BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF MECHANICAL ENGINEERING

FAKULTA STROJNÍHO INŽENÝRSTVÍ

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

ÚSTAV AUTOMATIZACE A INFORMATIKY

DESIGN OF AN ADVANCED ROBOTIC CELL IN THE CONTEXT OF INDUSTRY 4.0

NÁVRH POKROČILÉ ROBOTICKÉ BUŇKY V KONTEXTU PRŮMYSLU 4.0

DOCTORAL THESIS

DIZERTAČNÍ PRÁCE

AUTHOR AUTOR PRÁCE	Ing. Roman Parák
SUPERVISOR ŠKOLITEL	prof. Ing. Radomil Matoušek, Ph.D.

BRNO 2023

Abstract

Text ...

Keywords

keyword 1, keyword 2, etc.

Abstrakt

Text ...

Klíčová slova

klíčové slovo 1, klíčové slovo 2, atd.

Bibliographic Citation

PARÁK, Roman. Design of Advanced Methods in the Field of Industrial Robotics, Fitting into the Concept of Industry 4.0. Brno, 2024. Doctoral Thesis. Brno University of Technology, Faculty of Mechanical Engineering, Institute of Automation and Computer Science. Supervised by prof. Ing. Radomil Matoušek, Ph.D.

Declaration of Authorship

I declare that I have written my doctoral thesis on the theme of "Design of Advanced Methods in the Field of Industrial Robotics, Fitting into the Concept of Industry 4.0" independently, under the guidance of the doctoral thesis supervisor, and using the technical literature and other sources of information, which are all quoted in the thesis and detailed in the list of literature at the end of the thesis.

As the author of the doctoral thesis, I furthermore declare that, as regards the creation of this doctoral thesis, I have not infringed any copyright. In particular, I have not unlawfully encroached on anyone's personal and/or ownership rights, and I am fully aware of the consequences in the case of breaking Regulation S 11 and the following of the Copyright Act No. 121/2000 Sb., and of the rights related to intellectual property rights and changes in some Acts (Intellectual Property Act) and formulated in later regulations, inclusive of the possible consequences resulting from the provisions of Criminal Act No. 40/2009 Sb., Section 2, Head VI, Part 4.

February 29, 2024

....	....
Date	Ing. Roman Parák

Preface

Text ...

Acknowledgements

Text ...

Contents

CHAPTER 1

Introduction

"An automated machine that does just one thing is not a robot. It is simply automation. A robot should have the capability of handling a range of jobs at a factory."

— Joseph Engelberger (1925 - 2015), *"The Father of Robotics"*

Text ...

CHAPTER 2

Current State in the Field of Industry 4.0

The following chapter presents the state-of-the-art in Industry 4.0, the basic vision of which was first presented in 2011 by Professor Wolfgang Wahlster at the Hannover Messe trade fair in Germany [1]. A detailed concept of the Fourth Industrial Revolution was later presented at the same fair in 2013 [2].

The chapter begins with a concise overview of the historical context surrounding the emergence of the Fourth Industrial Revolution (Section 2.1). Following this, it delves into the fundamental characteristics of Industry 4.0 (Section 2.2) and explores the key pillars underpinning this industrial revolution (Section 2.3). The emphasis is placed on the essential aspects of system integration (Subsection 2.3.1), autonomous robotics (Subsection 2.3.2), and simulation (Subsection 2.3.3). While providing a succinct overview of the remaining pillars (Subsection 2.3.4), the chapter concludes by examining future trends in the industrial field (Section 2.4). This exploration considers the possibility of a new revolution or the evolution of the existing concept of Industry 4.0, with a focus on improving its key pillars.

2.1 History of the Industrial Revolution

In this section, we briefly discuss the history of the rise of the Fourth Industrial Revolution (usually referred to as Industry 4.0) [3]. The historical process of industrial modernization, from the First to the Third Industrial Revolution, is thoroughly depicted in the book "The Industrial Revolution in World History" [4], and a brief review of these three revolutions can be found in [5, 6, 7]. The Fourth Industrial Revolution is discussed in the book "The Fourth Industrial Revolution" [8], but as a still relatively new area of research, it is more widely described in scientific publications (see [9, 10, 11, 12, 13]).

The historical process of the sequence of industrial revolutions with key pillars is depicted in Figure 2.1.

