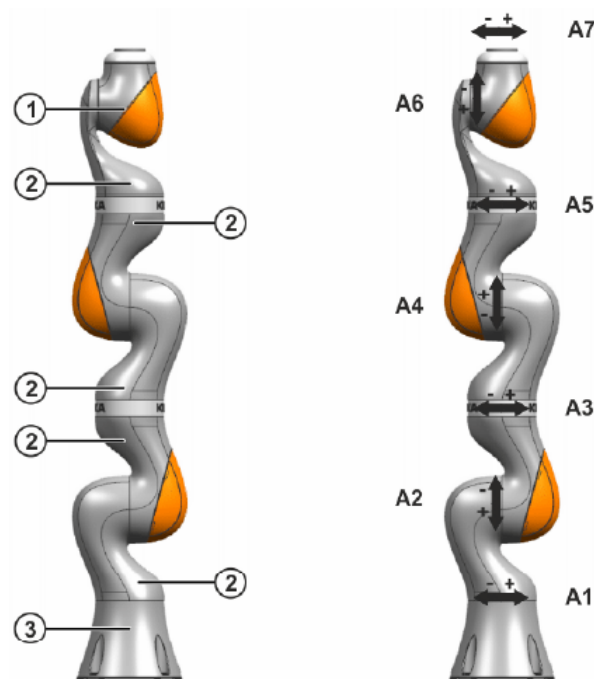


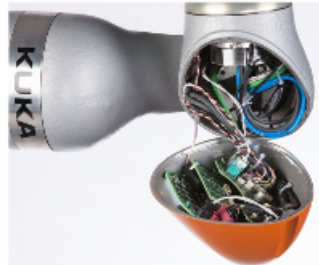
Figure 2.6.: KUKA LBR iiwa: (1) In-line wrist, (2) Joint module, (3) Base frame, A1-A7 motors

work and bring all its motions to a standstill. Indeed, in the Safety Stop 0 all drivers are deactivated immediately and the brakes are turned on. In the Safety Stop 1, instead, the manipulator is braked and is brought to a standstill with the drivers. In fact, the manipulator remains in the programmed path and the drivers are switched off after 1 s at the latest. This last modality is only supported by LBR iiwa. To minimize the risk of a HRC, KUKA company provides others safety functions related to velocity monitoring, workspaces, safeguarded zones, force monitoring, collision and tool detection.

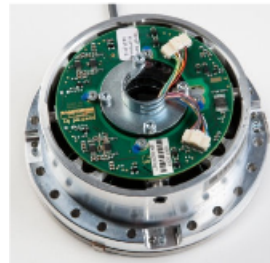
Therefore, LBR iiwa robots are extremely sensitive with their high-performance servo control and each component can be easily mounted with the utmost precision. In particular, an axis-specific torque accuracy is $\pm 2\%$ respect the maximum torque. The implementation of these cobots can help the operators in their work, without creating crushing or shearing hazards.

Table 2.3.: LBR iiwa: Motion range and Speed with rated payload for each axis.

		Motion range	Speed with rated payload
LBR iiwa 7 R800	A1	$\pm 170^\circ$	$\pm 98^\circ/\text{s}$
	A2	$\pm 120^\circ$	$\pm 98^\circ/\text{s}$
	A3	$\pm 170^\circ$	$\pm 100^\circ/\text{s}$
	A4	$\pm 120^\circ$	$\pm 130^\circ/\text{s}$
	A5	$\pm 170^\circ$	$\pm 140^\circ/\text{s}$
	A6	$\pm 120^\circ$	$\pm 180^\circ/\text{s}$
	A7	$\pm 175^\circ$	$\pm 180^\circ/\text{s}$
LBR iiwa 14 R820	A1	$\pm 170^\circ$	$\pm 85^\circ/\text{s}$
	A2	$\pm 120^\circ$	$\pm 85^\circ/\text{s}$
	A3	$\pm 170^\circ$	$\pm 100^\circ/\text{s}$
	A4	$\pm 120^\circ$	$\pm 75^\circ/\text{s}$
	A5	$\pm 170^\circ$	$\pm 130^\circ/\text{s}$
	A6	$\pm 120^\circ$	$\pm 135^\circ/\text{s}$
	A7	$\pm 175^\circ$	$\pm 135^\circ/\text{s}$



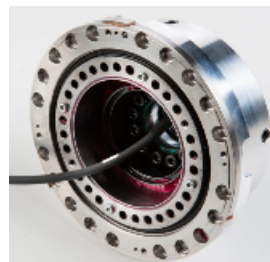
(a) Joint electronics



(b) Joint torque sensors



(c) Motor



(d) Gear unit

Figure 2.7.: Electrical and mechanical components of LBR iiwa.

Chapter 3.

Kinematic models

Mathematical models of robotic and human arm are indispensable tools for the study of kinematics. Indeed, the kinematic model analyzes the movement of a rigid body (i.g. robot and human arm) not considering all forces and torques that cause its motion. Therefore, in order to manipulate an object in the space, it is necessary describe the end-effector position and orientation (pose), related to the base of the manipulator. This chapter is dedicated first to the planning mode of trajectory motion through point-to-point modality. It is followed by the study of the direct kinematics equation of human and robotic arm through a linear algebra approach. The pose will be expressed as a function of the joint variables of the mechanical structure with respect to a reference frame. At the end, there are the solutions of the inverse kinematics problem, in which the joint variables corresponding to a given end-effector pose are determined. The kinematic models are all implemented Matlab environment.

3.1. Motion Planning

Trajectory planning algorithms allow the execution of a specific task, generating reference inputs to the motion control system, which guarantees the achievement of the planned trajectory. Planning consists of generating a time sequence of the values acquired by an interpolating function (i.g. polynomial) of the desired trajectory. There are two possible techniques for trajectory generation: point-to-point motion and motion through a sequence of points. In the first modality, the initial and final point are assigned, while in the second one, a finite sequence of points along a specific path are given. In this study only the first modality will be treated.

A trajectory is a path on which a timing law is specified, in terms of velocities and/or accelerations at each point. Moreover, to plan a desired trajectory is necessary to specify the geometric path and the motion law, both constraints on the continuity (smoothness) of the trajectory and on its time-derivatives up to a given degree. Indeed, the geometric

Chapter 3. Kinematic models

path can be defined in the work-space or in the joint-space, as

$$\begin{aligned} p &= p(s) && \text{in work-space} \\ q &= q(\sigma) && \text{in joint-space} \end{aligned} \tag{3.1}$$

where $s = s(t)$ and $\sigma = \sigma(t)$. Typically, a geometric path cannot be fully specified by the user, but only reduced number of parameters, as extremal points or possible intermediate points. The work-space corresponds to the volume or the complete set of poses which the end-effector can reach, while the joint-space is the configuration assumed by each joint of the manipulator.

The motion timing law, instead, is necessary to specify continuous function up to a given order of derivations (at least the first and the second order, i.e. velocity and acceleration).

$$s(t) = a_0 + a_1t + a_2t^2 + \dots + a_nt^n \tag{3.2}$$

Also, this is not typically specified at each point of the geometric path, but it regards the total trajectory time, the constraints on the maximum velocities and accelerations. On the basis of these informations, the trajectory planning algorithm generates a time sequence variables that describe end-effector position and orientation over time in respect of the imposed constraints. In point-to-point motion, the manipulator moves from an initial to a final joint configuration in a given time t_f , optimizing its performance. The optimization process can be determined by solving differential equations with the known constraints: the initial position q_i , velocity \dot{q}_i and acceleration \ddot{q}_i and the corresponding final values q_f , \dot{q}_f and \ddot{q}_f . In these algorithms also the initial and final time of the trajectory has been assigned.

The first trajectory considered is the trapezoidal velocity profile, in which there is a constant acceleration in the starting phase, a cruise velocity, and a constant deceleration in the arrival phase. This causes a trajectory formed by a linear segment that connects two parabolic segments to the initial and final positions. Both initial and final velocities are assumed to be null and the segments with constant accelerations have all the same time duration, having an equal magnitude. From the Figure 3.1 can be noted that the trajectory is symmetric respect to an average point (q_m), at the time $t_m = t_f/2$. To ensure the transition from q_i to q_f in t_f , the velocity at the end of the parabolic segment must be equal to the constant velocity of the linear segment:

$$\ddot{q}_c t_c = (q_m - q_c)/(t_m - t_c) \tag{3.3}$$

3.1. Motion Planning

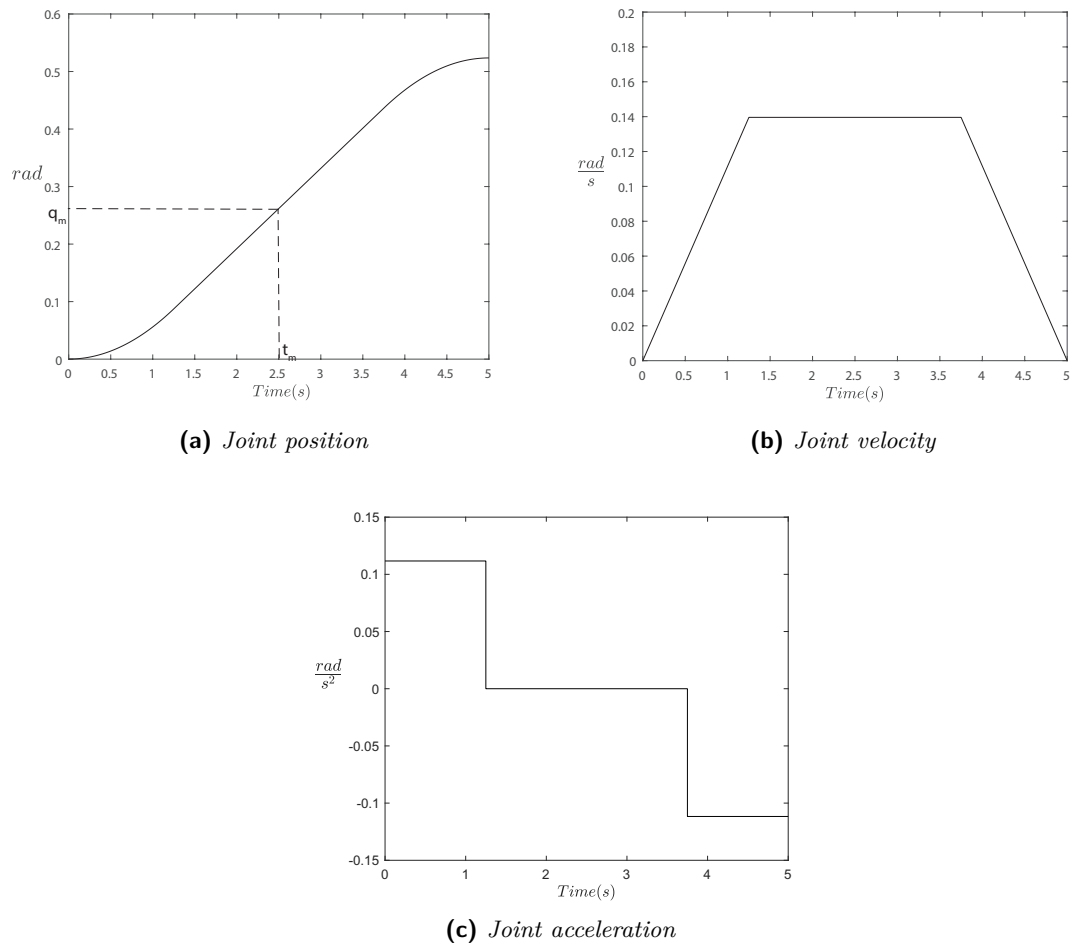


Figure 3.1.: Timing law with trapezoidal velocity profile.

Chapter 3. Kinematic models

where q_c is the value of the joint variable at the end of the parabolic segment at time t_c with constant acceleration \ddot{q}_c , having $\dot{q}(0) = 0$. Hence, is

$$q_c = q_i + \frac{1}{2}\ddot{q}_c t_c^2 \quad (3.4)$$

Combining 3.3 and 3.4 gives

$$\ddot{q}_c t_c^2 - \ddot{q}_c t_f t_c + q_f - q_i = 0 \quad (3.5)$$

For given t_f , q_i and q_f , the solution for t_c is computed from 3.5, as $t_c \leq t_f/2$

$$t_c = \frac{t_f}{2} - \frac{1}{2}\sqrt{\frac{t_f^2 \ddot{q}_c - 4(q_f - q_i)}{\ddot{q}_c}} \quad (3.6)$$

Acceleration is then subject to the constraint

$$|\ddot{q}_c| \geq \frac{4|q_f - q_i|}{t_f^2} \quad (3.7)$$

When the acceleration \ddot{q}_c is chosen so as to satisfy 3.7 with the equality sign, the resulting trajectory does not feature the constant velocity segment any more and has only the acceleration and deceleration segments, giving another planning trajectory, i.e. the triangular profile.

Considering the 3.2, has been chosen a fifth-order polynomial function to generate a valid joint trajectory (Figure 3.2). The degree $n = 5$ of the polynomial depends on the number of boundary conditions (q_i , q_f , \dot{q}_i , \dot{q}_f , \ddot{q}_i , \ddot{q}_f) and on the smoothness of the trajectory. Hence, the motion timing law for a generic joint is given by

$$q(t) = a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0 \quad (3.8)$$

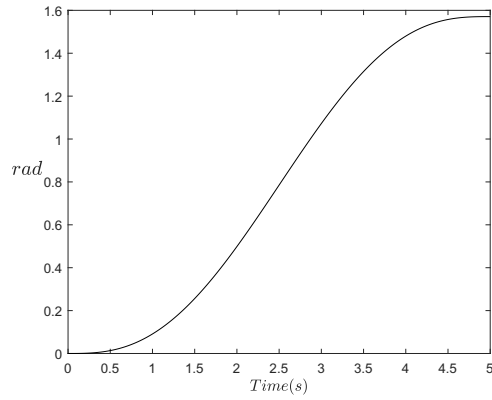
whose coefficients are computed by imposing the conditions for $t = 0$ and $t = t_f$ on the joint variable $q(t)$ and on its first two derivatives. The parabolic velocity profile is

$$\dot{q}(t) = 5a_5 t^4 + 4a_4 t^3 + 3a_3 t^2 + 2a_2 t + a_1 \quad (3.9)$$

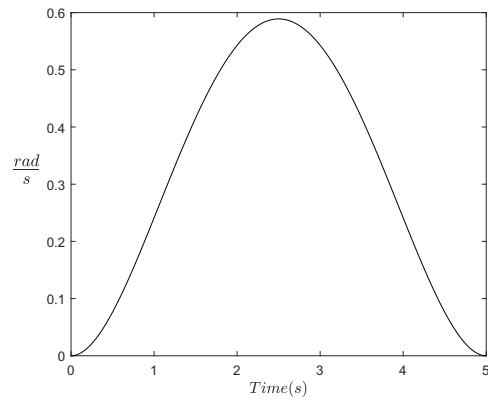
and a linear acceleration profile

$$\ddot{q}(t) = 20a_5 t^3 + 12a_4 t^2 + 6a_3 t + 2a_2 \quad (3.10)$$

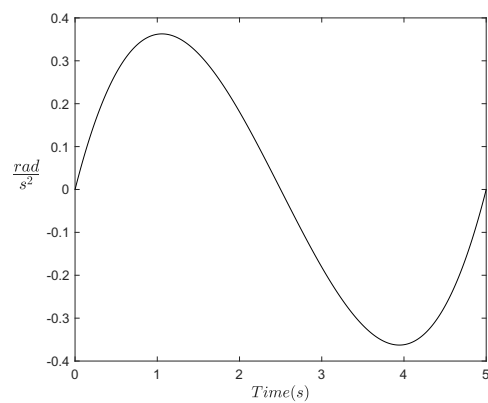
3.1. Motion Planning



(a) Joint position



(b) Joint velocity



(c) Joint acceleration

Figure 3.2.: Timing law of polynomial function of the 5th order.

Since six coefficients are available, it is possible to impose the initial and final joint position values q_i and q_f , the initial and final velocity, generally set to zero ($\dot{q}_i = 0$ and $\dot{q}_f = 0$), the initial and final acceleration $\ddot{q}_i = 0$ and $\ddot{q}_f = 0$. The determination of a specific trajectory is given by the solution to the following system of equations:

$$\begin{aligned}
 q_i &= a_0 \\
 q_f &= a_5 t_f^5 + a_4 t_f^4 + a_3 t_f^3 + a_2 t_f^2 + a_1 t_f \\
 \dot{q}_i &= a_1 \\
 \dot{q}_f &= 5a_4 t_f^4 + 4a_3 t_f^3 + 3a_2 t_f^2 + 2a_1 t_f + a_0 \\
 \ddot{q}_i &= 2a_2 \\
 \ddot{q}_f &= 20a_5 t_f^3 + 12a_4 t_f^2 + 6a_3 t_f + 2a_2
 \end{aligned} \tag{3.11}$$

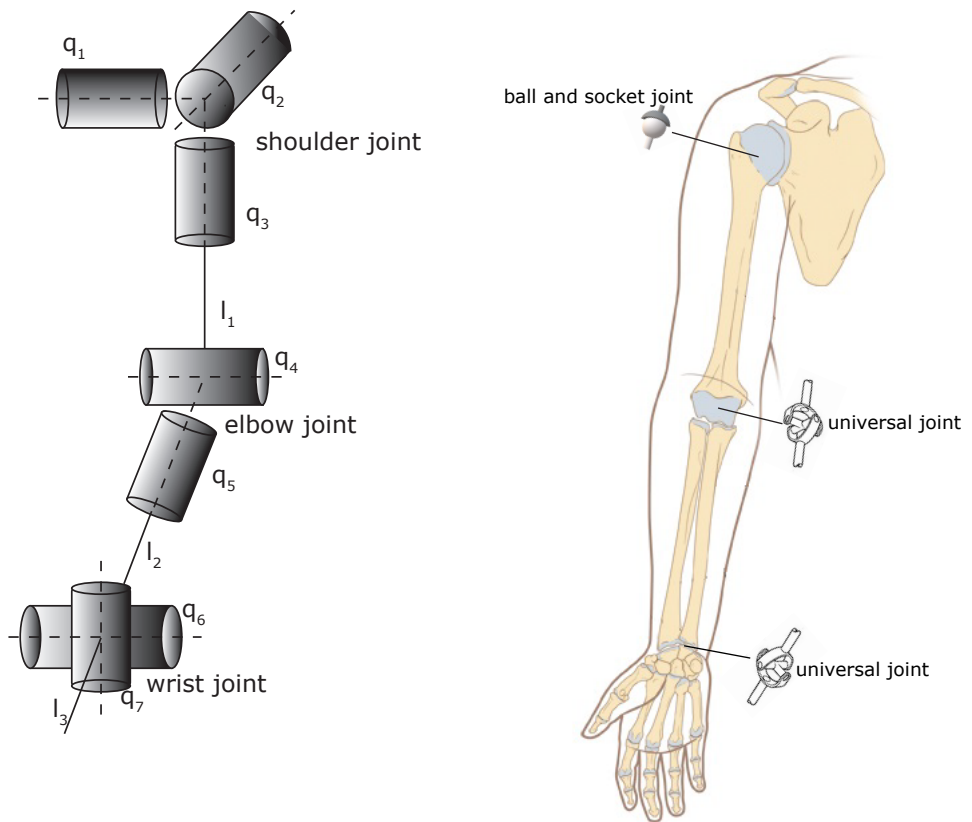
In this study has been adopted the polynomial 5th order trajectory profile due to its greatest performance index and its significant accuracy.

3.2. Direct and Inverse Kinematic of human arm

To define a mathematical model of the human arm it is necessary a first definition of the arm. In the studies of Desmurget and Prablanc [32], Lemay and Crago [33] and Raikova et al [34], the human arm is seen as a serial mechanism of three joints (shoulder, elbow, wrist) and tree segments (upper arm, forearm, hand). The hand is greatly simplified because of its complexity and high number of DoF, and consequently, should be studied separately [35]. The proposed kinematic model of the arm, for the purpose study, consists in the representation of the human arm as three rigid segments connected by frictionless joints with a total 7 DoF [4].

Since human joints are not ideal and do not hold a reference location to the segment they connect, to model the human joints is opportune to perform an high simplification. The human shoulder is considered as a simple ball-and-socket joint, where the ball-shaped surface of one rounded bone fits into the cup-like depression of another bone. This joint allows three rotational DoFs, attributed to flexion-extension (q_1 joint) abduction-adduction (q_2 joint), and external-internal rotation of the humerus relative to the scapula (q_3 joint). The elbow joint, instead, is seen like an universal joint with two DoFs, which are flexion-extension (q_4 joint) and pronation-supination (q_5 joint). The wrist joint is also related to an universal joint able to perform the flexion-extension (q_6 joint) and abduction-adduction (q_7 joint). Each joint is shown in Figure 3.3. The axes in elbow and wrist joints are assumed to be orthogonal and intersecting. The upper arm, the forearm and the hand are approximated as rigid bodies [36]. Even though this model is largely accepted, it is not

3.2. Direct and Inverse Kinematic of human arm



(a) Kinematic structure of human arm.

(b) Joints with 3 DoF (ball and socket joint) and 2 DoF (universal joint).

Figure 3.3.: Representation of upper limb joints.

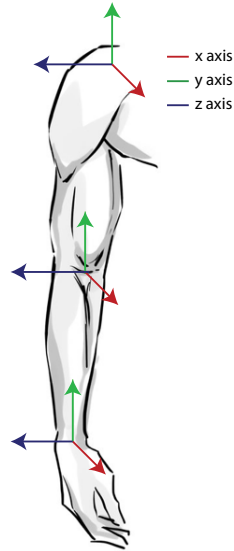


Figure 3.4.: Human arm axis convention

totally exact. In particular, the forearm segment is not rigid because of the rotations of the radius relative to the ulna and the center of the shoulder joint as well as the axes of rotations in the elbow and wrist can move relative to the adjacent segments during the movements. However, it is a reasonable compromise between accuracy and simplicity of the human arm description.

The representation of the arm's end-effector pose (i.e. the pose of the hand), through the geometries of the limb and the initial joint position, gives the study for the Direct Kinematics of an open chain. Initially, the reference frame (Figure 3.4) for each joint has been attached based on the convention and adopted in OpenSim software. The x axis corresponds to the sagittal axis, the y axis is the longitudinal axis and the z axis is fixed to the transversal axis. It was considered a man with a height of 1.70 m and all the anthropometric data with respect to arm lengths and joint limits are reported respectively in Table 3.1 and in Table 3.2 .

The direct kinematic expressed by the homogeneous transformation matrix of the end-

3.2. Direct and Inverse Kinematic of human arm

Table 3.1.: Anthropometric values for human arm [4].

Lengths (m)	Formula
upper arm	$0.161 \cdot height$
forearm	$0.146 \cdot height$
hand	$0.106 \cdot height$

Table 3.2.: Admissible range for human arm joints [5].

Joint		Minimum value	Maximum value
Shoulder	flexion-extension	-90°	$+90^\circ$
	abduction-adduction	-120°	$+90^\circ$
	external-internal rotation	-90°	$+90^\circ$
Elbow	flexion-extension	0°	150°
	pronation-supination	0°	180°
Wrist	flexion-extension	-70°	$+70^\circ$
	abduction-adduction	-25°	$+35^\circ$

effector related to the base is given by

$$\mathbf{T}_e^b(\mathbf{q}) = \left[\begin{array}{ccc|c} \mathbf{R}_e^b(\mathbf{q}) & & & \mathbf{p}_e^b(\mathbf{q}) \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (3.12)$$

where $\mathbf{R}_e^b(\mathbf{q})$ is the rotational matrix that represents the orientation of the end effector respect the base frame and $\mathbf{p}_e^b(\mathbf{q})$ is the position vector of the origin of such a frame with respect the origin of the base frame. Both are function of joint \mathbf{q} .

Since the human arm has been approximated as an open chain composed by consecutive links, the arm kinematic is calculated from Link 0 to Link n , i.e. from the shoulder (the base) to the hand (end-effector). The direct kinematics function is obtained by simple products of each homogeneous transformation matrices in a recursive way, as

$$\mathbf{T}_e^b(\mathbf{q}) = \mathbf{T}_0^b \mathbf{T}_n^0(\mathbf{q}) \dots \mathbf{T}_e^n \quad (3.13)$$

where \mathbf{T}_0^b and \mathbf{T}_e^n are typically two homogeneous transformations that describes respectively the pose of Frame 0 with respect to the base frame and the end-effector frame with respect to Frame n . In this study, starting from the three rotations of the shoulder, it is possible to define three homogeneous transformations:

$$\begin{aligned} \mathbf{T}_1^s(\mathbf{q}_1) &= \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 & 0 \\ \sin q_1 & \cos q_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ \mathbf{T}_2^1(\mathbf{q}_2) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos q_2 & -\sin q_2 & 0 \\ 0 & \sin q_2 & \cos q_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ \mathbf{T}_3^2(\mathbf{q}_3) &= \begin{bmatrix} \cos q_3 & 0 & \sin q_3 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin q_3 & 0 & \cos q_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (3.14)$$

3.2. Direct and Inverse Kinematic of human arm

Considering the position vector of the upper arm from the shoulder to the elbow joint $\mathbf{p}_{1 \times 3}$, expressed as

$$\mathbf{T}_{\mathbf{p}_1}^3 = \left[\begin{array}{ccc|c} \mathbf{I}_{3 \times 3} & & & \mathbf{p}_{1 \times 3} \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (3.15)$$

where $\mathbf{0}_{1 \times 3}$ is a zero vector with (1×3) dimension. To find the forward kinematic for the shoulder, a pre-multiplication of the previous homogeneous matrices has been performed:

$$\mathbf{T}_{\mathbf{p}_1}^s = \mathbf{T}_1^s \mathbf{T}_2^1 \mathbf{T}_3^2 \mathbf{T}_{\mathbf{p}_1}^3 \quad (3.16)$$

For the elbow there are only two DoFs, therefore:

$$\mathbf{T}_4^{\text{el}} = \begin{bmatrix} \cos q_4 & -\sin q_4 & 0 & 0 \\ \sin q_4 & \cos q_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.17)$$

$$\mathbf{T}_5^4 = \begin{bmatrix} \cos q_5 & 0 & \sin q_5 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin q_5 & 0 & \cos q_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{\mathbf{p}_2}^5 = \left[\begin{array}{ccc|c} \mathbf{I}_{3 \times 3} & & & \mathbf{p}_{2 \times 3} \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (3.18)$$

$$\mathbf{T}_{\mathbf{p}_2}^{\text{el}} = \mathbf{T}_4^{\text{el}} \mathbf{T}_5^4 \mathbf{T}_{\mathbf{p}_2}^5 \quad (3.19)$$

Chapter 3. Kinematic models

where $\mathbf{p}_{2 \times 3 \times 1}$ is the position vector of the forearm from the elbow to the wrist. For the wrist joint, the calculation is:

$$\mathbf{T}_6^w = \begin{bmatrix} \cos q_6 & -\sin q_6 & 0 & 0 \\ \sin q_6 & \cos q_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.20)$$

$$\mathbf{T}_7^6 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos q_7 & -\sin q_7 & 0 \\ 0 & \sin q_7 & \cos q_7 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{p_3}^7 = \left[\begin{array}{ccc|c} & & & \\ & \mathbf{I}_{3 \times 3} & & \mathbf{p}_{3 \times 1} \\ & & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (3.21)$$

$$\mathbf{T}_{p_3}^w = \mathbf{T}_6^w \mathbf{T}_7^6 \mathbf{T}_{p_3}^7 \quad (3.22)$$

where $\mathbf{p}_{3 \times 3 \times 1}$ is the position vector of the segment that connect the wrist with the hand center of gravity.

To calculate the forward kinematic model for all the arm system:

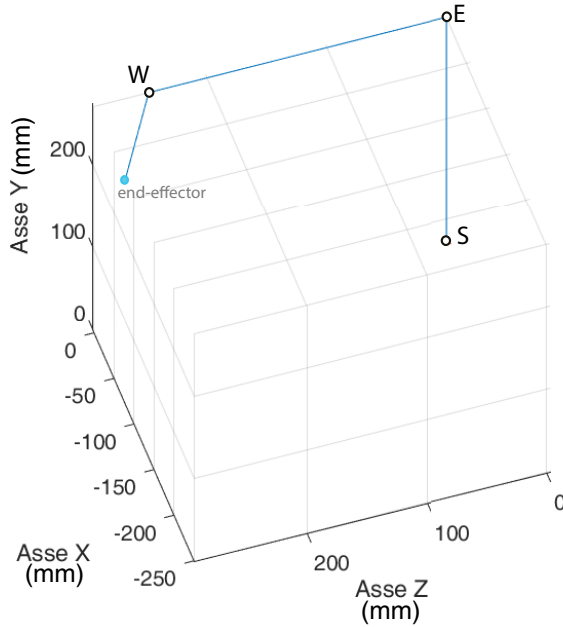
$$\mathbf{T}_w^s = \mathbf{T}_{p_1}^s \mathbf{T}_{p_2}^{el} \mathbf{T}_{p_3}^w = \left[\begin{array}{ccc|c} & & & \\ & \mathbf{R}_{w_{3 \times 3}}^s & & \mathbf{p}_{w_{3 \times 1}}^s \\ & & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (3.23)$$

From 3.23 can be easily calculated the Cartesian position coordinates ($\mathbf{p}_{w_{3 \times 1}}^s$), and the orientation of the hand respect the shoulder:

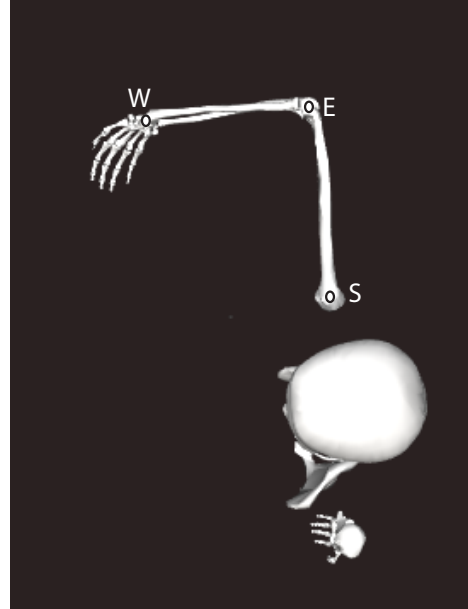
$$\begin{aligned} \beta &= \sin^{-1}(r_{32}) \\ \gamma &= \text{atan2}(-r_{31}, r_{33}) \\ \alpha &= \text{atan2}(-r_{12}, r_{22}) \end{aligned} \quad (3.24)$$

where $r_{a,b}$ corresponds to the element located in a row and b column of the matrix \mathbf{R}_w^s , whereas rotation angles refer to the following composition rule: $\mathbf{R}_w^s = R_z(\alpha)R_x(\beta)R_y(\gamma)$.

3.2. Direct and Inverse Kinematic of human arm



(a) Human arm plot in Matlab environment.



(b) Human arm representation in OpenSim environment.

Figure 3.5.: Example of human arm configuration.

In the Figure 3.5 is reported one example of upper limb configuration with a shoulder adduction of 90° , an elbow flexion of 90° and a wrist deviation of 30° . All the other joints are set to zero.

The inverse kinematics problem, instead, is focused on the joint variable calculation related to a given end-effector pose during a fixed period of time. Thanks to this algorithm is easily possible to switch from the end-effector space (i.e. the operational space) to the joint space motions that lets the accomplishment of the desired path. The inverse kinematics of human arm is a complicated problem for the high number of DoF of the limb. Since the given position and orientation of the hand specify six independent quantities, rather than seven, the human arm is a redundant system and, consequently, there is non-unique solution for the joint configuration [37,38]. For this reason is necessary an optimal selection among the possible solutions that belong to the manipulator workspace. Several inverse kinematics algorithms for human arm have been developed, but the one proposed in this study is the pseudoinverse Jacobian iteration procedure. Since the Jacobian is one of the most important tools for manipulator characterization, it is useful for finding singularities, analyzing redundancy and determining inverse kinematic algorithms. The

Jacobian utilized is the analytical one. From 3.23 the pose of the end-effector depends on the vector of the joint variables and is possible to express its linear velocity ($\dot{\mathbf{p}}_e$) and angular velocity (ω_e) as a function of the joint velocities ($\dot{\mathbf{q}}$). They can be expressed as:

$$\begin{aligned}\dot{\mathbf{p}}_e &= \mathbf{J}_p(\mathbf{q}) \dot{\mathbf{q}} \\ \omega_e &= \mathbf{J}_o(\mathbf{q}) \dot{\mathbf{q}}\end{aligned}\tag{3.25}$$

In a compact form, the 3.25 can be written as

$$\mathbf{v}_e = \begin{bmatrix} \dot{\mathbf{p}}_e \\ \omega_e \end{bmatrix} = \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}}\tag{3.26}$$

representing the manipulator differential kinematics equation. The matrix \mathbf{J} is the Jacobian function of the joint variable with $(6 \times n)$ dimension, where n is the number of joints ($n = 7$ for human arm joints). The analytical form is given by

$$\mathbf{J}(q) = \begin{bmatrix} \frac{\partial \mathbf{K}}{\partial \mathbf{q}_1} & \frac{\partial \mathbf{K}}{\partial \mathbf{q}_2} & \dots & \frac{\partial \mathbf{K}}{\partial \mathbf{q}_n} \end{bmatrix}\tag{3.27}$$

where \mathbf{K} is the expression of the direct kinematics based on the end-effector position and orientation. Therefore, to calculate the analytical Jacobian are necessary partial derivatives of the coordinates in the task space. The task is to determine a feasible joint trajectory ($\mathbf{q}(t), \dot{\mathbf{q}}(t)$) that reproduces the given trajectory. Considering the 3.26, the inverse kinematics can be obtained by simple inversion of the Jacobian matrix

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q}) \mathbf{v}_e\tag{3.28}$$

If the initial manipulator posture $\mathbf{q}(0)$ is known, is possible to integrate the velocity over time to compute joint position. The integration can be performed in discrete time by different numerical techniques. The simplest one is based on the Euler integration method, in which an integration interval is given Δt , the joint positions and velocities at time t_k are known and the joint positions at time $t_{k+1} = t_k + \Delta t$ can be computed as

$$\mathbf{q}(t_{k+1}) = \mathbf{q}(t_k) + \dot{\mathbf{q}}(t_k) \Delta t\tag{3.29}$$

To solve the 3.28 equation is necessary that the Jacobian is square and full rank. Since the human arm is redundant, the Jacobian matrix has more columns than rows and infinite solutions exists to 3.26. A solution is to formulate the problem as a constrained linear optimization problem. Indeed, it is desired to find the solutions $\dot{\mathbf{q}}$ that satisfy the linear

3.2. Direct and Inverse Kinematic of human arm

equation in 3.26 and minimize the quadratic cost functional of joint velocities

$$g(\dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{W} \dot{\mathbf{q}} \quad (3.30)$$

where \mathbf{W} is a suitable $(n \times n)$ symmetric positive definite weighting matrix, given by $\dot{\mathbf{R}}\mathbf{R}^T$. To solve this, the method of Lagrange multipliers have been adopted, considering the modified cost functional

$$g(\dot{\mathbf{q}}, \lambda) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{W} \dot{\mathbf{q}} + \lambda^T (\mathbf{v}_e - \mathbf{J} \dot{\mathbf{q}}) \quad (3.31)$$

where λ is an $(r \times 1)$ vector of unknown multipliers that allows the incorporation of the constraint 3.26 in the functional to minimize. The request solution has to satisfy the necessary following conditions:

$$\begin{aligned} \left(\frac{\partial g}{\partial \dot{\mathbf{q}}} \right)^T &= 0 \\ \left(\frac{\partial g}{\partial \lambda} \right)^T &= 0 \end{aligned} \quad (3.32)$$

From the first one, the 3.31 becomes

$$\mathbf{W} \dot{\mathbf{q}} - \mathbf{J}^T \lambda = 0 \quad (3.33)$$

and thus

$$\dot{\mathbf{q}} = \mathbf{W}^{-1} \mathbf{J}^T \lambda \quad (3.34)$$

where the inverse of \mathbf{W} exists. The solution 3.34 is a minimum, since $\frac{\partial^2 g}{\partial \dot{\mathbf{q}}^2} = \mathbf{W}$ is positive definite. From the second condition above, the constraint

$$\mathbf{v}_e = \mathbf{J} \dot{\mathbf{q}} \quad (3.35)$$

is recovered. Combining the two conditions gives

$$\mathbf{v}_e = \mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T \lambda \quad (3.36)$$

under the assumption that \mathbf{J} has full rank, $\mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T$ is a square matrix and invertible. Solving for λ it is

$$\lambda = \left(\mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T \right)^{-1} \mathbf{v}_e \quad (3.37)$$

which substituted in 3.34 gives the optimal solution

$$\dot{\mathbf{q}} = \mathbf{W}^{-1} \mathbf{J}^T (\mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T)^{-1} \mathbf{v}_e \quad (3.38)$$

where $\mathbf{W}^{-1} \mathbf{J}^T (\mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T)^{-1}$ is defined as the weighted pseudoinverse Jacobian (\mathbf{J}_w^\dagger). A particular case, instead, occurs when the weighting matrix \mathbf{W} is equal to the identity matrix \mathbf{I} and the solution simplifies into

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger \mathbf{v}_e \quad (3.39)$$

where the matrix is

$$\mathbf{J}^\dagger = \mathbf{J}^T (\mathbf{J} \mathbf{J}^T)^{-1} \quad (3.40)$$

and this solution locally minimizes the norm of joint velocities. For redundant manipulators, instead, the solution is given by

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger \mathbf{v}_e + (\mathbf{I}_n - \mathbf{J}^\dagger \mathbf{J}) \dot{\mathbf{q}}_0 \quad (3.41)$$

where the first term is related to the minimum norm joint velocities and the second one, i.e. homogeneous solution, satisfy the additional constraint to specify $\dot{\mathbf{q}}_0$.

Following the 3.41 has been performed first \mathbf{J}^\dagger (3.40) with the derivative of the direct kinematics. Then, the weighted pseudoinverse Jacobian matrix (\mathbf{J}_w^\dagger) has been calculated. In particular, the weighted matrix (\mathbf{W}) has on the diagonal the weights related to each joint, designed so that it constraints movement of certain joints with their own limits. The weighted matrix takes the form defined by Matlab function:

$$\mathbf{W} = \text{diag}(w_i) \quad (3.42)$$

where the diagonal elements (w_i) are fixed as:

$$w_i = 1 + k \left| \frac{q_i - q_{imid}}{q_{imax} - q_{imin}} \right| \quad (3.43)$$

where q_{imid} is the midpoint of the joint i , delimited in the range $[q_{imin}, q_{imax}]$ and k is a scalar value used to give weight to the absolute value.

The 3.43 ensure that movements are amplified in the joints in between closer to the middle of joint limits. Since the weighted pseudoinverse method has incorporated joint limits, some selected joints can move less than the others when the arm is moving. Indeed, when the value of the element (j, j) in \mathbf{W} is large compared with the other elements, the

3.3. Direct and Inverse Kinematic of robotic arm

incrementation of the joint j is small. On the contrary, if it is small the incrementation is large. \mathbf{W} is a dynamic matrix that in each instant of time changes its values. The inverse kinematics of human arm computed with \mathbf{J}^\dagger and $\mathbf{J}_{\dagger_w}^\dagger$ has reported different value for the joint position configuration, as shown in Figure 3.6. Going more into detail, the difference between the joints (without weight and weighted) is also reported in the Figure 3.7, where some joints as q_3 and q_5 differ significantly with the respective weighted. It means that they are really close to each other with a divergence easily approximated to zero.