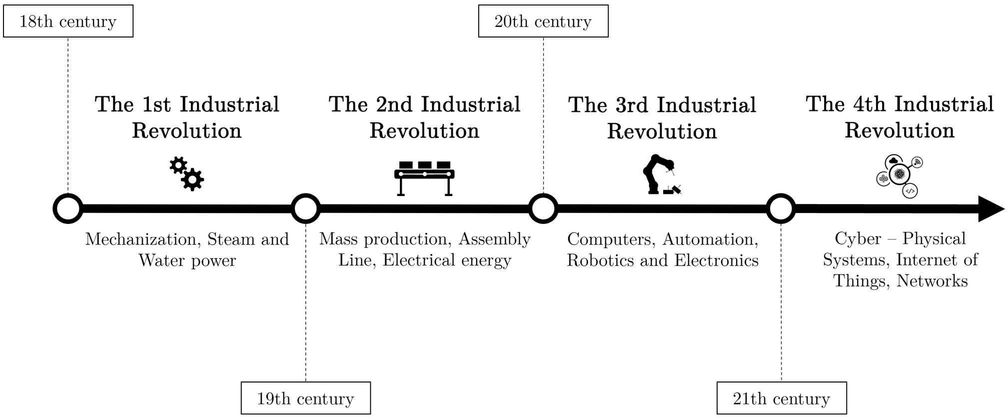

Figure 2.1: A visualization of the historical process of industrial revolutions that began in the 18th century and continues to the present day.

The First Industrial Revolution

The first phase of the Industrial Revolution began at the end of the 18th century, more precisely in 1760, and lasted until 1880.

The major milestones of the first industrial revolution include the invention of the steam engine and the development of steam power, as well as the use of turbine engines and water as power sources. The steam engine allowed the transition from agriculture to a new production process. This transition involved the use of coal as the main source of energy.

The combination of steam power and mechanized production caused a significant change in performance, not only in terms of the growth of the regional and global market economy but also in education, where the science and technology sector was inspired to restructure academic fields.

The Second Industrial Revolution

The second phase of the Industrial Revolution began in 1880 and lasted until 1950.

One of the major milestones of the first industrial revolution was the invention of the internal combustion engine. This invention facilitated technological advances in the industry, leading to rapid industrialization through the utilization of oil and electricity for mass production and assembly lines. A wave of systemic change has led to the belief that science and technology are the ways to a better life, and that progress is, in many ways, necessary. The revolution has brought fundamental changes in standardization and precision manufacturing, as well as large-scale technological infrastructure, such as electricity grids and new forms of public transport based on the internal combustion engine.

In addition to innovations such as the steamship, the telephone, and the gas turbine, the public developed a desire for goods, travel, and, not least, information, which were major factors in future development.

The Third Industrial Revolution

The third phase of the Industrial Revolution began in 1950 and lasted until 2010.

A characteristic feature of the Third Industrial Revolution was the implementation of electronics and information technology to automate production. With the advent of computers, the infrastructure was established, resulting in a significant change in information theory and the potency of data. Last but not least, new channels were created to share information. In many ways, the rapid advancement towards enhanced computing power has led to a more interconnected and complex problems that need to be addressed.

Innovations, such as programmable logic controllers and single / multiple purpose robotic systems, as well as the advancement of nuclear power, have opened the door to new areas of research, including space, robotics, and biotechnology.

The Fourth Industrial Revolution

The fourth phase, also called Industry 4.0, was first introduced in 2011 by Professor Wolfgang Wahlster at the Hannover Messe trade fair in Germany [1] and continues to the present day.

A characteristic feature of Industry 4.0 is the transformation of industrial production through the integration of digital and internet technologies, the utilization of cyber-physical systems, artificial intelligence techniques, augmented reality, physical simulation, additive technologies, and other key aspects to achieve the greatest possible flexibility in the production process.

A more detailed description of Industry 4.0, including the characteristics of the industrial concept and a brief introduction to the main pillars, is described in Section 2.2.

2.2 Characteristics of the Fourth Industrial Revolution

As the title implies, the following section introduces the characteristics of the Fourth Industrial Revolution.

The main idea of the Industry 4.0 concept involves the integration of intelligent machines and systems into the manufacturing processes of industrial enterprises [14, 15]. The concept of the Fourth Industrial Revolution is based on the nine main pillars (see Fig. 2.2) [16, 17, 18], which together form the core idea of the digitization of industry. The aim of the concept is to increase work efficiency and personalization, which leads to flexibility in changes to the production of a designated range of products. The Fourth Industrial Revolution focuses not only on changes in technology development but also on the way people work and the utilization of their creativity in various industries. By increasing the level of information processing and evaluation through the integration of AI techniques, the concept achieves improvements in various areas such as security, human-machine collaboration [19], predictive maintenance, visual inspection, etc. In addition to the industrial sector, where the Industry 4.0 initiative is an integral part, it also affects the development of education [20, 21], transportation, agriculture, and many other fields.