3.3. Direct and Inverse Kinematic of robotic arm

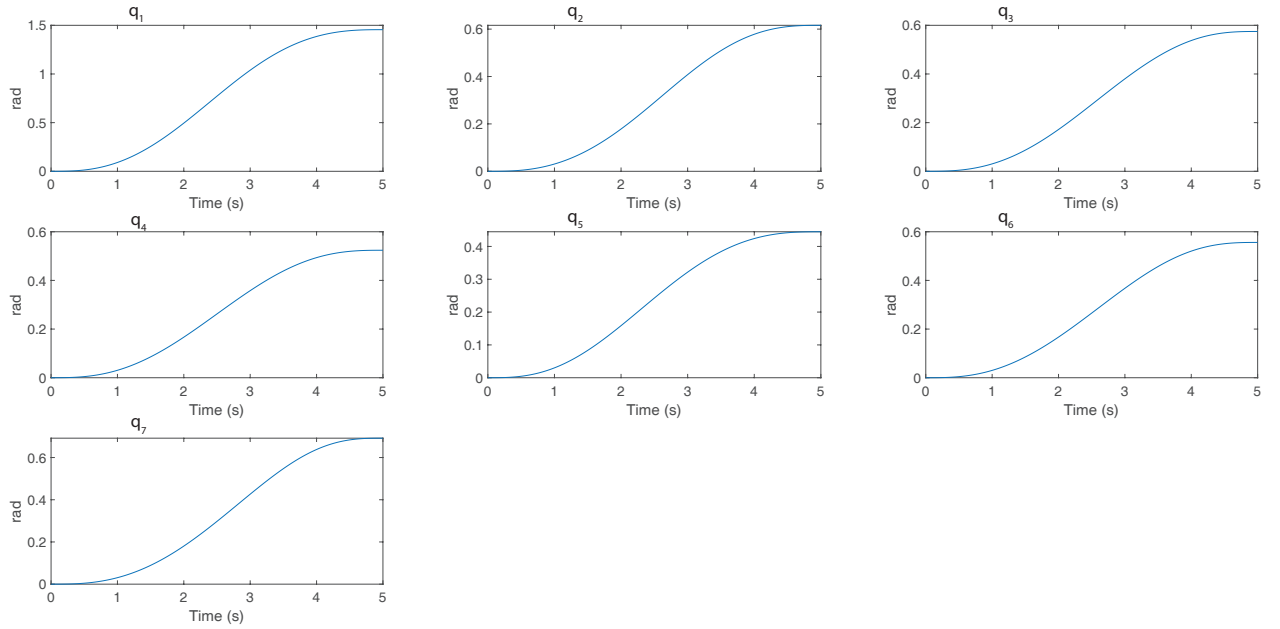
The robot taken into account is the UR5 of Universal Robot company and its mathematical model for forward and inverse kinematic is presented. The UR5 robot has six DoF with six revolute joints: the base joint (q_1), the shoulder joint (q_2), the elbow joint (q_3) and three wrist joints (q_4, q_5, q_6) (Figure 3.8, 3.9). These last three do not act as a coincidental wrist. Therefore, all the six joints contribute to the transformational and rotational movements of its end-effector [39]. This characteristic makes the kinematics analysis more complex in comparison with other manipulators having coincidental wrist. In the Figure 3.10 are shown the lengths of the robotic links. Since this arm has 6 DoF, the system is not redundant. Therefore, the arm manipulability is not maximum and the singularities can occur more [40]. The UR5 is equipped with a high-level controller that can easily control the robot both in joint and Cartesian space. A desired trajectory describes the pose of the end-effector in the Cartesian space and, in the meantime, needs to be mapped from the task space to a reference trajectory in the joint space, where the actuators provide their input.

In this study the Denavit-Hartenberg (DH) parameters are used to calculate the forward kinematics [40, 41]. The parameters are reported in Table 3.3 with the corresponding coordinate frames illustrated in Figure 3.11. The direct kinematic, as for human arm, is given by 3.12 and the transformation matrix from the base to the end-effector is expressed by:

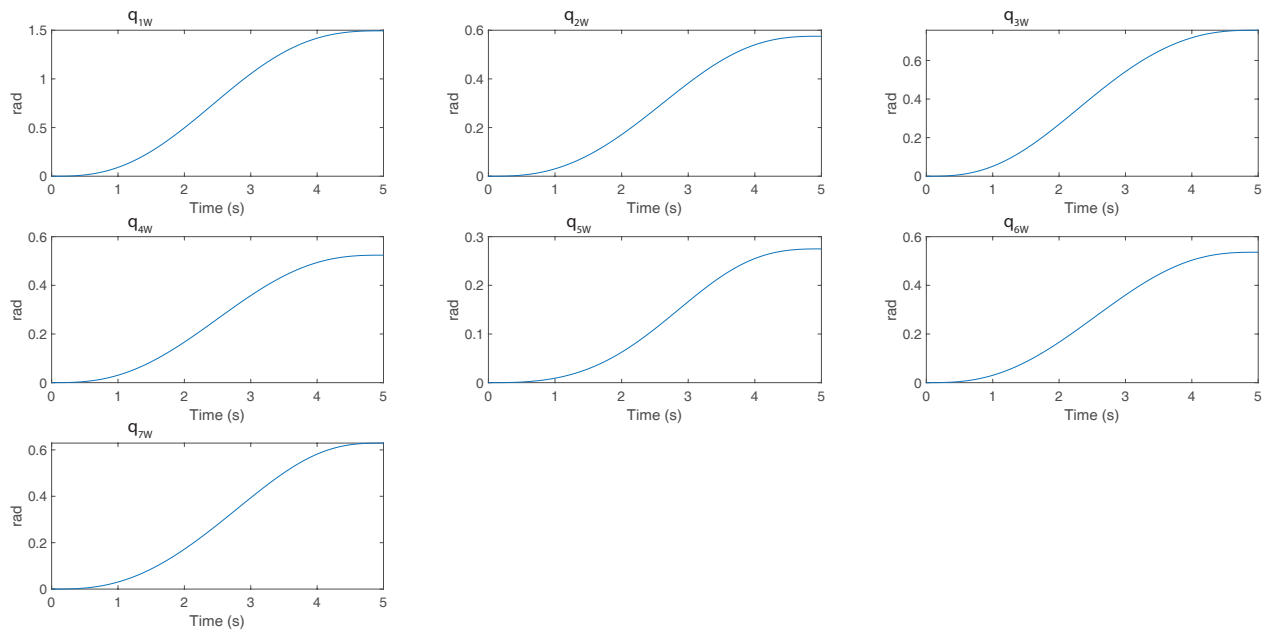
$$\mathbf{T}_e^0(q_1, q_2, q_3, q_4, q_5, q_6) = \mathbf{T}_1^0(q_1) \mathbf{T}_2^1(q_2) \mathbf{T}_3^2(q_3) \mathbf{T}_4^3(q_4) \mathbf{T}_5^4(q_5) \mathbf{T}_6^5(q_6) = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.44)$$

in which the robot position is easily obtained from the fourth column of the \mathbf{T}_e^0 . The DH-parameters are used to write the transformations for each link. The general analytical

Chapter 3. Kinematic models



(a) no-weighted joints.



(b) weighted joints

Figure 3.6.: Example of joint position configuration following a fixed trajectory.

3.3. Direct and Inverse Kinematic of robotic arm

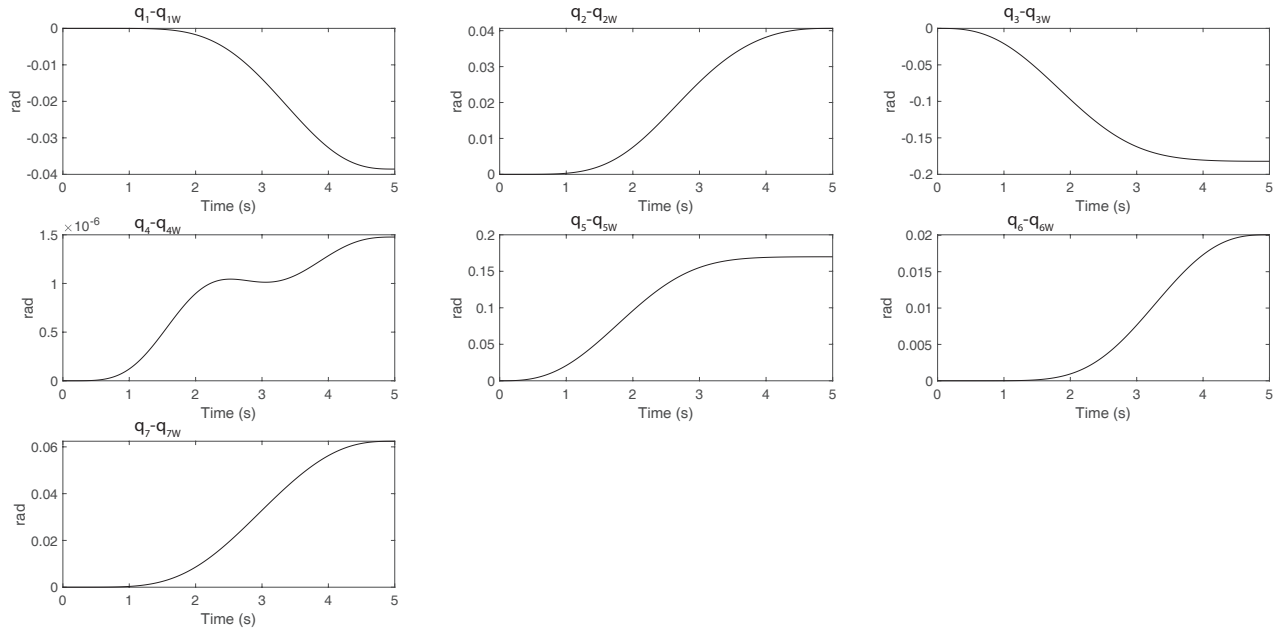


Figure 3.7.: Difference between no-weighted and weighted joints.

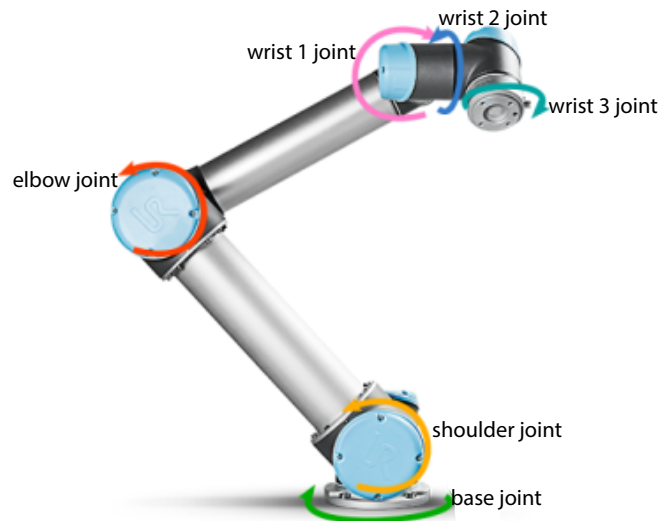


Figure 3.8.: Revolute joints of eUR5.

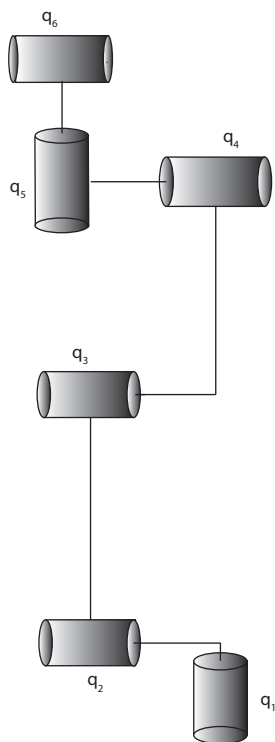


Figure 3.9.: Kinematics of UR5.

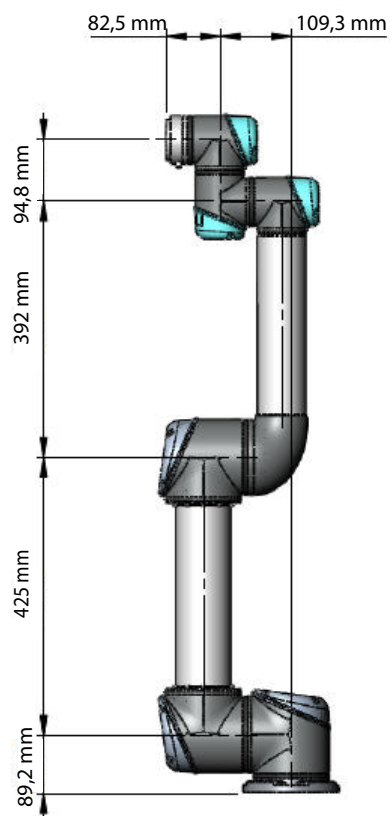


Figure 3.10.: UR5 dimensions.

Table 3.3.: DH parameters of UR5

joint	a_i (mm)	α_i (rad)	d_i (mm)	θ_i (rad)
0	0	0	-	-
1	0	$\frac{\pi}{2}$	82,2	q_1
2	425	0	0	q_2
3	392	0	0	q_3
4	0	$\frac{\pi}{2}$	109,3	q_4
5	0	$-\frac{\pi}{2}$	94,8	q_5
6	-	-	82,5	q_6

3.3. Direct and Inverse Kinematic of robotic arm

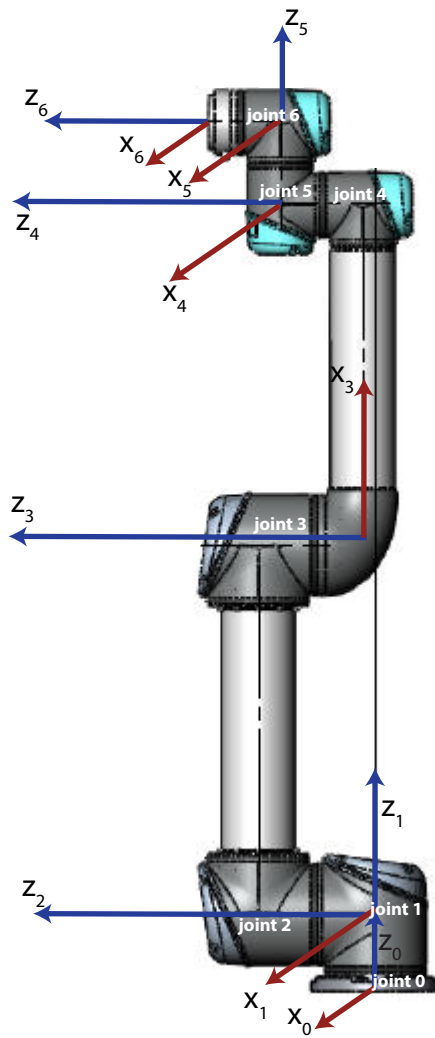


Figure 3.11.: DH parameters in the robot vertical position.

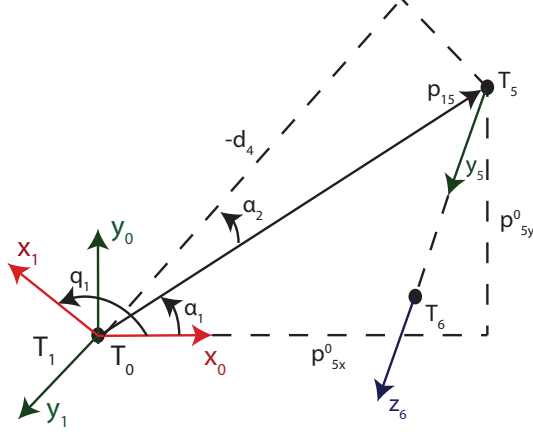


Figure 3.12.: Geometry scheme for calculating q_1 .

transformation between link $i - 1$ and i is given by:

$$\mathbf{T}_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \sin(\theta_i)\cos(\alpha_{i-1}) & \cos(\theta_i)\cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & -\sin(\alpha_{i-1})d_i \\ \sin(\theta_i)\sin(\alpha_{i-1}) & \cos(\theta_i)\sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & \cos(\alpha_{i-1})d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.45)$$

This transformation is applied to each joint, following the matrix multiplications 3.44. The inverse kinematic algorithm, instead, is focused on finding the set joint configurations $\mathbf{Q} = [q_1, q_2, q_3, q_4, q_5, q_6] \in [0, 2\pi]^6$ knowing the pose of the end-effector. The executed procedure follows the subsequent steps just below [39, 41]. Since the distance between the frame 6 and the frame 5 is known (d_6), the first joint calculated is q_1 . Considering the wrist, the rotation from frame 0-to-1 (q_1), should be equal to the difference between the rotations from 0-to-5 and 1-to-5, as shown in Figure 3.12.

$$q_1 = \alpha_1 + (\alpha_2 + \frac{\pi}{2}) = \text{atan2}(p_{5y}^0, p_{5x}^0) \pm \cos^{-1}\left(\frac{d_4}{\sqrt{(p_{5y}^0)^2 + (p_{5x}^0)^2}}\right) + \frac{\pi}{2} \quad (3.46)$$

where two possible solutions (left or right) for the shoulder joint are admissible. Since the y component of \mathbf{p}_6^0 depends only on q_5 (Figure 3.13), the wrist joint can be computed by:

$$q_5 = \pm \cos\left(\frac{p_{6x}^0 \sin(q_1) - p_{6y}^0 \cos(q_1) - d_4}{d_6}\right) - 1 \quad (3.47)$$

3.3. Direct and Inverse Kinematic of robotic arm

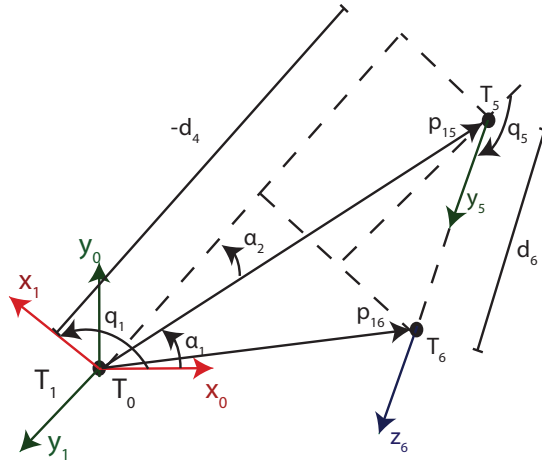


Figure 3.13.: Geometry scheme for calculating q_5 .

in which there are two possible solutions, corresponding to the wrist up or down, respectively. In fact, the joint sum $(q_2 + q_3 + q_4)$ can cause the end-effector to be located in the same position, but with the wrist flipped. Then, the orientation can be adjusted by q_6 . A solution is defined as long as the module of the argument of the cosine is less than 1. To determine q_6 is better express spherical coordinate for the vector \mathbf{y}_1^6 where q_6 is the azimuthal angle and q_5 is the polar angle (Figure 3.14). The x and y coordinates of this vector form a system which can be easily solved as:

$$q_6 = \text{atan2}\left(\frac{y_y \cos(q_1) - y_x \sin(q_1)}{\sin(q_5)}, \frac{x_x \cos(q_1) - x_y \sin(q_1)}{\sin(q_5)}\right) \quad (3.48)$$

When $\sin(q_5) = 0$ and the joint axes 2,3,4 and 6 are aligned, the 3.48 is undetermined. When this occurs, a desired arbitrary value of q_6 can be supplied to fully determine the system. If both the numerators of 3.48 are equal to zero, the solution is also undetermined. For the remaining three joints (2,3 and 4) the procedure to find the joint angles is the same because of they constitute a planar 3R-manipulator (Figure3.15). The joints are given by the following equations:

$$q_3 = \pm \cos\left(\frac{|p_{4xz}^1|^2 - a_2^2 - a_3^2}{2a_2a_3}\right)^{-1} \quad (3.49)$$

$$q_2 = \alpha_1 - \alpha_2 = \text{atan2}(-p_{4z}^1, -p_{4x}^1) - \sin\left(\frac{-a_3 \sin(q_3)}{|p_{4xz}^1|}\right)^{-1} \quad (3.50)$$

$$q_4 = \text{atan2}(x_{4y}^3, x_{4x}^3) \quad (3.51)$$

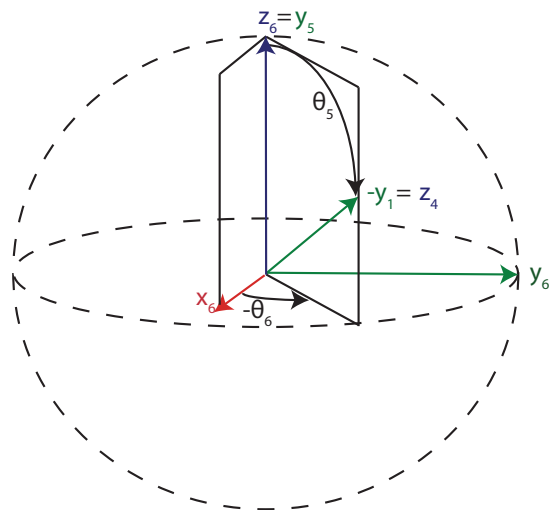


Figure 3.14.: Geometry for q_6 .

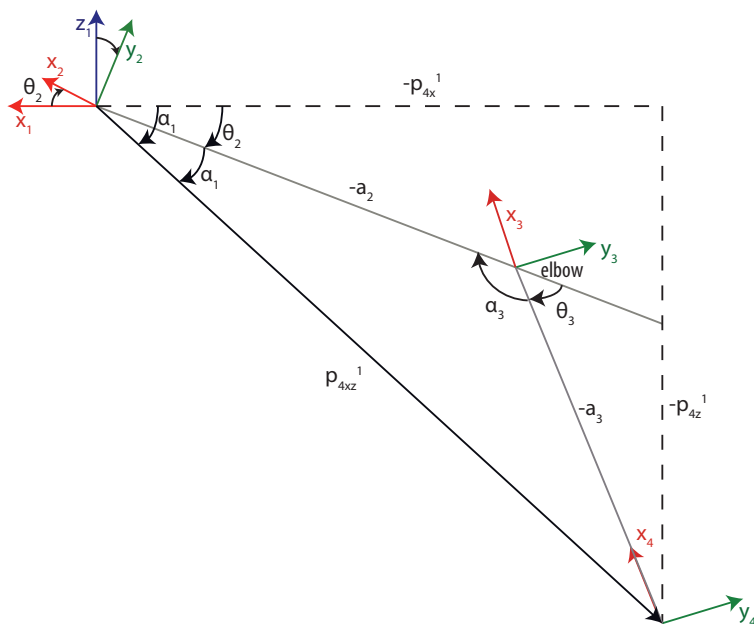
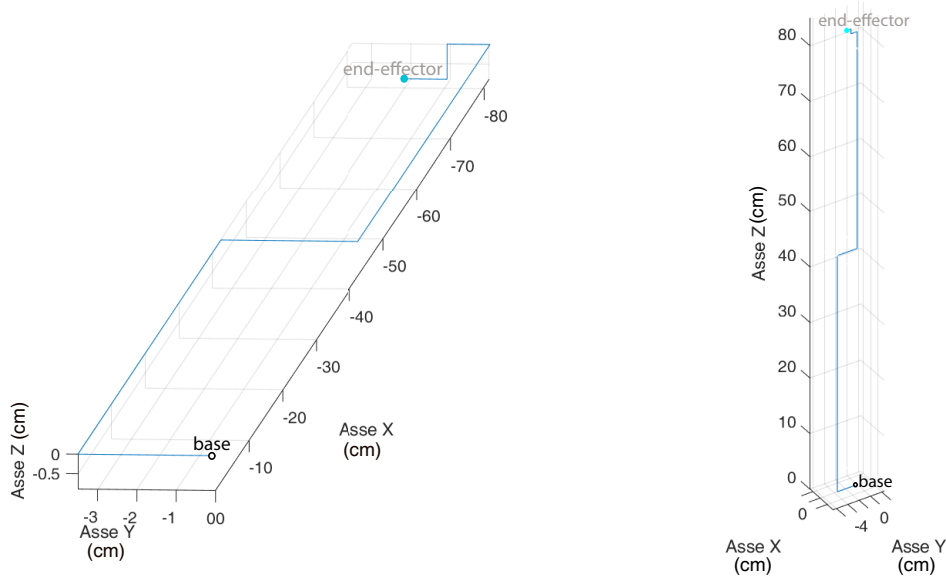


Figure 3.15.: Geometry for q_4 .

3.3. Direct and Inverse Kinematic of robotic arm



(a) Initial robot joint position, where from q_1 to q_6 are zero values. (b) Final robot joint position, where $q_1 = 0, q_3 = 0, q_5 = 0, q_6 = 0$ and $q_2 = 90^\circ, q_4 = 90^\circ$.

Figure 3.16.: Example of the initial and final configuration of UR5.

Also for these equations the possible configurations are two, where the arm is elbow up or down. No solutions exist when the distance to the 4th joint exceeds the sum $|a_2 + a_3|$ or is less than the difference $|a_2 - a_3|$. If $a_2 = a_3$, a displacement singularity exists when $q_3 = \pi$, making q_2 arbitrary.

In this study the algorithm used for inverse kinematics of UR5 is the one proposed by Michael Kutzer based on DH parameters. It establishes 2^9 solutions and the ones admissible are those who contain the minimum norm. In the Figure3.16 is reported the pose of the robot end-effector in initial and a final configuration.

Chapter 4.

Simulations

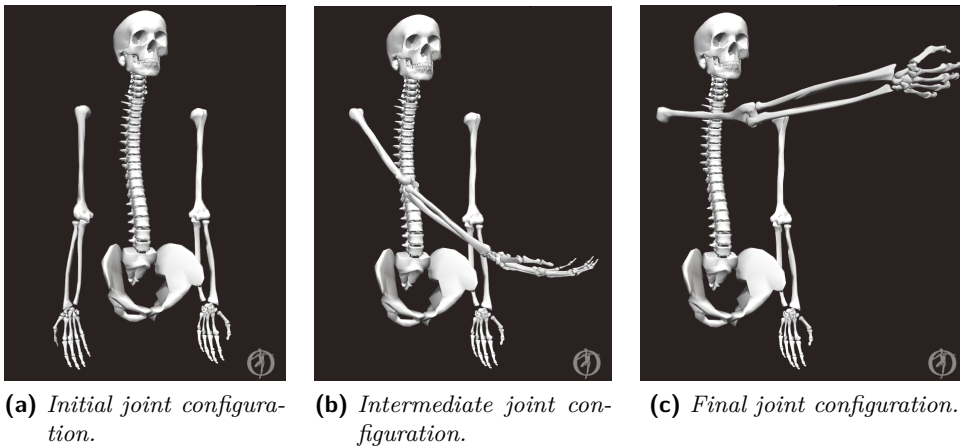
In this chapter are reported the software utilized for the simulations of human-robot movements. Indeed, the first section describes OpenSim software, which includes tools and graphical user interface to model human system. Then, is reported the Simulink-Adams co-simulation to design a control system and a virtual prototype model for human and robot arm. The results obtained are located in the last section.

4.1. OpenSim software

OpenSim is an open-source software system used to model, simulate and analyze the neuro-musculoskeletal human system and create dynamic simulations of a wide variety of human movements. Moreover, OpenSim can perform numerical integrations and solve constrained non-linear optimization problems. In this way, for solving the inverse kinematics, the software resolves internal coordinates from available spatial marker positions corresponding to known landmarks on rigid segments. It is also possible to determine the set of generalized forces and to generate trajectories of states by integrating system dynamical equations [42]. In the OpenSim graphical user interface, the user is able to access a suite of high-level tools for viewing models, editing muscles, plotting results and other functions. The software is written in ANSI C++, while the graphical user interface is written in Java, allowing OpenSim to compile and run on common operating systems [5]. Human body models and simulations are generally used to generate muscle dynamics and musculoskeletal geometry. In particular, simulations allow to establish cause-and-effect relationships giving insights into muscle function [43]. In every models are reported the admissible joint range for each articulation of each limb. For this study, OpenSim software (version 4.0) has been applied to confirm the human arm movement previously established in Matlab[®] environment. The human model utilized is the Full Body Running Model developed by Samuel Hamner. It is simplified in order to obtain only a simple human arm model without the representations of muscles. These graphics-based musculoskeletal tools

Table 4.1.: Initial and final configuration of human joints.

joint	initial value (°)	final value (°)
q_1	0	90
q_2	0	30
q_3	0	60
q_4	0	30
q_5	0	0
q_6	0	30
q_7	0	30

**Figure 4.1.:** Example of one right arm motion.

are mostly adopted for visualizing human movement and analyzing the functional capability of muscles. Sometimes can be used also to design improved surgical procedures. The movement of human arm can be visible in OpenSim software through a motion file (.mot) that contains the movement of each joint in fixed period of time. The algorithm was developed for the right human arm, but can be easily adoptable for the left part. Considering the initial and final condition of each joint reported in Table 4.1, in Figure 4.1 is shown the time excursion of the right arm, based on the previous planning motion trajectory (3.1).

4.2. Adams models

ADAMS[®] (*Automatic Dynamic Analysis of Mechanical Systems*) is a virtual prototyping and motion simulation software used mostly in academic research. It allows to model a mechanical system, mathematically simulate and visualize 3D motion system and its force behaviour. This software is able to automatically converts a graphically defined model to dynamic equations of motion, and then solves equations typically in time domain. It is also able to resolve redundant constraints, handle unlimited degrees of freedom, and perform static equilibrium, kinematic and dynamic analysis. Systems may be comprised of any number of rigid and flexible bodies and can be subjected to any variety of internal or external forces. It is possible to define a number of constrains such as joints, joints primitive, time-dependent motions, higher-pair contacts and user-written [44].

In this study are built two specific mechanical systems: a phantom from the waist and UR5 robot. First, both are designed through CAD software and then, imported in ADAMS environment (version 2018.1) where are added proper constrains (Figure 4.2).

For the phantom (Figure 4.2a) is represented only one arm (the right one) with a simple design of the upper arm, forearm and hand as rigid bodies. The human fixed reference frame is located on the shoulder with the axis convention defined in Figure 3.4. Since the shoulder joint allows 3 rotations, because of its definition of ball-and-socket joint, it is possible to simplify into three revolute joints. In particular, the involved rotations include each axis respectively flexion-extension (about z axis), abduction-adduction (about x axis) and external-internal rotation (about y axis). The universal joints of the elbow and the wrist, instead, are designed as two revolute joints for each one. The rotations for the elbow joint are respectively flexion-extension (about z) and pronation-supination (about y). In the wrist joint, the rotations occur around z axis and x axis corresponding to the flexion-extension and abduction-adduction. In ADAMS[®] all the rotations are possible only on z direction, and, consequently, the mobile reference frames are proper located in order to obtain the motion in the correct direction. The anthropometric values of the phantom are referred to a man height 1,70 m and follows the formula reported in Table 3.1.

The UR5 model, (Figure 4.2b) is designed with its six joints (base, shoulder, elbow and three wrist joint) and the corresponding links based on company measurements. The robot fixed reference frame is located on the base joint with the DH convention. Since each revolute joint occurs about z axis, the rotational movements can be easily reported in ADAMS[®]. In order to perform motions for the rehabilitation process, it was also studied one kind of handle systems on the robot end-effector. Therefore, in this project a simple handle has been adopted for the human-robot interface. The handle is in line with the

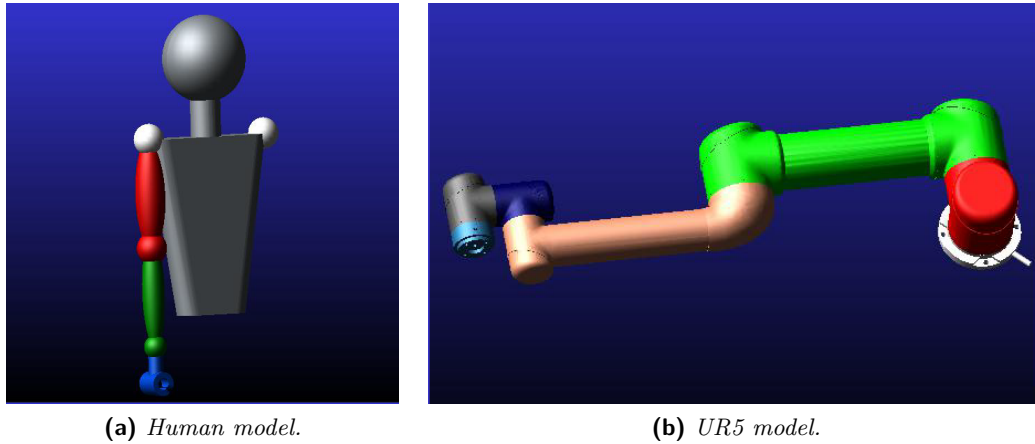
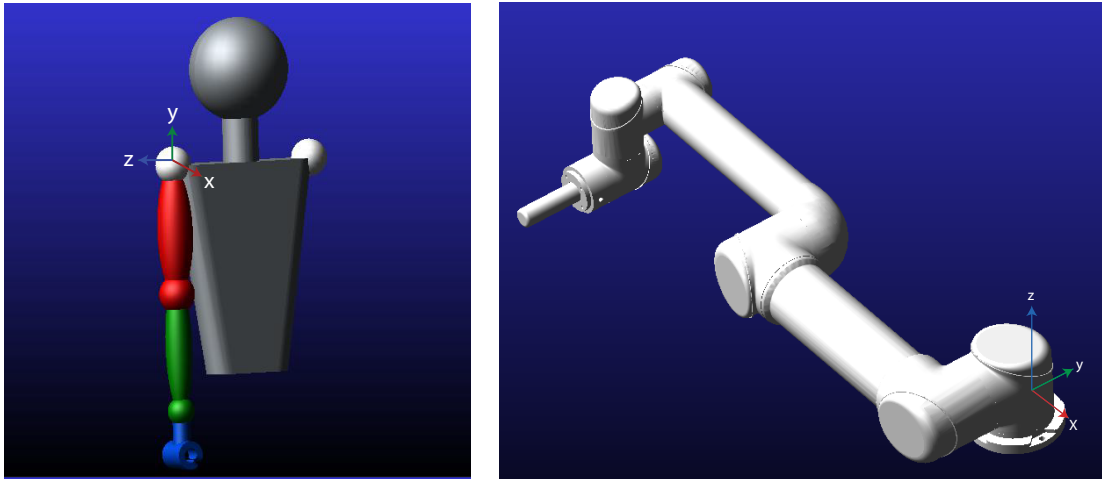


Figure 4.2.: CAD systems imported in ADAMS®.

structural parameters of the end-effector and it was presumed that the patient has the capability to grab it autonomously. The base of the robot is fixed corresponding to the center of human chest. This is due for simulating a suitable rehabilitation process with a collaborative robot located on a table. In the Figure 4.3 are reported the fixed reference frames for each model.

The starting phase of the human and the robot is shown in Figure 4.4, where the models are placed one in front of the other. In particular, the collaborative movement is obtained from the direct kinematics of human arm and inverse kinematics of robot arm. Considering these two systems together, an unique fixed reference frame has been fixed located on the human shoulder. Therefore, it was performed a roto-traslation of robot base frame respect the human one, as it will be reported in Section 4.3. In order to verify the correct functionality and the joint motion direction of the two models in ADAMS® environment, is also possible to give motions to every joint. The motion is expressed as a function time dependent, that can be a step, a sinusoidal or a polynomial function. In order to have the cooperation between the models is necessary that the two workspace intersect each others. Indeed, the hand of the phantom has to reach the handle of the robot. Since the lengths of the human and robot arms are known, the robot base is located at a distance of 700 mm from the fixed shoulder frame along x axis.



(a) Human fixed reference axis on the right shoulder.

(b) UR5 fixed reference axis on the base joint.

Figure 4.3.: Fixed reference frames in ADAMS[®] environment.

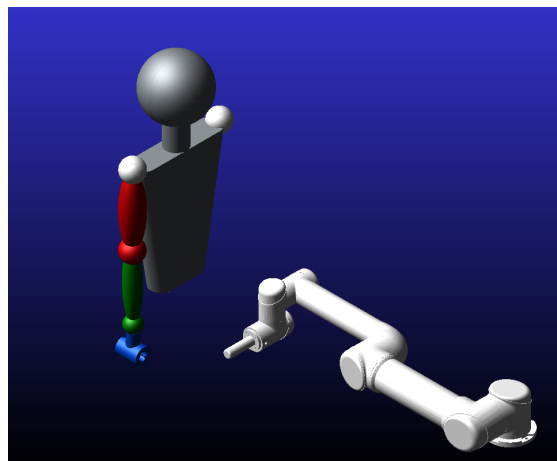


Figure 4.4.: Initial pose of the system.

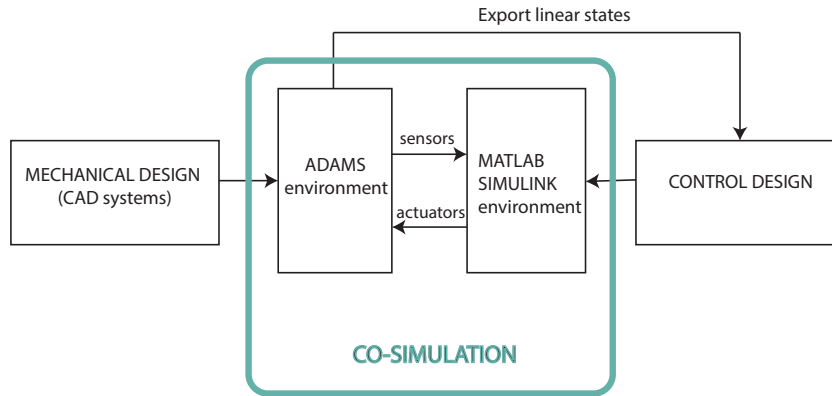


Figure 4.5.: Co-simulation of mechatronic system.

4.3. Simulink-Adams control simulation

A virtual prototype, based on the developed systems, has been performed with a co-simulation technique. In particular, the ADAMS and Simulink co-simulation is based on direct embedding of dynamic model of a mechanical system implemented in ADAMS into MATLAB environment. The aim of this co-simulation is not only to test and simulate the systems, but also to execute a control system design.

The typical procedure is:

1. to design and assemble a mechanical system in CAD software
2. to import the assembly model into ADAMS environment
3. to link the ADAMS model with Simulink environment through state variables.

In this way the whole virtual prototype is tested and a control design has been adopted [45]. In this way, the mechanical structure and control system apply the same model to design and simulate. This method can simplify the debugging process of the physical prototype testing and improve the efficiency of mechanical and electrical system design [46]. The process of co-simulation is shown in Figure 4.5. Indeed, a mechatronic approach is based on the integration of knowledge from different areas of physics and technical disciplines achieving the synergic effect. The mechatronic approach solves the development process as only one task, while the development task like mechanics, electronics, control system (including software) are processed together. The co-simulation discussed in this study is to import the direct kinematic of human arm and the inverse kinematic of robot arm from

4.3. Simulink-Adams control simulation

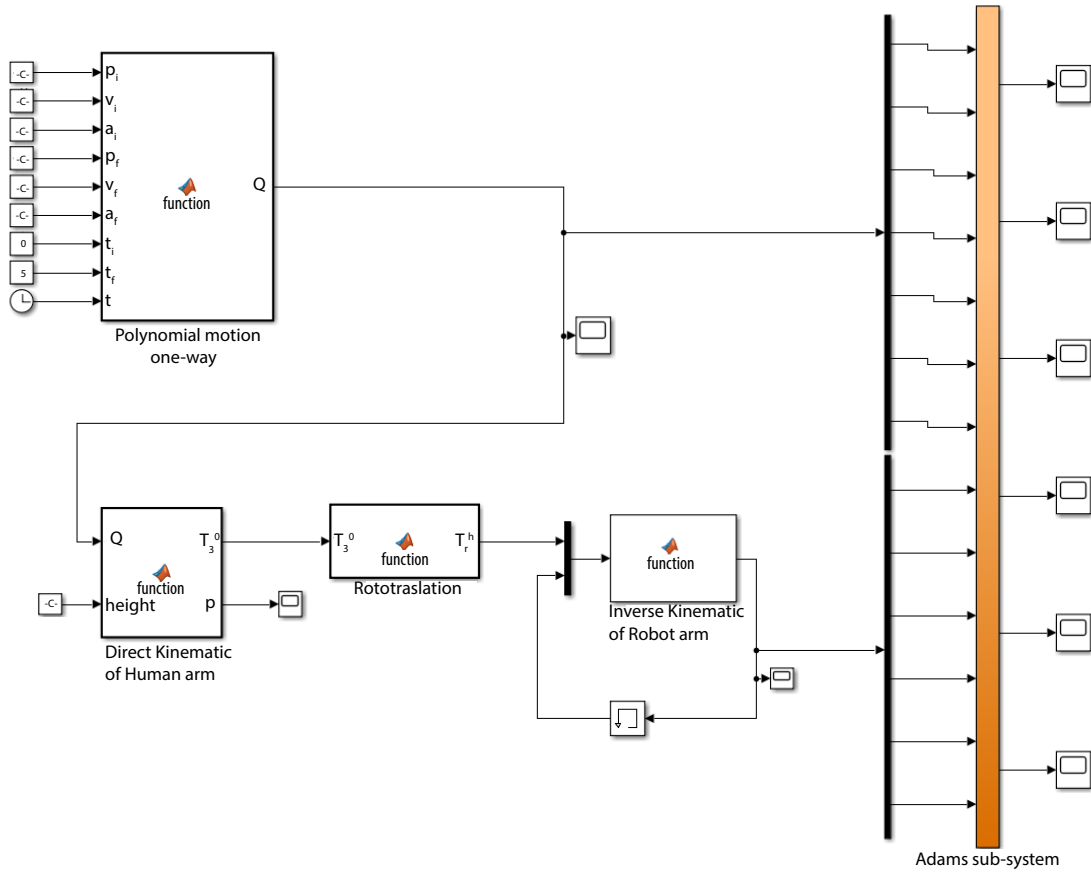


Figure 4.6.: Simulink-ADAMS control scheme.