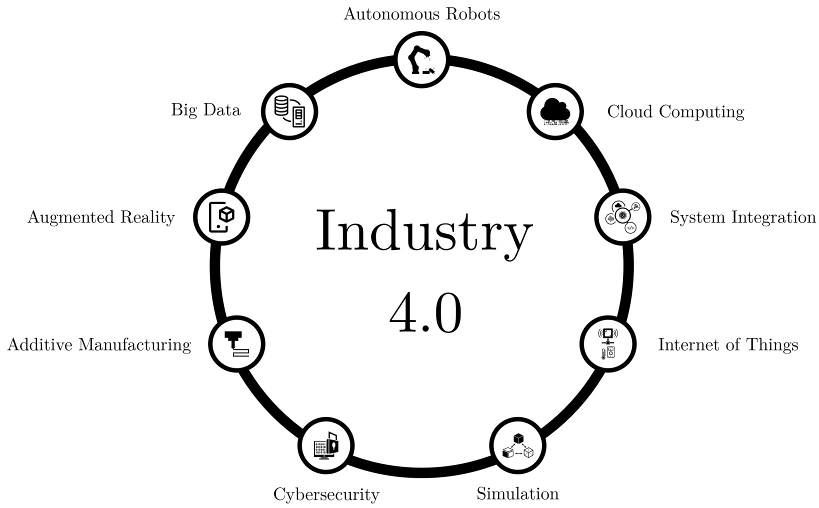

Figure 2.2: The nine main pillars of the concept of the Fourth Industrial Revolution.

Industry 4.0 encompasses six design principles in its characteristics [22, 23, 24], namely modularity, interoperability, etc. These principles are referred to as "design principles" because they contribute to the design or transition process from Industry 3.0 to Industry 4.0.

The main design principles of Industry 4.0

(a) Modularity

The principle of modularity refers to customization and adaptation to different requirements. This principle offers scalability, flexibility, and the ability to upgrade or replace specific components without affecting the entire system.

(b) Interoperability

The principle of interoperability refers to the fact that a cyber-physical system (CPS) comprises intelligent machines and intelligent storage systems and facilities capable of autonomously exchanging information, initiating actions, and controlling each other independently. This includes standardized communication protocols and data formats to ensure compatibility between different components and technologies.

(c) Decentralization

The principle of decentralization refers to the fact that different components and machines can make autonomous decisions based on real-time data, reducing the need for a central controller. Tasks are delegated to a higher level only in cases of failure.

(d) Real-time capability

The principle of real-time capability refers to the ability of systems, manufacturing processes, and intelligent machines to operate and respond to events in real-time or near-real-time.

(e) Virtualization

The principle of virtualization refers to the creation of virtual representations or simulations of physical entities, processes, or systems within the industrial environment. Sensor data is linked to virtual plant models and simulation models. Thus, a virtual copy of the physical world can be created.

(f) Service orientation

The principle of service orientation emphasizes the organization and delivery of functionality as services, marking a shift from merely selling products to offering integrated products and services that provide more value to the customer. This involves the use of SOA architecture (Service-Oriented Architectures).

2.3 Main Pillars of Industry 4.0

In this section, we briefly introduce the main pillars of the Fourth Industrial Revolution, as illustrated in Figure 2.2 from the previous section. Since some key pillars, such as system integration, autonomous robots, and simulation, are more crucial than others for the practical implementation of the presented thesis, we will pay more attention to them.

As we can see in [14, 15, 18, 25], the key pillars of Industry 4.0 do not include AI techniques such as machine learning, deep learning, etc. AI technologies are still a relatively new field in practical applications that have not been sufficiently tested to be incorporated into the pillars that form the Fourth Industrial Revolution. On the other hand, machine learning, deep learning, and other AI techniques can be found as important components in most of the key areas mentioned above.

2.3.1 System Integration

In the Industry 4.0 landscape, system integration plays a crucial role in the transformation of manufacturing processes into interconnected, intelligent, and efficient systems. System integration helps incorporate various technologies within the manufacturing process, including the Internet of Things (IoT), artificial intelligence techniques, cloud computing, robotics, single / multiple purpose machines, and cyber-physical systems. It ensures interoperability and real-time communication between different elements within an intelligent manufacturing environment.

There are three dimensions of system integration within Industry 4.0 [25]: vertical integration, horizontal integration, and end-to-end integration.

(a) Vertical Integration

Vertical integration involves the seamless interconnection and collaboration between different levels of the production process hierarchy. The concept of vertical integration refers to the interconnectedness of the entire industrial enterprise, i.e., all logical levels within the organization, from the production floor to the research department, product management, quality assurance, sales department, and so on. This type of integration refers to flexible and reconfigurable systems within the factory and the extent to which they are fully integrated with each other.

(b) Horizontal Integration

Horizontal integration involves the sharing of data outside the organization, i.e., from suppliers through manufacturers to distribution, to the end customer, and subsequent service. This type of integration facilitates the exchange of information and resources between different entities within a specific phase of the production process.

(c) End-to-End Integration

The goal of end-to-end integration is to create a seamless and interconnected flow of information and processes along the entire value chain, from the initial design phase to the delivery of the final product to the customer. This type of integration applies to all engineering processes throughout the product lifecycle.

Considering the fact that vertical system integration plays a role in supporting seamless interoperability and optimal interconnection between different components within Industry 4.0, which is an integral part of the presented thesis, we will briefly describe the most commonly used Ethernet-based communication protocols.

Ethernet POWERLINK

EtherNet/IP (EtherNet/Industrial Protocol) [26] is one of the most widely used standards that provides users with the tools to deploy standard Ethernet technology (IEEE 802.3 combined with the TCP/IP Suite) in industrial automation applications. The industrial protocol was developed as a result of the collaborative efforts of organizations such as ODVA (Open DeviceNet Vendor Association), ControlNet International, and Rockwell Automation.

EtherNet/IP is a fully compatible industrial protocol according to the IEEE 802.3 standard, which utilizes a standard communication model with a solution at the application layer. Compliance with IEEE Ethernet standards provides a network interface with speeds ranging from 10 Megabits per second (Mbps) to 1 Gigabit per second (Gbps), which can be adjusted to meet user requirements. Additionally, it offers a flexible network architecture that is compatible with commercially available Ethernet installations. Within the EtherNet/IP network, individual nodes are assigned device types with specific characteristics and functions. The assigned device types and the communication network's application layer are created by the Common Industrial Protocol (CIP), which is used in industrial communications such as DeviceNet and ControlNet, employing a producer-consumer communication principle. The use of the CIP protocol achieves interoperability among all networks that support this protocol.

EtherNet/IP

EtherNet/IP (EtherNet/Industrial Protocol) is one of the most widely used standards that provides users with the tools to deploy standard Ethernet technology (IEEE 802.3 combined with the TCP/IP Suite) in industrial automation applications. The industrial protocol was developed as a product of the collaborative efforts of organizations such as ODVA (Open DeviceNet Vendor Association), ControlNet International, and Rockwell Automation.

EtherNet/IP is a fully compatible industrial protocol according to the IEEE 802.3 standard, which utilizes a standard communication model with a solution at the application layer. Compliance with IEEE Ethernet standards provides a network interface with speeds ranging from 10 Megabits per second (Mbps) to 1 Gigabit per second (Gbps), which can be adjusted to meet user requirements. Additionally, it offers a flexible network architecture that is compatible with commercially available Ethernet installations. Within the EtherNet/IP network, individual nodes are assigned device types with specific characteristics and functions. The assigned device types and the communication network's application layer are created by the Common Industrial Protocol (CIP), which is used in industrial communications such as DeviceNet and ControlNet, employing a producer-consumer communication principle. The use of the CIP protocol achieves interoperability among all networks that support this protocol.

PROFINET

PROFINET (Process Field Network) [27, 28] is one of the most widely used industrial communication protocols that plays a crucial role in real-time communication and data exchange in control systems. The industrial protocol was developed and standardized by PROFIBUS & PROFINET International (PI), designed to meet the demanding requirements of modern industrial environments. The PROFINET communication standard can be used to control manufacturing with various elements of industrial automation, such as linking the production process control to collect data from individual sensors capable of processing information at 100 Mb/s, based on ISO/IEC 8802.3.

PROFINET can be divided into the so-called Profinet CBA (Component Based Automation), which is used for modular systems such as Programmable Logic Controllers (PLCs), sensors, etc., and Profinet IO, which is used for distributed field devices. Both Profinet IO and Profinet CBA systems can operate simultaneously on the same network and can be implemented in the same communication station. The industrial PROFINET network can communicate in real-time (RT) at a speed defined by the cycle time.

OPC UA

OPC UA (Open Platform Communications Unified Architecture) [29] is an international standard for secure, reliable, interoperable, and platform-independent industrial communication in the context of the Fourth Industrial Revolution. It provides a solid foundation for modern industrial automation and control systems. The OPC UA industrial standard was published in 2008 and was standardized by the OPC Foundation [30] with the IEC 62541 norm.