MATLAB to Simulink environment. Meanwhile, there is the combination of the mechanical structure of the ADAMS system through the interface of ADAMS-control module with Simulink space.

In ADAMS software, linear state variables are exported, considering the outputs as the end-effector position of human and robot arms and as inputs all human and robot joints. It means that the ADAMS model will have 13 inputs and 6 outputs. In this way, an ADAMS file will be created and it can be open in Matlab-Simulink environment (the ADAMS sub-system). In the Figure 4.6 is shown the basic control scheme adopted in this study. Each block component of this control system will be accurately analyzed.

The first block is the so called "Polynomial motion one-way" which contains the motion planning previously implemented (Section 3.1). It is a Matlab function that can be easily reported on Simulink environment and on its left side are shown the inputs p_i , v_i , a_i ,

Chapter 4. Simulations

$p_f, v_f, a_f, t_i, t_f, t$ respectively corresponding to initial position, initial velocity, initial acceleration, final position, final velocity, final acceleration, initial time, final time and the time instant. From p_i to a_f are row vectors of (1×7) dimension, while the initial and final time values are respectively only a value. On the contrary, the last input is not fixed, but is a clock that follows the current simulation time. The value $-C-$ is a short way through which Simulink expresses constant values. The output of the function is the trajectory that each joint has to follow. This motion block is for used for the production of only one-way trajectory. Later will be examined other kinds of motions.

The second block examined is the "Direct Kinematic of Human arm" implemented in Section 3.2. This too is a Matlab function imported in Simulink space. The inputs are the motion planning (Q) and a constant value for human height. The outputs of this are \mathbf{T}_3^0 , which corresponds to the homogeneous transformation matrix of the human end-effector (hand frame) to the base zero frame (shoulder frame) and \mathbf{p} is the pose of the hand.

The third block is the "Rototraslation" function, which is able to perform a translation and a rotation of the robot base frame respect to the shoulder frame. In fact, the input is the previous homogeneous transformation matrix \mathbf{T}_3^0 and the output is a new homogeneous transformation matrix that expresses the robot base frame respect to the shoulder fixed frame. The rotation is 90° about x axis in anticlockwise direction.

The fourth function block is the "Inverse Kinematic of Robot arm" in which the inverse kinematic is performed on the basis of the Section 3.3. The input is the concatenation of the pose of the human hand and the precedent values of robot joint angles calculated in each instant of time. In this way, a feedback loop is created for the purpose of obtaining a robot movement closer as possible to the previous one in order to avoid singularities.

On the right side, there is the "Adams sub-system" block that is generated after the exportation of Adams linear state variables. It has 13 inputs corresponding to the 7 human joints and 6 robot ones. The outputs of that block correspond to the position of human and robot end-effectors. Thanks to this block is possible to select different animation modes of the system operating. In particular, the interactive mode allows directly a graphical display of the ADAMS model in the Simulink environment, while in batch mode the control-system compiling is faster and the visualization of the model is done in ADAMS software. Attached to the Adams output there are "Scope" blocks used for plotting the joint position on time. These kinds of blocks are greatly located in the model for testing the correct joint path. The trajectories of the Scope system are represented in the next section.

The black bar are used to split or concatenate multiple input values (or one) into a precise number of outputs.

To have an accurate co-simulation is essential to set the integration time of the two soft-

4.3. Simulink-Adams control simulation

ware. Since Simulink and ADAMS operate in a different manner, a huge problem could be the estimation of the communication time, in which the two environments have to exchange informations. There is not a fixed and unique time value, but the choice depends on how both Simulink and ADAMS models are complex. The only restriction is that the ADAMS communication interval needs to be greater than the Simulink discrete integration time. In this study, the ADAMS time is set at 0.01 s and the Simulink one at 0.001 s. Meanwhile, the Simulink duration time can be chosen depending on the execution motion time. Thanks to the Simulink intuitive graphical interface and its object-oriented model representation, the building of the control-system can be easily constructed. In this project, the entire model is constructed in block diagram form with one block for each system component, including function block, which can be combine to form a dynamic model. The simulation in Simulink environment is performed also in off-line mode, while in ADAMS system only on-line analysis is available. In Figure 4.7 and 4.8 are shown the same basic control scheme with different motion planning utilized in the analysis of rehabilitation process, in which the first involves a round-trip motion and the second a trajectory that intersect three different fixed points. In the 3-points motion the inputs for the polynomial trajectory are increased because of the greater motion complexity. In particular, it means that there is another point in between the starting point and the last one that includes its final position (p_{f_B}). Each kind of motion planning has been accurately tested and suitable for the rehabilitation purpose.

The basic principle of the control-scheme is that through a simple polynomial trajectory is possible to give directly human joint motions and indirectly robot joint motions to ADAMS system. The first one is in a direct line linked to Adams subsystem, while the path for the robot joints is more complicated. This last is specifically obtained through the calculation of the inverse kinematic of robotic arm given by the known hand pose.

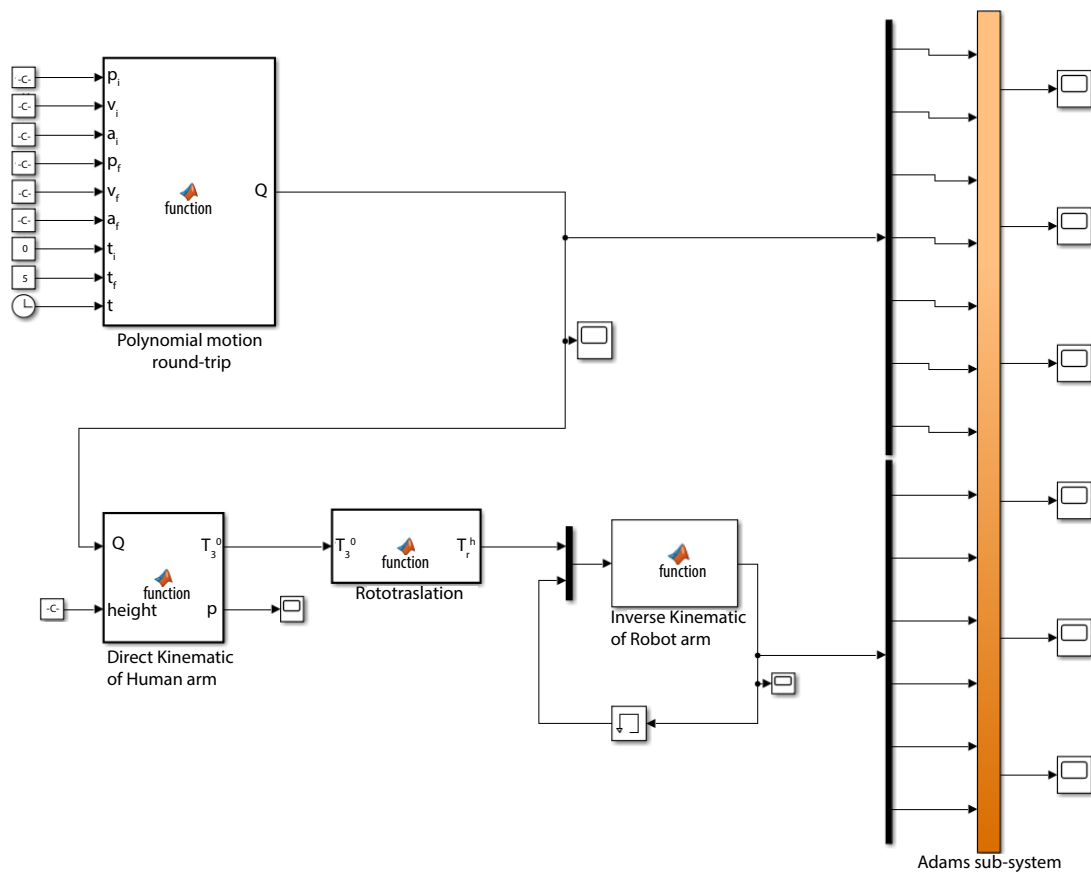


Figure 4.7.: Simulink-ADAMS control scheme with a round-trip polynomial motion.

4.3. Simulink-Adams control simulation

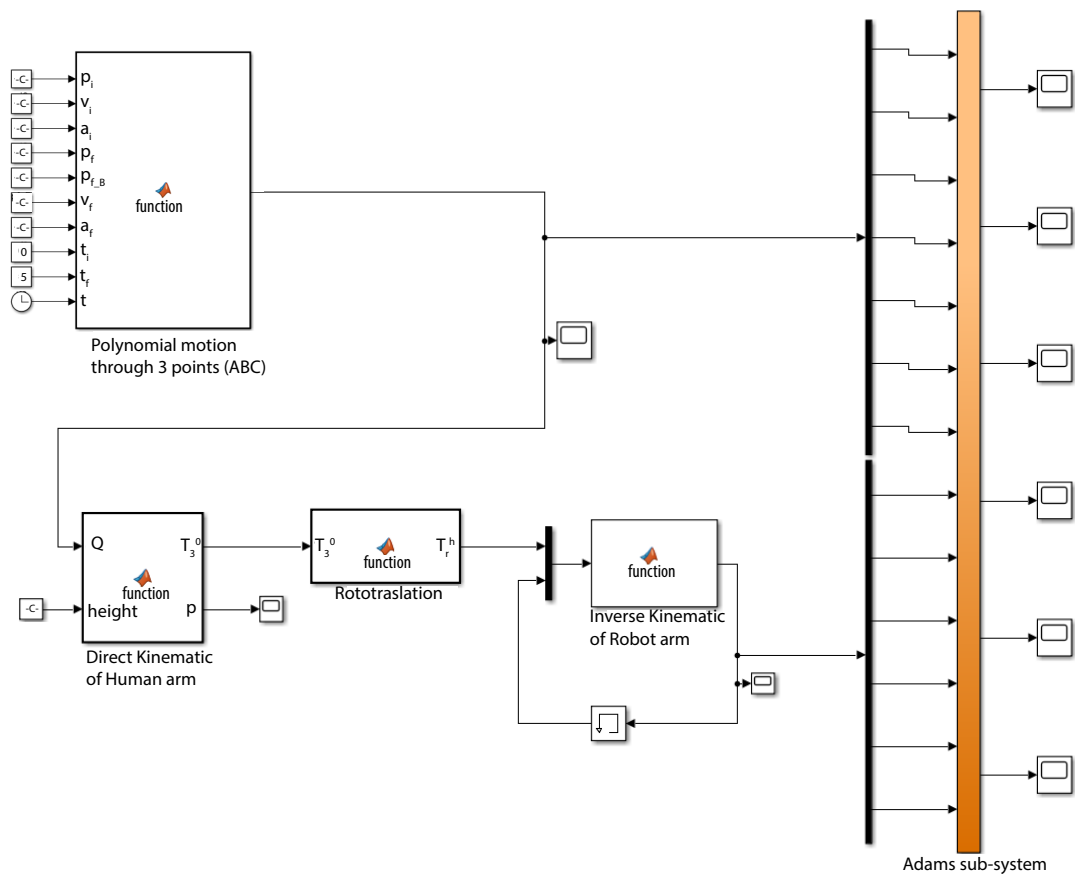


Figure 4.8.: Simulink-ADAMS control scheme with a polynomial motion through three fixed points.

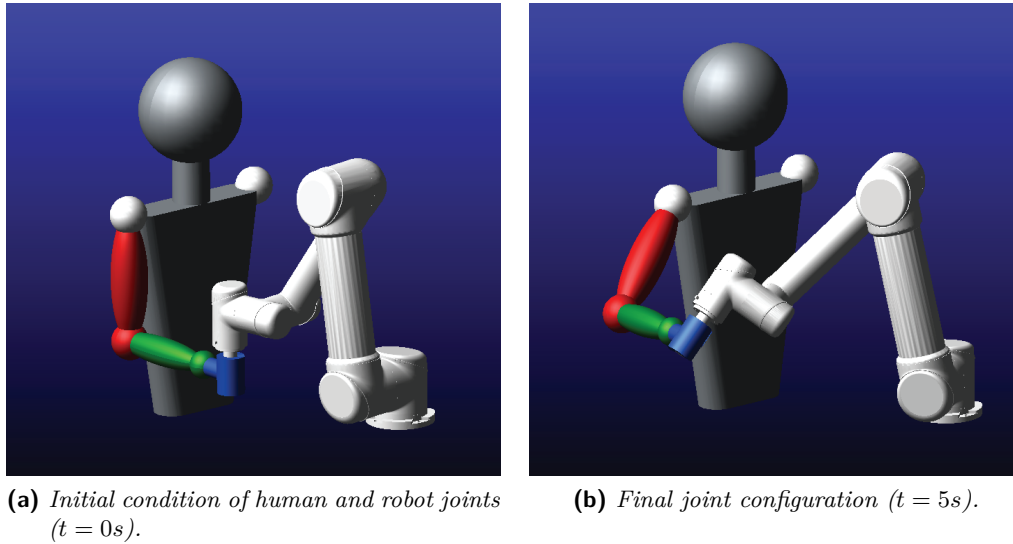


Figure 4.9.: Example of a one-way motion.

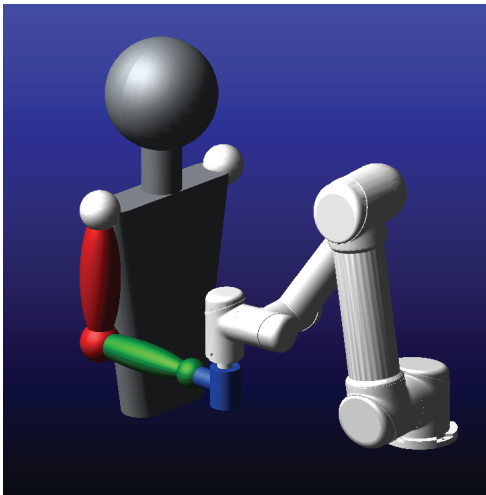
4.4. Results

In this section are reported all the results obtained in the study. First, it will be shown one example of a simple one-way motion in the task space. Then, two other examples respectively of a round-trip and 3-points motions. The time excursion of human and robot joints is also represented. The human-robot interface is an handle, which is embedded with the robot end-effector and easily grabbed by the phantom. The robot, instead, moves in a cooperation way with the human, where each upper limb motion corresponds to a dynamical movement of the machine. In the Table 4.2 are reported the initial and final human joint configuration for each motion. In all the simulations the initial human joint conditions are set at zero value, except the elbow flexion fixed at 90° . For the last motion, instead, are taken two reference final values because of the 3 fixed points. In the Figure 4.9 is reported an example of one-way motion. In the Figure 4.10 is shown a round-trip motion and in the Figure 4.11 there is an example of 3-points motion. All these visual simulations are easily obtained from ADAMS software (Adams Animation Controls) thanks to its capability of importing the last result file from Simulink environment.

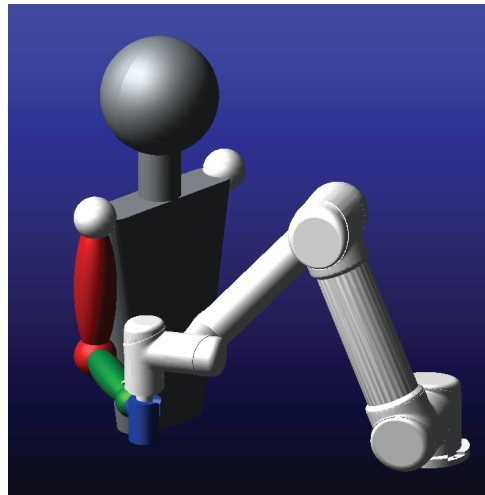
Starting from the example of one-way motion, in Figure 4.12 is shown the human joint excursion and in Figure 4.13 are reported the robot joint movement during all the performance. In Figure 4.14 and Figure 4.15, instead, is respectively illustrated the behaviour of human and robot joints during the round-trip motion. The same occurs for the 3-points

Table 4.2.: Initial and final joint values for each motion.

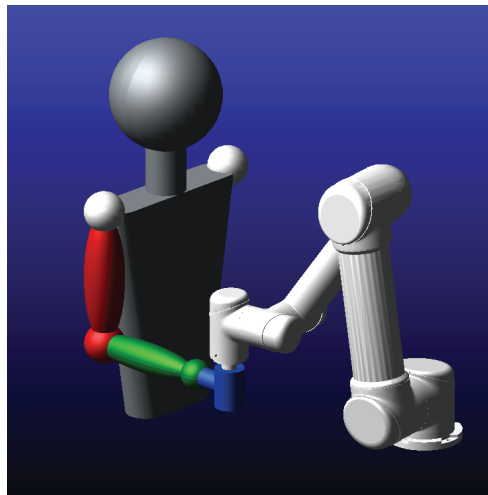
		Initial value	Final value	
one-way motion	q_1 joint	0°	20°	
	q_2 joint	0°	-45°	
	q_3 joint	0°	0°	
	q_4 joint	90°	70°	
	q_5 joint	0°	0°	
	q_6 joint	0°	0°	
	q_7 joint	0°	0°	
round-trip motion	q_1 joint	0°	0°	
	q_2 joint	0°	0°	
	q_3 joint	0°	-30°	
	q_4 joint	90°	90°	
	q_5 joint	0°	0°	
	q_6 joint	0°	0°	
	q_7 joint	0°	0°	
3-points motion	q_1 joint	0°	20°	90°
	q_2 joint	0°	-45°	0°
	q_3 joint	0°	0°	0°
	q_4 joint	90°	70°	0°
	q_5 joint	0°	-45°	0°
	q_6 joint	0°	0°	0°
	q_7 joint	0°	0°	0°



(a) Initial condition of human and robot joints ($t = 0s$).



(b) Intermediate joint configuration ($t = 5s$).



(c) Final joint configuration ($t = 10s$).

Figure 4.10.: Example of a round-trip motion.