It supports different communication patterns, including request/response and publish/subscribe, which can be adapted to different levels of real-time requirements. The real-time performance of OPC UA depends on several factors, including the network infrastructure, system load, and the specific configuration of the OPC UA application. In applications where strict real-time requirements are critical, additional measures may need to be taken, such as implementing Quality of Service (QoS) mechanisms, or optimizing network settings. Although OPC UA supports real-time communication, it is not a real-time protocol. The speed at which OPC UA can communicate depends on several factors, including network conditions, system load, and specific implementation details of OPC UA server and client applications.

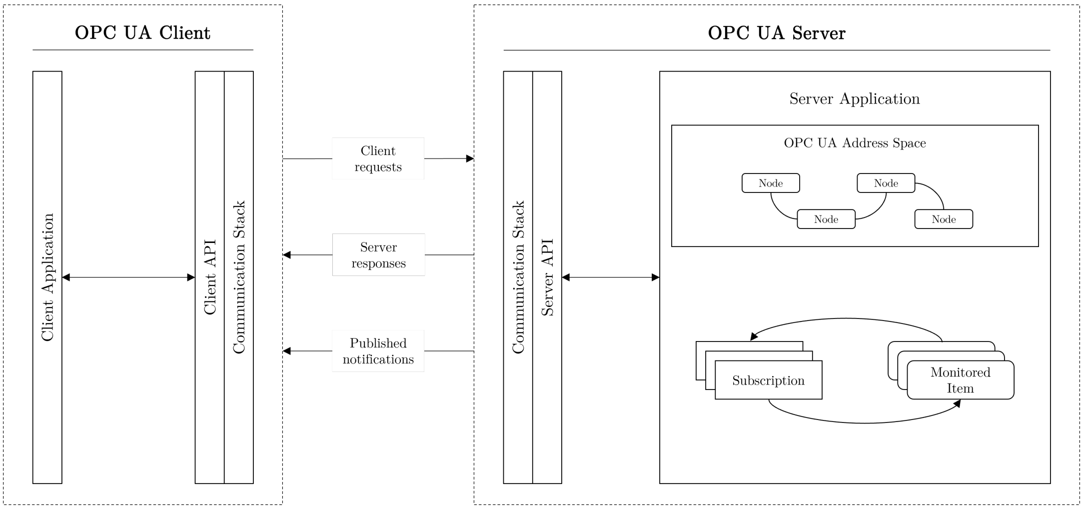

Figure 2.3: Typical client/server architecture of the OPC UA international standard [30].

The OPC UA standard is designed for seamless vertical integration in automation systems and allows any combination of client/server components at different levels of the automation hierarchy for a specific application. It also uses object-oriented techniques to model information, which are represented in the OPC UA address space.

2.3.2 Simulation

Simulation, specifically simulation tools, can be used in production processes to leverage real-time data and imitate the behavior of the physical world in a virtual model, known as the digital twin. The virtual model can include machines, sensors, products, and even people, resulting in increased productivity and quality of production. Using simulation tools, it is possible to optimize machine settings in a virtual environment before implementing them in the physical world. Simulations can be created for both 2D and 3D spaces, allowing us to optimize the production process, improve product quality, and, most importantly, prevent collisions and unexpected situations. The use of advanced simulation, which includes a physical representation of individual parts within the production process, can not only increase the efficiency, safety, and quality of production, but can also prevent production failures.

Digital Twin

The term digital twin has evolved over the years, but with the advent of the Fourth Industrial Revolution, it has become more generalized. In the context of Industry 4.0, a digital twin is a virtual representation of a physical object, system, or process. The virtual model is created through the integration of real-time data from sensors, devices, and other sources connected to the physical entity. The digital twin mimics the physical object in terms of its structure, behavior, and performance, and enables simulation, analysis, and monitoring throughout the entire lifecycle, i.e. from design and manufacturing to operation and maintenance.

According to [31, 32], digital twins can be divided into three subcategories (shown in Figure 2.4) based on the level of data integration. In the following text, all the major subcategories are briefly described: the digital model, the digital shadow, and the digital twin.

(a) Digital Model

A digital model is a representation of a physical object in digital form without automated data flow between the physical and digital objects. In other words, a change in the state of the physical object does not directly affect the digital object and vice versa.

(b) Digital Shadow

A digital shadow is a digital representation of an object with a unidirectional data flow between the physical and digital objects. In other words, a change of state in the real world is reflected in the virtual world, but not vice versa.

(c) Digital Twin

A digital twin is a fully integrated representation of a physical object in both directions, i.e. the data flow between the physical object and the digital object is bidirectional. A change in the state of the physical object leads directly to a change in the state of the digital object, and vice versa.

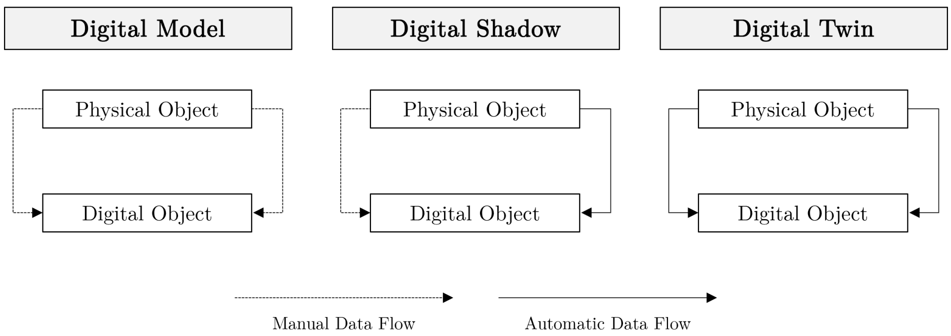

Figure 2.4: Three subcategories based on the data integration level of the digital twin model.

An article by Deloitte [33] provides a conceptual architecture that explains how the digital twin works in general (see Figure 2.5). It consists of five enabling components, including sensors, data, integration, analytics, and actuators, and a six-step process that includes creating, communicating, aggregating, analyzing, gaining insights, and taking action. Sensors and actuators are located in a physical space, while data analysis is performed in a virtual space. Integration technologies must be used to enable seamless data communication between the physical and virtual worlds. The conceptual architecture consists of a sequence of six operational steps based on digital twins: Create, Communicate, Aggregate, Analyze, Insight, and Act. More information about the conceptual architecture can be found in [34] and [35].

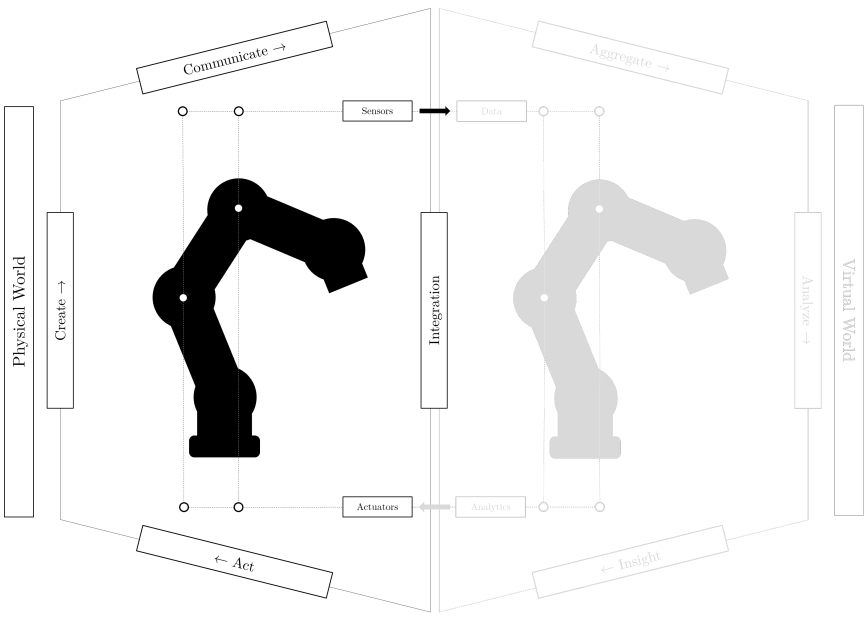

Figure 2.5: A conceptual architecture, consisting of five basic components and a six-step process, illustrates the general functioning of a digital twin.

2.3.3 Autonomous Robots

In the context of Industry 4.0, autonomous robotics is a key component. It is used mainly in the industry to address complex and dangerous tasks that are not suitable for humans. Additionally, autonomous robotics is used to replace stereotypical tasks traditionally performed by humans, thereby preventing the under-utilization of human potential. The use of robots in industrial enterprises is growing exponentially [36], as they significantly increase productivity, efficiency, repeatability, and human safety. Robots as autonomous systems can be used in various industries, such as manufacturing, agriculture, healthcare, etc., to perform a variety of tasks, including intelligent sorting, safe interaction, and visual inspection.

A more detailed description of the principles of autonomous robotics, including the methods used for motion planning and control, is discussed in Section 4.

2.3.4 Other Pillars

In addition to the technological pillars already described, there are other key aspects of the Fourth Industrial Revolution that are necessary to maintain the basic concept.