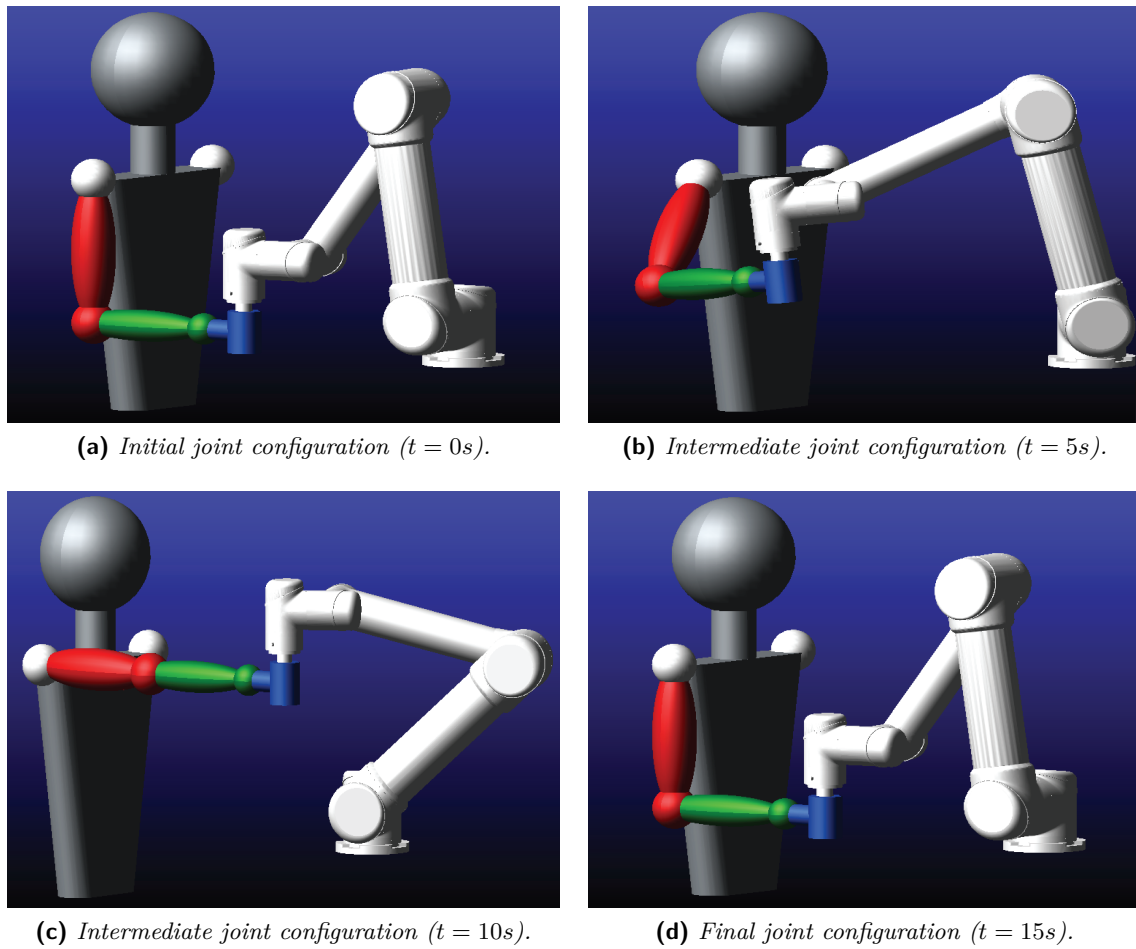
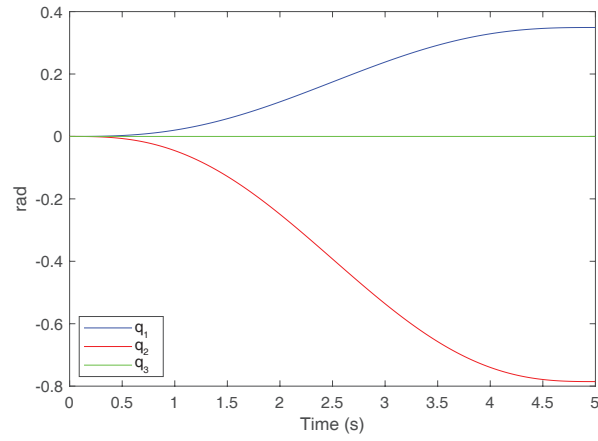


Figure 4.11.: Example of a 3-points motion.

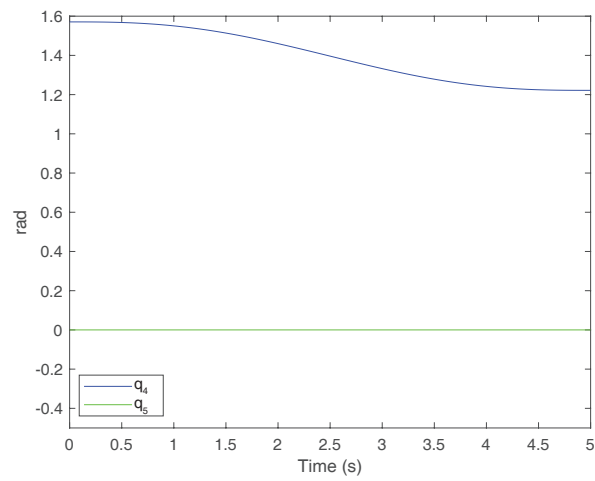
Chapter 4. Simulations

motion in Figure 4.16 and in Figure 4.17. One example of the "Scope" block result is reported in Figure 4.18, where are shown the displacements of the centre of gravity of human hand (or robot's end-effector) for each analyzed movement.

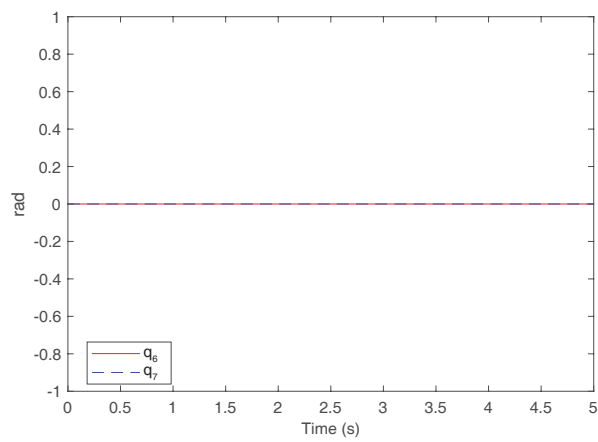
4.4. Results



(a) *Shoulder joint.*



(b) *Elbow joint.*



(c) *Wrist joint.*

Figure 4.12.: Human joint position during the one-way motion.

Chapter 4. Simulations

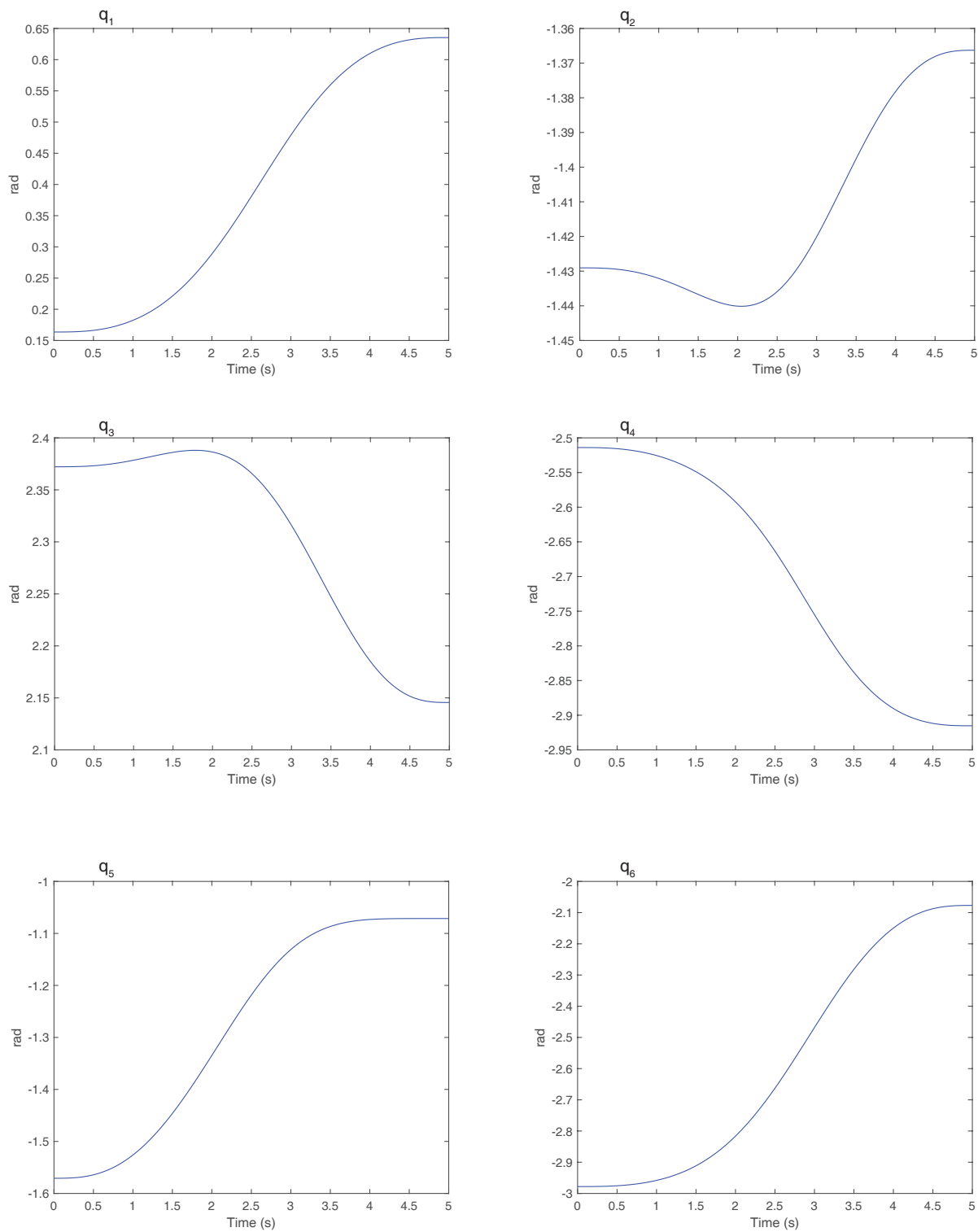
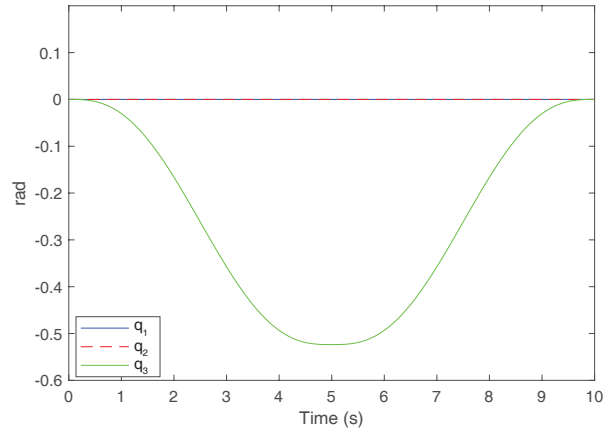
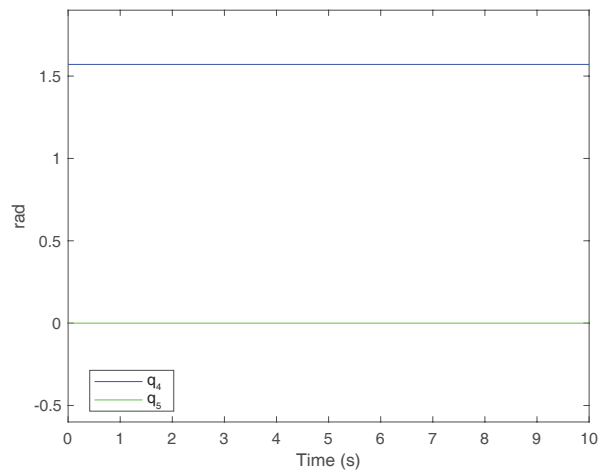


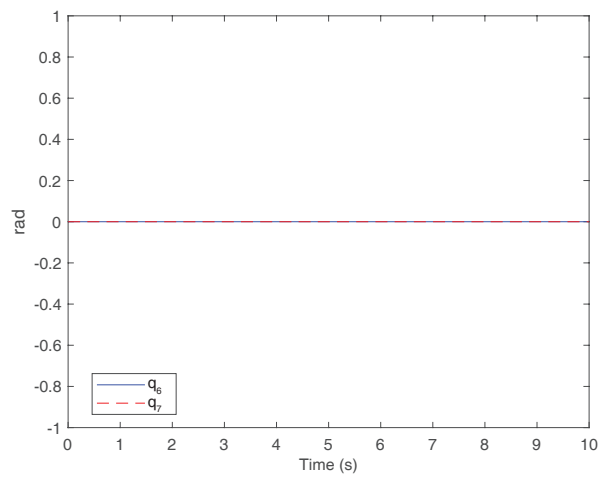
Figure 4.13.: Robot joint position during the one-way motion.



(a) *Shoulder joint.*



(b) *Elbow joint.*



(c) *Wrist joint.*

Figure 4.14.: Human joint position during the round-trip motion.

Chapter 4. Simulations

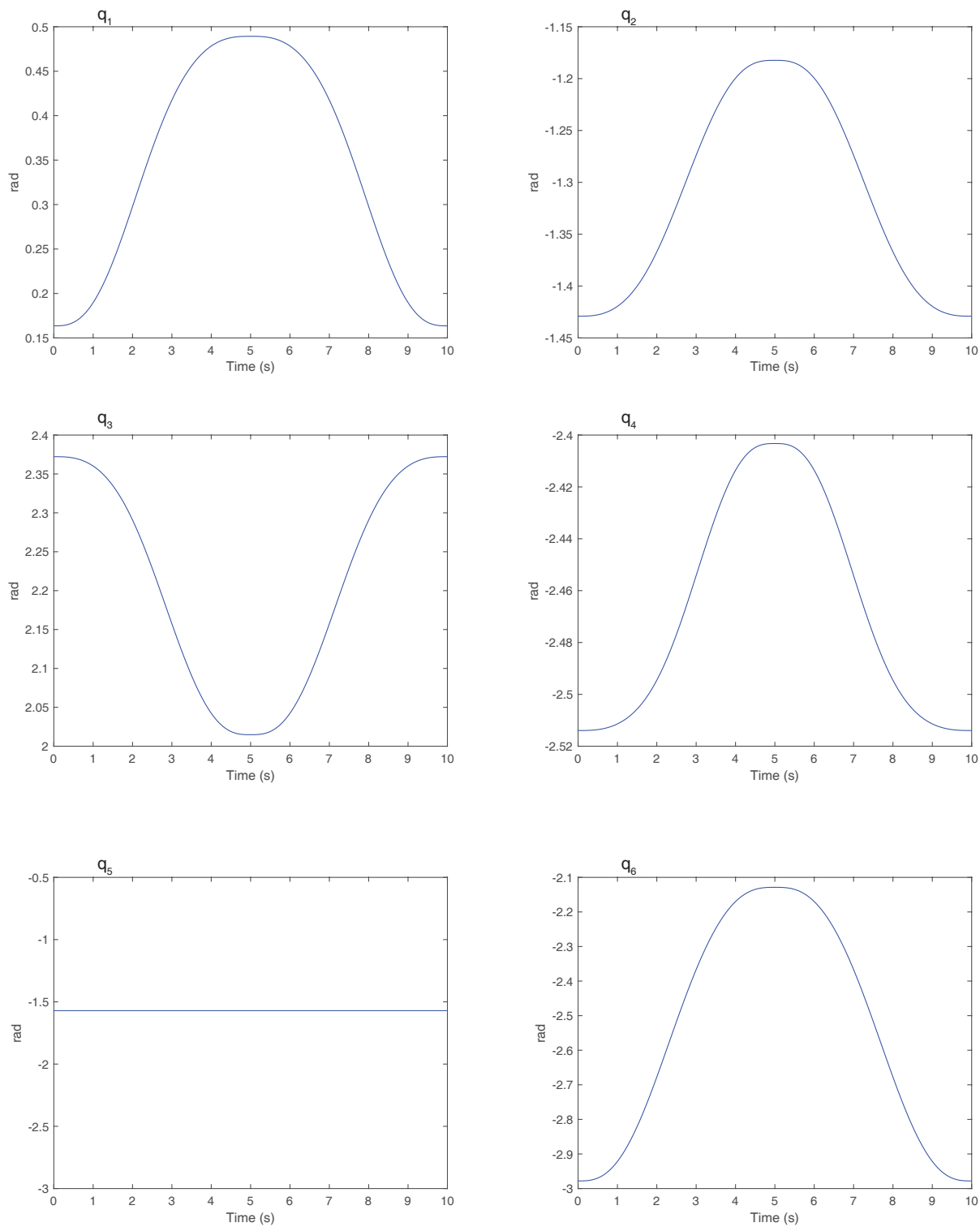
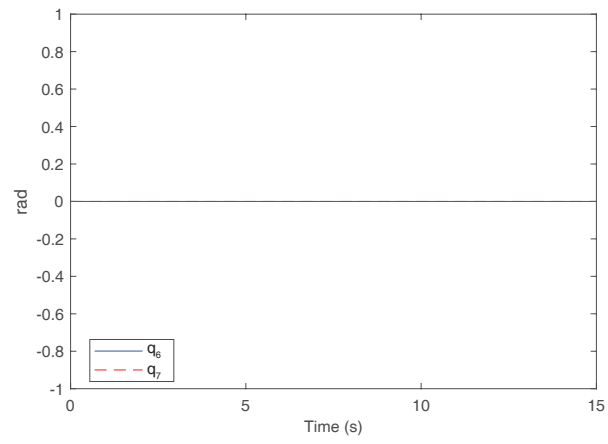
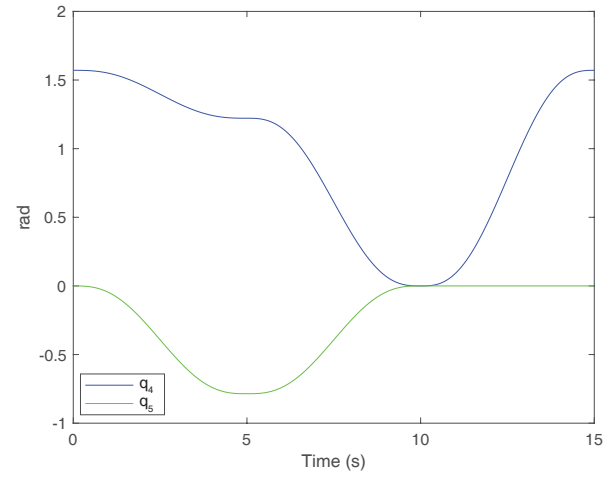
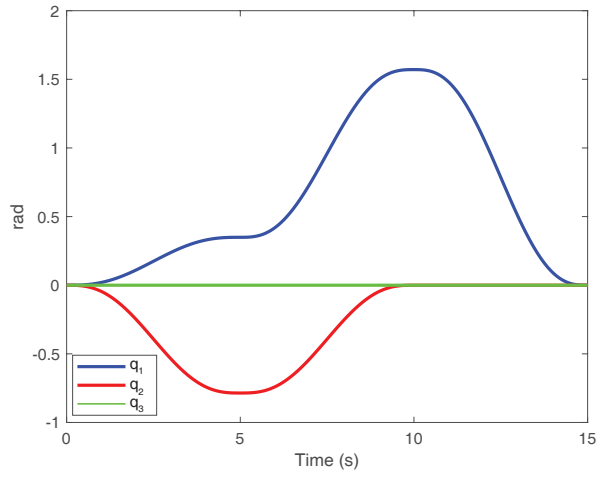
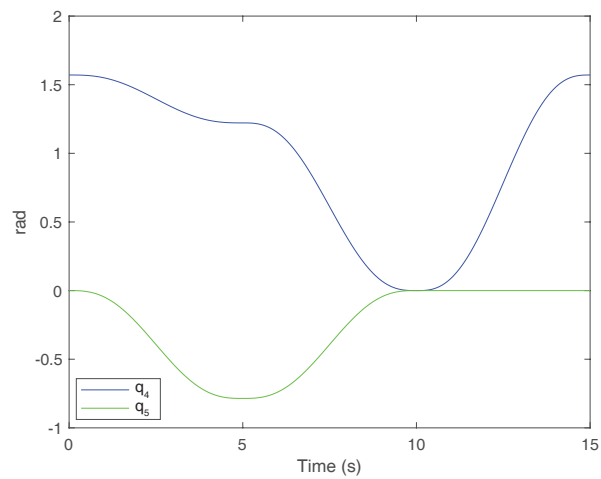


Figure 4.15.: Robot joint position during the round trip motion.