Cloud Computing

Cloud computing is one of the crucial components of Industry 4.0 because it allows industrial enterprises to manage and visualize data in real-time with minimal interaction with service providers. The restrictions on individual companies are minimized as the industrial revolution promotes increased data sharing between workplaces, driven by the imperative to optimize production. Cloud computing facilitates user mobility, resource conservation, and distributed data analytics to address various network-related issues. It also supports decentralization and intelligent processing of data generated by various Internet of Things (IoT) devices that integrate the physical world into cyberspace.

There are three main cloud service models [18, 37]: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

(a) Infrastructure as a Service (IaaS)

The IaaS cloud service model provides virtualized computing resources over the internet, including virtual machines, storage, and networking.

(b) Platform as a Service (PaaS)

The PaaS cloud service model offers a platform that includes not only the underlying infrastructure, but also the development tools and services needed to build, deploy, and manage applications.

(c) Software as a Service (SaaS)

The SaaS cloud service model provides software applications over the internet on a subscription basis. Users can access these applications through a web browser.

Internet of Things

The Internet of Things (IoT), specifically the Industrial IoT, deals with the industrial communication of interconnected and uniformly addressed objects (i.e. industrial devices) that communicate via standard protocols. In general, the IoT can provide advanced interconnection of systems, services, physical objects, as well as enable communication between industrial devices and the sharing of large amounts of data. By implementing IoT in industrial enterprises, the company can achieve greater integrability, agility, and competitive advantages.

The Internet of Things (IoT) consists of the Internet of Services (IoS), the Internet of Manufacturing Services (IoMs), the Internet of People (IoP), embedded systems, and the Integration of Information and Communication Technology (IICT) [25].

Cybersecurity

Cybersecurity is one of the most crucial components of Industry 4.0, and it can have a significant impact on the business environment due to various attacks. With increasing connectivity and the use of standard communication protocols (see Subsection 2.3.1) that accompany the Fourth Industrial Revolution, the need to protect critical industrial systems and production lines from cybersecurity threats increases dramatically. The key aspect of cybersecurity is secure and reliable communication, along with sophisticated machine access management. Securely connecting both the physical and digital worlds can enhance the quality of information required for planning, optimization, and production.

Closely related to cybersecurity is the concept of the Cyber-Physical System (CPS), which plays a crucial role in Industry 4.0. CPS has been defined as a system in which the physical world is fully integrated with computing, communication, and control systems, collectively referred to as the digital world. The main characteristics of CPS are decentralization and automatic control of the production process. The interconnection of the physical world, the service world, and the digital world can improve the quality of information needed for the planning, optimization, and operation of production systems.

Additive Manufacturing

Additive manufacturing, also defined as 3D printing, is considered the process of manufacturing various parts from computer-created 3D models. It deals with the production of small series of customized products according to customer requirements, offering construction advantages such as complex and lightweight designs. By eliminating the demanding technological preparation of production, shortening the construction process, and utilizing prototypes instead of finished products, the time required to bring the product to market can be reduced. Accurate estimation of the amount of material and simulation of the production process result in optimization of order management. Considering that the production process involves gradually adding material, it is possible to determine its consumption with relative precision.

Additive technologies enable the production of various types of structurally complex parts without the need to reconfigure the machine and also without the need for complex software modification. The production process using additive technologies can adapt the product to the specific needs of the customer.

Augmented Reality

Augmented reality is defined as an interactive technology that allows connecting the physical world with the virtual one, while the virtual world is used as a part of the real environment. This technology in industrial enterprises enables human-machine interaction, machine and equipment maintenance, visual product inspection, etc. In addition to navigation systems, a key aspect of augmented reality is spatial data, which enables the connection of various information to a specific location. By combining a computer-generated environment and physical objects, it could be used in many applications, as creativity knows no limits.

From a technical point of view, the augmented reality system must solve two spatial problems in real-time, which consist of the localization of the user and his spatial vision. Augmented reality uses a combination of sensors (gyroscope, accelerometer, etc.) and computationally intensive machine vision algorithms based on artificial intelligence techniques.

Big Data

The concept of big data applies to large, diverse, and complex amounts of data that influence the organizational decision-making processes of industrial enterprises. According to Forrester's definition [38], Big Data consists of four dimensions: (1) the amount of data, (2) the variety of data, (3) the speed of generation of new data and analysis, and (4) the value of data. The volume of data in the industry is growing exponentially, thereby increasing the potential amount of usable information. However, the ability to collect and comprehensively evaluate information is necessary for further development. Therefore, the increase in the level of data and the improvements in technological capabilities accelerate the increase in productivity and innovation.