4.4. Results



(a) *Shoulder joint.*



(b) *Elbow joint.*

Chapter 4. Simulations

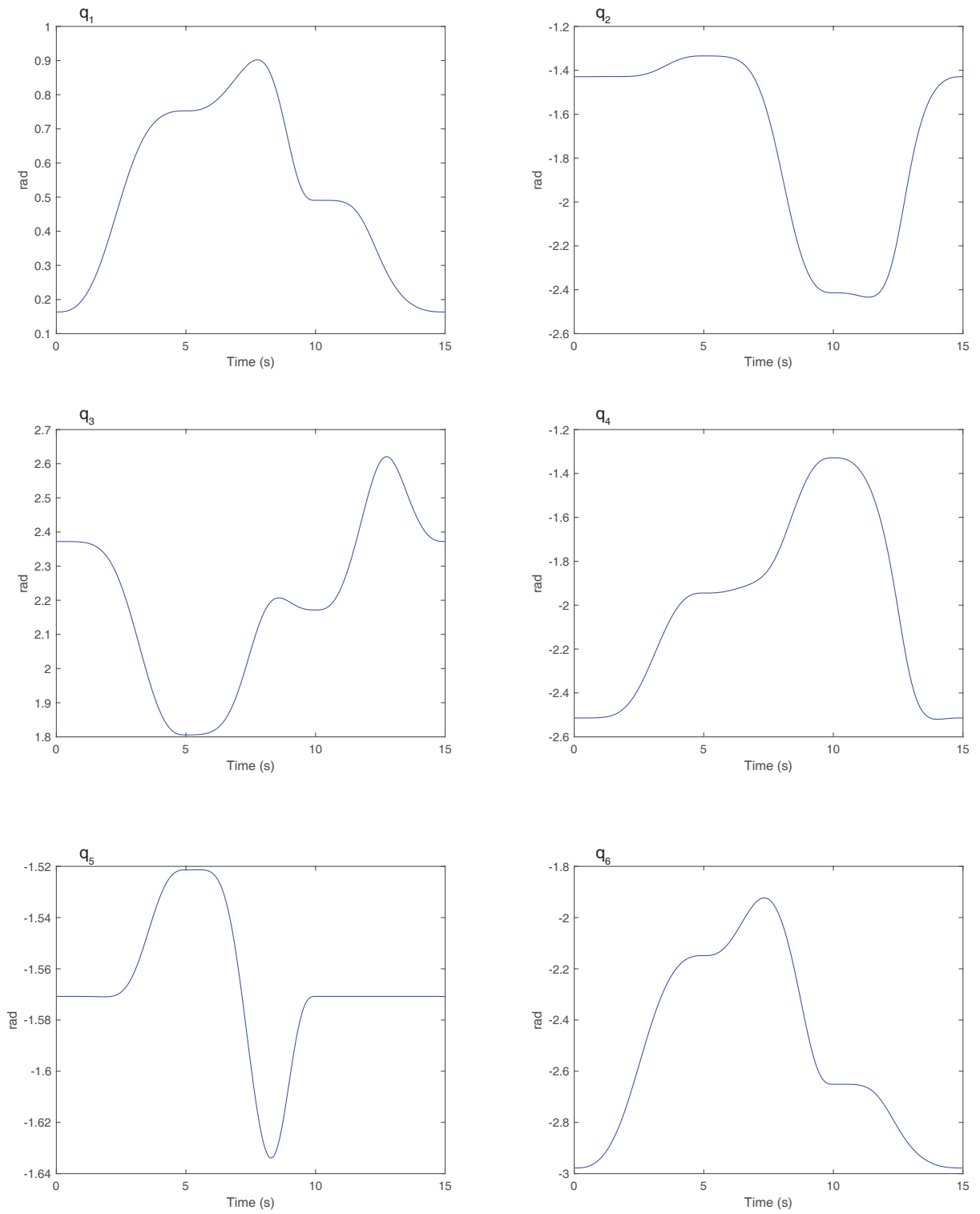


Figure 4.17.: Robot joint position during the 3-points motion.

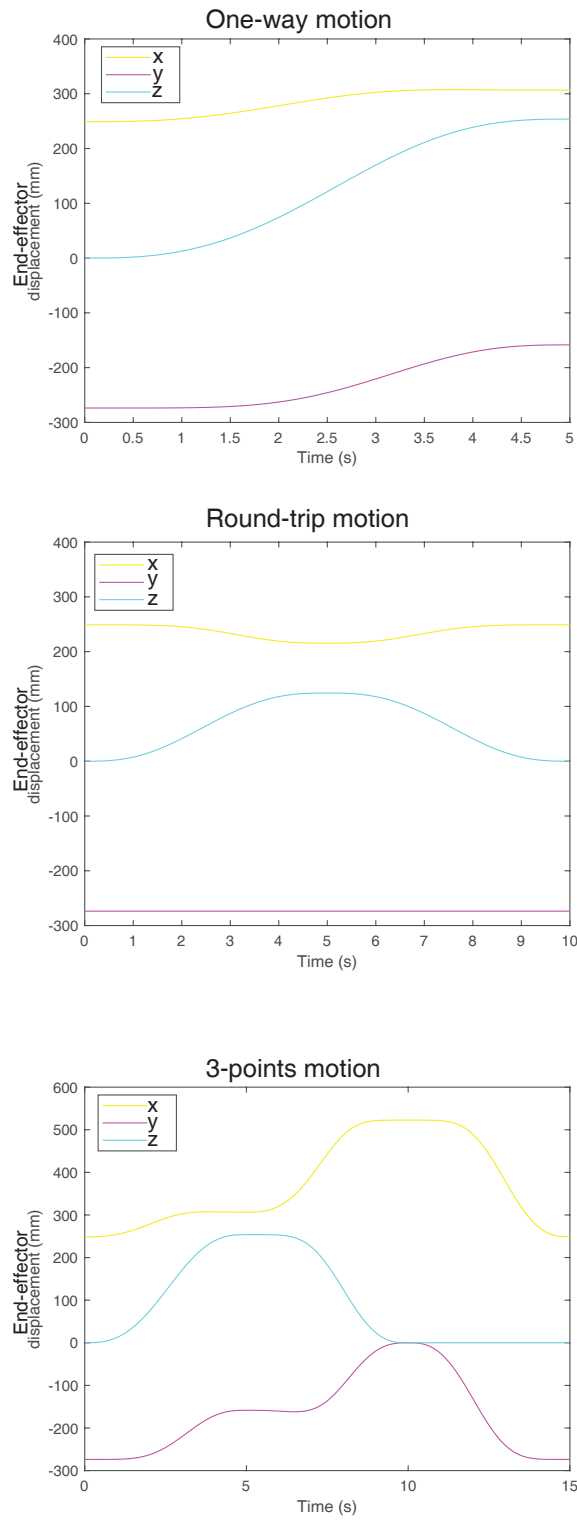


Figure 4.18.: Plot of Scope block related to the Adams' output sub-system.

Chapter 5.

Conclusions

Robots, unlike humans, can reach high accuracy, power and endurance over time. Thanks to their reliability and security, the cobots are able to compensate for the patient's inadequate strength, providing so a continuous feedback for the subjective perception of improvement. The automating rehabilitation training by robotic system is recommended for the capability of generating high number of exercise without overburdening therapists. For these reasons, the thesis project involves the cobots activity only in rehabilitation processes. In particular, it was assumed that a non-specific patient was capable of grabbing the robot's handle and able to reach toward the target with maximal effort. In the realization of human and robot models it has relied on the literature, focusing only on models which have standard measures and known reference frames. In this way, the human and robot designs for their direct and inverse kinematics are simple and reliable. The kinematics for the UR5 is really accurate, while the ones for the upper limb can be optimized including the synergy of muscles, tendons and ligaments. One more simplification was adopted considering the body segments as rigid bodies for the purpose of describing the human motion. Even though the two systems are extremely simplified, they allow a good quantitative and qualitative estimation.

However, the kinematic analysis is the science of motion. It provides the study of the positions and angles in human and robot arm during their motion, without regarding the forces that cause it. The computation of the motion planning algorithm is focused on the realization of automatic movement in off-line robot programming systems. This would allow the user to specify tasks more declaratively by stating what the operator wants rather than how to do it. Since in the collaborative rehabilitation process is the physiotherapist the one responsible for the assessment of the upper limb motion, the goal of the proposed algorithm is to design the robot joint trajectory perfectly in accordance with the human movement. In fact, the two models can perfectly execute the fixed movement without computational errors. In Adams-Simulink control model the human forward kinematic has been executed to compute the pose of the centre of gravity of human hand by using

Chapter 5. Conclusions

a given set of joint angles referred to the shoulder base frame. The inverse kinematic, instead, is the reverse of the forward ones. It is used to compute all possible sets of robot joint angles by using the known position of human hand. This approach provides a primary tool for the development of human-robot interface.

The physiological motions have been defined on the basis of possible rehabilitation procedures. Depending on the nature of the functional impairment, the patient is often unable to perform some simple exercises. Therefore, there are several rehabilitation protocols suitable for each pathological condition. The three upper limb movements performed in this study (one-way, round-trip and 3-points) are based on primitive movements, including both planar and spatial performance. The one-way motion is one example of shoulder flexion, shoulder abduction and elbow flexion in the spatial setting. This movement allows an engagement of two upper limb joints which constitute a number of basic human movements. The 3-points motion is a prolongation of the previous one including also the total flexion of the shoulder. The round-trip motion, instead, is a planar curved movement used for the execution of shoulder rotation and elbow flexion. The mobility in the upper extremities is often more difficult than the lower ones, which can seriously impact the progress of rehabilitation. In general, stroke patients had severe arm paresis or the arm remained entirely non-functional despite comprehension rehabilitation efforts. Thus, the flexion-extension of the upper arm and the forearm are essential exercises to be performed in a standard rehabilitation process and several safety features are incorporated into the system.

The proposed methodology will be further assessed including many other arm movements authorised by competent operators that are in line with rehabilitation procedures (i.g. circle motion). Moreover, future studies could involve the analysis of the three different working modalities (active, active-assisted and passive), focusing on the control system of human-robot. The research could make possible the analysis of the impact of rehabilitation on brain plasticity to adapt treatment resources to meet the needs of each patient and optimize the recovery process (centring on the recovery of muscle tone or neuromuscular rehabilitation). Future research should identify what features are essential to the efficacy of robotic manipulator. Integration of the collaborative robots into current rehabilitation practice holds the promise of improving the quality of physical rehabilitation, alleviating its labour-intensive aspects and increasing the efficiency of therapists.

Appendix A.

Denavit-Hartenberg convention

Denavit-Hartenberg convention (DH) is a method to define the relative position and orientation of two consecutive links defined by some fixed rules. In the Figure A.1 are shown DH kinematic parameters. Considering the axis i as the axis of the Joint connecting Link $i - 1$ to Link i , the convention is used to delimit link Frame i :

- Choose axis z_i along the axis of Joint $i + 1$.
- Locate the origin O_i at the intersection of axis z_i with the common normal to axes z_{i-1} and z_i . Also, locate O'_i at the intersection of the common normal with axis z_{i-1} .
- Choose axis x_i along the common normal to axes z_{i-1} and z_i with direction from Joint i to Joint $i + 1$.
- Choose axis y_i so as to complete a right-handed frame.

In some cases it is possible that DH convention does not give an unique definition of the link frame. The cases are:

- If for Frame 0 is specified only the direction of axis z_0 , the O_0 and x_0 can be arbitrarily chosen.
- For Frame n , since is no Joint $n + 1$, z_n is not uniquely defined while x_n has to be normal to axis z_{n-1} . Typically, Joint n is revolute, and thus z_n is to be aligned with the direction of z_{n-1} .
- When two consecutive axes are parallel, the common normal between them is not uniquely defined.
- When two consecutive axes intersect, the direction of x_i is arbitrary.
- When Joint i is prismatic, the direction of z_{i-1} is arbitrary.

Appendix A. Denavit-Hartenberg convention

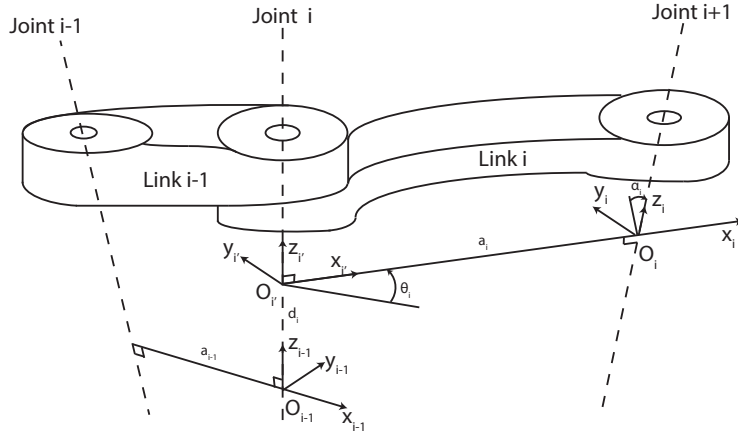


Figure A.1.: Denavit-Hartenberg kinematic parameters.

Then, the position and orientation of the Frame i with respect the Frame $i - 1$ are completely defined by the following parameters:

- a_i is the distance between O_i and O'_i ;
- d_i is the coordinate of O'_i along z_{i-1} ;
- α_i is the angle between axes z_{i-1} about axis x_i to be taken positive when rotation is made counter-clockwise;
- θ_i is the angle between axes x_{i-1} and x_i about axis z_{i-1} to be taken positive when rotation is made counter-clockwise.

Two of the four parameters (a_i and α_i) are always constant and dependent only on the geometry of the system. Among the remaining parameters, only one is variable depending on the type of joint that connects Link $i - 1$ to Link i . In particular, the variable is θ_i if the Joint i is revolute, otherwise is d_i for a prismatic joint.

At this point, it is possible to express the coordinate transformation between Frame i and Frame $i - 1$ according to the following steps:

- Choose a frame aligned with Frame $i - 1$.
- Translate the chosen frame by d_i along axis z_{i-1} ; this sequence aligns the current

frame with Frame i' and is described by the homogeneous transformation matrix

$$\mathbf{A}_{i'}^{i-1} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.1})$$

- Translate the frame aligned with the Frame i' by a_i along axis $x_{i'}$ and rotate it by α_i about axis $x_{i'}$; this sequence aligns the current frame with Frame i and is described by the homogeneous transformation matrix

$$\mathbf{A}_i^{i'} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.2})$$

- The resulting coordinate transformation is obtained by post-multiplication of the single transformations as

$$\mathbf{A}_i^{i-1}(q_i) = \mathbf{A}_{i'}^{i-1} \mathbf{A}_i^{i'} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.3})$$

Therefore, the transformation matrix from Frame i to Frame $i - 1$ is a function only of the joint variable q_i , that is θ_i for a revolute joint or d_i for a prismatic joint.

Appendix B.

Matlab code

In this section are reported the Matlab scripts adopted in this project, organized in the following list.

Matlab script B.1: One-way motion planning with a fifth-order polynomial function.

```
% output: Q (joint position in each instant of time)
% inputs: pi (initial position),
%         vi (initial velocity),
%         ai (initial acceleration),
%         pf (final position),
%         vf (final velocity),
%         af (final acceleration),
%         ti (initial time),
%         tf (final time),
%         t (instant of time).

function Q = polynomial_oneway_motion(pi,vi,ai,pf,vf,af,ti,
    tf,t)

T = tf-ti;

a0 = pi;
a1 = vi;
a2 = ai/2;
a3 = (20*(pf-pi)-(8*vf+12*vi)*T-(3*af-ai)*T^2)/(2*T^3);
a4 = (30*(pi-pf)+(14*vf+16*vi)*T+(3*af-2*ai)*T^2)/(2*T^4);
a5 = (12*(pf-pi)-6*(vf+vi)*T-(af-ai)*T^2)/(2*T^5);
```

Appendix B. Matlab code

```
Q=a0 + a1*(t-ti) + a2*(t-ti)^2 + a3*(t-ti)^3 + a4*(t-ti)^4 +  
    a5*(t-ti)^5;  
Q=deg2rad(Qa);  
end
```

Matlab script B.2: Round-trip motion planning with a fifth-order polynomial function.

```
function Q = polynomial_roundtrip_motion(pi,vi,ai,pf,vf,af,  
    ti,tf,t)  
  
T = tf-ti;  
  
a0 = pi;  
a1 = vi;  
a2 = ai/2;  
a3 = (20*(pf-pi)-(8*vf+12*vi)*T-(3*af-ai)*T^2)/(2*T^3);  
a4 = (30*(pi-pf)+(14*vf+16*vi)*T+(3*af-2*ai)*T^2)/(2*T^4);  
a5 = (12*(pf-pi)-6*(vf+vi)*T-(af-ai)*T^2)/(2*T^5);  
  
Qa=a0 + a1*(t-ti) + a2*(t-ti)^2 + a3*(t-ti)^3 + a4*(t-ti)^4 +  
    a5*(t-ti)^5;  
Qa=deg2rad(Qa);  
  
pi_a=pi;  
vi_a=vi;  
ai_a=ai;  
pf_a=pf;  
vf_a=vf;  
af_a=af;  
  
pi=pf_a;  
vi=vf_a;  
ai=af_a;  
pf=pi_a;  
vf=vi_a;  
af=ai_a;
```



```

a0 = pi;
a1 = vi;
a2 = ai/2;
a3 = (20*(pf-pi)-(8*vf+12*vi)*T-(3*af-ai)*T^2)/(2*T^3);
a4 = (30*(pi-pf)+(14*vf+16*vi)*T+(3*af-2*ai)*T^2)/(2*T^4);
a5 = (12*(pf-pi)-6*(vf+vi)*T-(af-ai)*T^2)/(2*T^5);

Qr=a0 + a1*(t-tf) + a2*(t-tf)^2 + a3*(t-tf)^3 + a4*(t-tf)^4 +
    a5*(t-tf)^5;
Qr=deg2rad(Qr);
if t<=tf
    Q=Qa;
else
    Q=Qr;
end
end

```

Matlab script B.3: 3-points motion planning with a fifth-order polynomial function.

```

%ABC= 3 points motion
function Q = polynomial_ABC_motion(pi,vi,ai,pf,pf_B,vf,af,ti
    ,tf,t)

T = tf-ti;
a0 = pi;
a1 = vi;
a2 = ai/2;
a3 = (20*(pf-pi)-(8*vf+12*vi)*T-(3*af-ai)*T^2)/(2*T^3);
a4 = (30*(pi-pf)+(14*vf+16*vi)*T+(3*af-2*ai)*T^2)/(2*T^4);
a5 = (12*(pf-pi)-6*(vf+vi)*T-(af-ai)*T^2)/(2*T^5);

Q_ab=a0 + a1*(t-ti) + a2*(t-ti)^2 + a3*(t-ti)^3 + a4*(t-ti)^4
    + a5*(t-ti)^5;
Q_ab=deg2rad(Q_ab);

tab=T;

```

Appendix B. Matlab code

```
%A
pi_A=pi; %pi_A (inititial position of A point)
vi_A=vi;
ai_A=ai;
pf_A=pf;
vf_A=vf;
af_A=af;

%B
pi=pf_A; %pf_A (final position of A point)
vi=vf_A;
ai=af_A;
pf=pf_B; %adding another input (final point in B)
vf=vi_A;
af=ai_A;

a0 = pi;
a1 = vi;
a2 = ai/2;
a3 = (20*(pf-pi)-(8*vf+12*vi)*T-(3*af-ai)*T^2)/(2*T^3);
a4 = (30*(pi-pf)+(14*vf+16*vi)*T+(3*af-2*ai)*T^2)/(2*T^4);
a5 = (12*(pf-pi)-6*(vf+vi)*T-(af-ai)*T^2)/(2*T^5);

Q_bc=a0 + a1*(t-tab) + a2*(t-tab)^2 + a3*(t-tab)^3 + a4*(t-
    tab)^4 + a5*(t-tab)^5;
Q_bc=deg2rad(Q_bc);
tbc=tab+tf;

%B
pi_B=pi;
vi_B=vi;
ai_B=ai;
pf_B=pf;
vf_B=vf;
af_B=af;
```

```

%C
pi=pf_B;
vi=vf_B;
ai=af_B;
pf=pi_A;
vf=vi_B;
af=ai_B

a0 = pi;
a1 = vi;
a2 = ai/2;
a3 = (20*(pf-pi)-(8*vf+12*vi)*T-(3*af-ai)*T^2)/(2*T^3);
a4 = (30*(pi-pf)+(14*vf+16*vi)*T+(3*af-2*ai)*T^2)/(2*T^4);
a5 = (12*(pf-pi)-6*(vf+vi)*T-(af-ai)*T^2)/(2*T^5);

Q_ca=a0 + a1*(t-tbc) + a2*(t-tbc)^2 + a3*(t-tbc)^3 + a4*(t-
tbc)^4 + a5*(t-tbc)^5;
Q_ca=deg2rad(Q_ca);

if t<=tab
    Q=Q_ab;
elseif (tab<t) && (t<=tbc)
    Q=Q_bc;
else
    Q=Q_ca;
end

end
end

```

Matlab script B.4: Direct kinematic function of the human arm.

```

function [T03,p] = directkinematic_human(q,height)
% 7 human joints
q1=q(1);

```

Appendix B. Matlab code

```
q2=q(2);
q3=q(3);
q4=q(4);
q5=q(5);
q6=q(6);
q7=q(7);

% anthropometric values
upper_arm = height*0.161;
forearm = height*0.146;
hand = height*0.106;
L1=upper_arm*[0;-1;0];
L2=forearm*[0;-1;0];
cog_hand=hand*0.506;
L3=cog_hand*[0;-1;0];

% shoulder
T0i = [cos(q1) -sin(q1) 0 0;
       sin(q1)  cos(q1) 0 0;
       0        0 1 0;
       0        0 0 1];
Tij = [ 1      0      0 0;
       0 cos(q2) -sin(q2) 0;
       0 sin(q2)  cos(q2) 0;
       0      0      0 1];
Tjk = [cos(q3) 0 sin(q3) 0;
       0 1      0 0;
       -sin(q3) 0 cos(q3) 0;
       0 0      0 1];
Tk1 = [eye(3) L1; 0 0 0 1];
T01 = T0i*Tij*Tjk*Tk1;

% elbow
T1i = [cos(q4) -sin(q4) 0 0;
       sin(q4)  cos(q4) 0 0;
       0        0 1 0];
```

```

        0      0 0 1];
Tij = [cos(q5) 0 sin(q5) 0;
        0 1      0 0;
       -sin(q5) 0 cos(q5) 0;
        0 0      0 1];
Tj2 = [eye(3) L2; 0 0 0 1];
T12 = T1i*Tij*Tj2;

% wrist
T2i = [cos(q6) -sin(q6) 0 0;
        sin(q6)  cos(q6) 0 0;
         0      0 1 0;
         0      0 0 1];
Tij = [ 1      0      0 0;
        0 cos(q7) -sin(q7) 0;
        0 sin(q7)  cos(q7) 0;
        0      0      0 1];
Tk3 = [eye(3) L3; 0 0 0 1];
T23 = T2i*Tij*Tk3;

T03 = T01*T12*T23;
p = T03(1:3,4);
end

```

Matlab script B.5: Inverse kinematic function of the human arm.

```

function [Q] = inverse_kinematic_human(X,dX,q_i,dt,h,j_limits
)
%anthropometric values
upper_arm = height*0.161; %Parametro antropometrico corretto
forearm = height*0.146;
cog_hand=height*0.106*0.506;

D=j_limits(:,2)-j_limits(:,1); % D = (q_max-q_min)
Av=1/2*(j_limits(:,2)+j_limits(:,1)); %Av is the average
value of q

```

Appendix B. Matlab code

```

n=size(X,1);
Q(1,:)=q_i;
for i=1:n-1

    v=dX(i,1:3).';
    a=X(i,4);
    b=X(i,5);
    c=X(i,6);
    da=dX(i,4);
    db=dX(i,5);
    dc=dX(i,6);
    R=rotationZ(a)*rotationX(b)*rotationY(c);
    dR = [ - da*(cos(c)*sin(a) + cos(a)*sin(b)*sin(c)) - dc*(
        cos(a)*sin(c) + cos(c)*sin(a)*sin(b)) - db*cos(b)*sin(
        a)*sin(c),    db*sin(a)*sin(b) - da*cos(a)*cos(b), dc*(
        cos(a)*cos(c) - sin(a)*sin(b)*sin(c)) - da*(sin(a)*sin
        (c) - cos(a)*cos(c)*sin(b)) + db*cos(b)*cos(c)*sin(a);
        da*(cos(a)*cos(c) - sin(a)*sin(b)*sin(c)) - dc*(sin(a)
        )*sin(c) - cos(a)*cos(c)*sin(b)) + db*cos(a)*cos(b)
        )*sin(c), - da*cos(b)*sin(a) - db*cos(a)*sin(b),
        da*(cos(a)*sin(c) + cos(c)*sin(a)*sin(b)) + dc*(
        cos(c)*sin(a) + cos(a)*sin(b)*sin(c)) - db*cos(a)*
        cos(b)*cos(c);
        db*sin(b)*sin(c) - dc*cos(b)*cos(c),
                                db*cos(b),

        - db*cos(c)*sin(b) - dc*cos(b)*sin(c)];
    omega_tilde=dR*R. ';
    omega=[omega_tilde(3,2) omega_tilde(1,3) omega_tilde(2,1)
        ]. ';
    J=J_function(Q(i,:),h);
    lambda=10;
    w=1+lambda*abs((Q(i,:)'-Av)./D); %formula for weighed
        elements
    W=diag(w);
    % if I want only the identity matrix:

```

```

    % W=eye(7);
    Jinv=W\J.'*inv(J*(W\J.'));
    Q(i+1,:)=Q(i,:)+dt*(Jinv*[v; omega]).';
end
end

```

Matlab script B.6: Rototraslation function to refer robot base frame to the shoulder fixed frame.

```

%% Rototraslation to refer the robot base frame (R) respect
the shoulder fixed frame (H or O)
function TH_R =rototraslation_RH(T3_0)
%% I relate the hand with the robot end-effector changing
only the orientation (Tori)
% Tori=[-1 0 0 0;0 1 0 0;0 0 -1 0;0 0 0 1];
% T3_0_ori=T3_0*Tori;

%% Computation of the matrix T (robot-shoulder) TR_0
Rr_0=[1 0 0;...
      0 0 1;...
      0 -1 0];
pR_0=[700; -300; -170]; % [mm] is the distance between the
robot base frame and the shoulder fixed frame
%pR_0=[0; 0; 0]; % in case the robot is located on human
shoulder

TR_0=[Rr_0,pR_0;0 0 0 1];

%% Computation of the matrix TH_R (robot-hand)
TH_R=reshape(TR_0\T3_0,[16 1]); % using reshape function to
be in line with the Adams inputs
end

```

Matlab script B.7: Function for generating OpenSim motion files.

```

function filemotjoints (Q,tablename,filename)

time = Q(:,1);

```

Appendix B. Matlab code

```
arm_flex_r = Q(:,2);
arm_add_r = Q(:,3);
arm_rot_r = Q(:,4);
elbow_flex_r = Q(:,5);
pro_sup_r = Q(:,6);
wrist_flex_r = Q(:,7);
wrist_dev_r = Q(:,8);

T = table(time,arm_flex_r,arm_add_r,arm_rot_r,elbow_flex_r,
          pro_sup_r,wrist_flex_r,wrist_dev_r);
writetable(T,tablename,'Delimiter',' ');

r = length(Q(:,1));
c = 8;
header = fopen(filename,'w');
fprintf(header,'arm_motion\n\nRows=%d\n\nColumns=%d\n\n
           ninDegrees=yes\n\nendheader\n',r,c);
fclose(header);

end
```

Matlab script B.8: Direct kinematic function of the robotic arm.

```
function [p6,p5]=directkinematic_robot(theta)
n = length(theta);
%%DH parameters
d1=8.916; d2=0; d3=0; d4=10.915; d5=9.456; d6=8.23;
d=[d1,d2,d3,d4,d5,d6];
a1=0; a2=42.2; a3=39.225; a4=0; a5=0; a6=0;
a=[a1, a2,a3,a4,a5,a6];
alpha1= pi/2; alpha2=0; alpha3=0; alpha4=pi/2; alpha5=-pi/2;
alpha6=0;
alpha=[alpha1, alpha2, alpha3, alpha4, alpha5, alpha6];
% The rototraslation matrix between one joint and the next
one has always
% this standardization
% A=[cos(theta_i) -cos(alpha_i)*sin(theta_i) sin(alpha_i)*
```



```

    sin(theta_i)    ai*cos(theta_i);
%   sin(theta_i)    cos(alpha_i)*sin(theta_i)    -sin(alpha_i)*
    sin(theta_i)    ai*sin(theta_i);
%           0           sin(alpha_i)           cos
    (alpha_i)           di;
%           0           0
           0           1]
F6=eye(4);
for i=1:n
    F_=[cos(theta(i))    -cos(alpha(i))*sin(theta(i))    sin(
        alpha(i))*sin(theta(i))    ai*cos(theta(i));
        sin(theta(i))    cos(alpha(i))*sin(theta(i))    -sin(
            alpha(i))*sin(theta(i))    ai*sin(theta(i));
            0           sin(alpha(i))
                cos(alpha(i))           di;
            0           0
                0           1];
    F6=F6*F_;
end
A1=rototraslation(theta(1), alpha1,a1,d1); %from 0 to 1
A2=rototraslation(theta(2), alpha2,a2,d2); %from 1 to 2
A3=rototraslation(theta(3), alpha3,a3,d3); %from 2 to 3
A4=rototraslation(theta(4), alpha4,a4,d4); %from 3 to 4
A5=rototraslation(theta(5), alpha5,a5,d5); %from 4 to 5
A6=rototraslation(theta(6), alpha6,a6,d6); %from 5 to 6
origin=[0;0;0;1];
zero=zeros(4,1);
frame1=A1*origin;
frame2=A1*A2*origin;
frame3=A1*A2*A3*origin;
frame4=A1*A2*A3*A4*origin;
frame5=A1*A2*A3*A4*A5*origin;
frame6=A1*A2*A3*A4*A5*A6*origin;
position=[zero frame1 frame2 frame3 frame4 frame5 frame6];
X=position(1,:);
Y=position(2,:);

```

Appendix B. Matlab code

```
Z=position(3,:);  
%output of the function is the position and orientation  
%angular convention: z-x-y  
betaframe6=asin(F6(3,2));  
gammaframe6=asin(-F6(3,1)/cos(betaframe6));  
if F6(3,3) <=0  
    gammaframe6=pi-gammaframe6;  
end  
alphaframe6=asin(-F6(1,2)/cos(betaframe6));  
if F6(2,2) <=0  
    alphaframe6=pi-alphaframe6;  
end  
  
p5=[frame5(1,1); frame5(2,1); frame5(3,1)];  
p6=[frame6(1,1); frame6(2,1); frame6(3,1); alphaframe6;  
    betaframe6; gammaframe6];  
end
```

For the inverse kinematic function of UR5 the Matlab code is taken from the Kutzer function, covered by Copyright[©]. In that function some adjustments have been done, as the increase of the inputs and the addition of the handle system.

Bibliography

- [1] A. Vysocky and P. Novak, “Human-robot collaboration in industry,” *MM Science Journal*, vol. 9, no. 2, pp. 903–906, 2016.
- [2] G. Michalos, S. Makris, P. Tsarouchi, T. Guasch, D. Kontovrakis, and G. Chrysolouris, “Design considerations for safe human-robot collaborative workplaces,” *Procedia CIRP*, vol. 37, pp. 248–253, 2015.
- [3] U. Robots, “User manual: Ur5,” 2013.
- [4] G. Legnani and G. Palmieri, *Fondamenti di meccanica e biomeccanica del movimento*. CittàStudi, 2016, vol. 424.
- [5] S. L. Delp, F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, E. Guendelman, and D. G. Thelen, “Opensim: open-source software to create and analyze dynamic simulations of movement,” *IEEE transactions on biomedical engineering*, vol. 54, no. 11, pp. 1940–1950, 2007.
- [6] M. A. K. Bahrin, M. F. Othman, N. N. Azli, and M. F. Talib, “Industry 4.0: A review on industrial automation and robotic,” *Jurnal Teknologi*, vol. 78, no. 6-13, pp. 137–143, 2016.
- [7] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier, and P. Fraisse, “Collaborative manufacturing with physical human–robot interaction,” *Robotics and Computer-Integrated Manufacturing*, vol. 40, pp. 1–13, 2016.
- [8] A. M. Zanchettin, N. M. Ceriani, P. Rocco, H. Ding, and B. Matthias, “Safety in human-robot collaborative manufacturing environments: Metrics and control,” *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 882–893, 2015.
- [9] C. Thomas, B. Matthias, and B. Kuhlenkötter, “Human-robot-collaboration-new applications in industrial robotics,” in *International Conference on Competitive Manufacturing*, 2016, pp. 293–299.

Bibliography

- [10] P. Maciejasz, J. Eschweiler, K. Gerlach-Hahn, A. Jansen-Troy, and S. Leonhardt, “A survey on robotic devices for upper limb rehabilitation,” *Journal of neuroengineering and rehabilitation*, vol. 11, no. 1, p. 3, 2014.
- [11] E. Oña, R. Cano-de la Cuerda, P. Sánchez-Herrera, C. Balaguer, and A. Jardón, “A review of robotics in neurorehabilitation: Towards an automated process for upper limb,” *Journal of healthcare engineering*, vol. 2018, 2018.
- [12] L. Pignolo, “Robotics in neuro-rehabilitation,” *Journal of Rehabilitation Medicine*, vol. 41, no. 12, pp. 955–960, 2009.
- [13] M. Caimmi, E. Visani, F. Digiaco, A. Scano, A. Chiavenna, C. Gramigna, L. Molinari Tosatti, S. Franceschetti, F. Molteni, and F. Panzica, “Predicting functional recovery in chronic stroke rehabilitation using event-related desynchronization-synchronization during robot-assisted movement,” *BioMed research international*, vol. 2016, 2016.
- [14] J. Nielsen, A. S. Sørensen, T. S. Christensen, T. R. Savarimuthu, and T. Kulvicius, “Individualised and adaptive upper limb rehabilitation with industrial robot using dynamic movement primitives,” in *ICRA 2017 Workshop on Advances and challenges on the development, testing and assessment of assistive and rehabilitation robots: Experiences from engineering and human science research*, vol. 1, 2017, p. 40.
- [15] P. S. Lum, C. G. Burgar, P. C. Shor, M. Majmundar, and M. Van der Loos, “Robot-assisted movement training compared with conventional therapy techniques for the rehabilitation of upper-limb motor function after stroke,” *Archives of physical medicine and rehabilitation*, vol. 83, no. 7, pp. 952–959, 2002.
- [16] C. Duret, A.-G. Grosmaire, and H. I. Krebs, “Robot-assisted therapy in upper extremity hemiparesis: Overview of an evidence-based approach,” *Frontiers in neurology*, vol. 10, 2019.
- [17] T. Nef, M. Guidali, and R. Riener, “Armin iii—arm therapy exoskeleton with an ergonomic shoulder actuation,” *Applied Bionics and Biomechanics*, vol. 6, no. 2, pp. 127–142, 2009.
- [18] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.
- [19] B. Siciliano, “Kinematic control of redundant robot manipulators: A tutorial,” *Journal of intelligent and robotic systems*, vol. 3, no. 3, pp. 201–212, 1990.

- [20] I. O. for Standardization (ISO), “Safety of machinery—general principles for design—risk assessment and risk reduction,” 2010.
- [21] I. 10218-2, “Robots and robotic devices—safety requirements for industrial robots—part 2: robot systems and integration,” 2011.
- [22] M. J. Rosenstrauch and J. Krüger, “Safe human-robot-collaboration-introduction and experiment using iso/ts 15066,” in *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*. IEEE, 2017, pp. 740–744.
- [23] I. ISO, “Ts 15066 (2016): robots and robotic devices—collaborative robots,” *Geneva: International Organization for Standardization*, 2016.
- [24] Å. Fast-Berglund, F. Palmkvist, P. Nyqvist, S. Ekered, and M. Åkerman, “Evaluating cobots for final assembly,” *Procedia CIRP*, vol. 44, pp. 175–180, 2016.
- [25] E. ISO, “9283: 2003, manipulating industrial robots—performance criteria and related test methods,” *Geneva: International Standards Organization*, 2003.
- [26] S. El Zaatari, M. Marei, W. Li, and Z. Usman, “Cobot programming for collaborative industrial tasks: An overview,” *Robotics and Autonomous Systems*, vol. 116, pp. 162–180, 2019.
- [27] A. Bauer, D. Wollherr, and M. Buss, “Human–robot collaboration: a survey,” *International Journal of Humanoid Robotics*, vol. 5, no. 01, pp. 47–66, 2008.
- [28] S. A. Green, M. Billingham, X. Chen, and J. G. Chase, “Human-robot collaboration: A literature review and augmented reality approach in design,” *International journal of advanced robotic systems*, vol. 5, no. 1, p. 1, 2008.
- [29] E. Kyrkjebø, M. J. Laastad, and Ø. Stavadahl, “Feasibility of the ur5 industrial robot for robotic rehabilitation of the upper limbs after stroke,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–6.
- [30] M. Safeea and P. Neto, “Kuka sunrise toolbox: Interfacing collaborative robots with matlab,” *IEEE Robotics & Automation Magazine*, vol. 26, no. 1, pp. 91–96, 2018.
- [31] S. Mokaram, J. M. Aitken, U. Martinez-Hernandez, I. Eimontaite, D. Cameron, J. Rolph, I. Gwilt, O. McAree, and J. Law, “A ros-integrated api for the kuka lbr iiwa collaborative robot,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15 859–15 864, 2017.

Bibliography

- [32] M. Desmurget and C. Prablanc, “Postural control of three-dimensional prehension movements,” *Journal of neurophysiology*, vol. 77, no. 1, pp. 452–464, 1997.
- [33] M. A. Lemay and P. E. Crago, “A dynamic model for simulating movements of the elbow, forearm, and wrist,” *Journal of biomechanics*, vol. 29, no. 10, pp. 1319–1330, 1996.
- [34] R. Raikova, “A general approach for modelling and mathematical investigation of the human upper limb,” *Journal of biomechanics*, vol. 25, no. 8, pp. 857–867, 1992.
- [35] A. Gams and J. Lenarcic, “Humanoid arm kinematic modeling and trajectory generation,” in *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006*. IEEE, 2006, pp. 301–305.
- [36] R. Prokopenko, A. Frolov, E. Biryukova, and A. Roby-Brami, “Assessment of the accuracy of a human arm model with seven degrees of freedom,” *Journal of biomechanics*, vol. 34, no. 2, pp. 177–185, 2001.
- [37] M. Mihelj, “Inverse kinematics of human arm based on multisensor data integration,” *Journal of Intelligent and Robotic Systems*, vol. 47, no. 2, pp. 139–153, 2006.
- [38] T. Flash, Y. Meirovitch, and A. Barliya, “Models of human movement: Trajectory planning and inverse kinematics studies,” *Robotics and Autonomous Systems*, vol. 61, no. 4, pp. 330–339, 2013.
- [39] P. M. Kebria, S. Al-Wais, H. Abdi, and S. Nahavandi, “Kinematic and dynamic modelling of ur5 manipulator,” in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2016, pp. 004 229–004 234.
- [40] S. Moe, G. Antonelli, A. R. Teel, K. Y. Pettersen, and J. Schrimpf, “Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis, and experimental results,” *Frontiers in Robotics and AI*, vol. 3, p. 16, 2016.
- [41] K. P. Hawkins, “Analytic inverse kinematics for the universal robots ur-5/ur-10 arms,” Georgia Institute of Technology, Tech. Rep., 2013.
- [42] A. Seth, M. Sherman, J. A. Reinbolt, and S. L. Delp, “Opensim: a musculoskeletal modeling and simulation framework for in silico investigations and exchange,” *Procedia Iutam*, vol. 2, pp. 212–232, 2011.

- [43] J. A. Reinbolt, A. Seth, and S. L. Delp, "Simulation of human movement: applications using opensim," *Procedia Iutam*, vol. 2, pp. 186–198, 2011.
- [44] S. Frimpong and Y. Li, "Virtual prototype simulation of hydraulic shovel kinematics for spatial characterization in surface mining operations," *International Journal of Surface Mining, Reclamation and Environment*, vol. 19, no. 4, pp. 238–250, 2005.
- [45] T. Brezina, Z. Hadas, and J. Vetiska, "Using of co-simulation adams-simulink for development of mechatronic systems," in *14th International Conference Mechatronika*. IEEE, 2011, pp. 59–64.
- [46] D.-l. Zhu, J.-y. Qin, Y. Zhang, H. Zhang, and M.-m. Xia, "Research on co-simulation using adams and matlab for active vibration isolation system," in *2010 International Conference on Intelligent Computation Technology and Automation*, vol. 2. IEEE, 2010, pp. 1126–1129.