In the context of Industry 4.0, big data analytics is beneficial in several areas of an industrial enterprise. It aids in predictive maintenance, improving equipment service, optimizing production quality, and, last but not least, saving energy. Analysis of previously recorded data is used to detect threats that have occurred in various production processes and to predict new problems that may arise, in order to prevent them.

2.4 Forecasting the Future Trends of the Industry

Predicting the future of any industry involves a degree of uncertainty, as it depends on a variety of factors, including technological advances, economic conditions, regulatory changes, consumer preferences, and, last but not least, global crises. In this section, we present the key trends and research areas that are likely to shape the future of the industry.

Although the era of Industry 4.0 is still in its development phase, some scientific publications (see [39, 40, 41]) are already predicting the next step that will lead to improvements such as sustainability, modularity, efficiency, human-centered solutions, and universality. The next phase can be the next industrial revolution, referred to as Industry 5.0, or the evolution of the existing Industry 4.0 with the enhancement of its main pillars.

One of the most discussed terms in the future development of the industrial segment will undoubtedly be artificial intelligence (AI) techniques, which will be integrated into the improvement of individual technologies within the overall concept. Autonomous robots will be even more important due to the fact that it will be necessary to increase productivity without removing human workers from manufacturing [42, 43]. The use of advanced physical simulations [44], which will contain a fully integrated representation of both the physical and virtual worlds, will become a crucial part. Last but not least, supply chain assessment and optimization in smart and sustainable manufacturing processes, along with the transformation of data driven by IoT, Big Data, and the use of cloud computing.

The next revolution, or evolution, can be more flexible, scalable, and adaptable to humans through the use of improved technology. Efficient and modular system integration, higher levels of robotic automation with a focus on autonomy, the power of human creativity, and more, will lead to increased productivity.

CHAPTER 3

Kinematics

Kinematics in robotic manipulators is a fundamental aspect that deals with the study of motion principles, without considering forces and torques [45, 46]. It covers the transformation of objects based on the position and orientation of various components within a robot's mechanical structure. In general, the kinematic description is a geometric representation of the robot mechanism defined by specific conventions known as the Denavit-Hartenberg conventions.

The problem of manipulator kinematics is typically divided into two categories [47]: forward kinematics and inverse kinematics. Forward kinematics establishes the relationship between joint variables and the end-effector's homogeneous transformation matrix (Eq. 3.18), while inverse kinematics determines the joint configurations necessary to achieve a desired end-effector pose (Eq. 3.19).

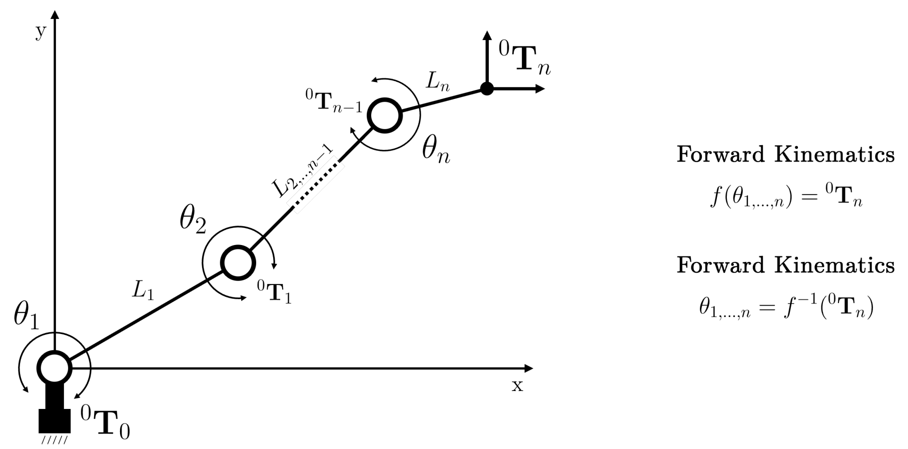

Figure 3.1: A simple representation of an n-line robotic manipulator in two-dimensional space, illustrating the basic concept of kinematics computation.

The chapter begins by delving into the fundamentals of coordinate transformations (Section 3.1), clarifying their role in representing position (Subsection 3.1.1) and rotation (Subsection 3.1.2) within Euclidean space. This discussion leads to an exploration of homogeneous transformations (Subsection 3.1.3), a pivotal aspect in the study of robotic manipulator kinematics. Following this introduction, the chapter provides a concise overview of the general representation of robotic manipulators (Section 3.2). Within the context of kinematics, the geometric representation of robotic mechanisms is elucidated, using the Denavit-Hartenberger convention (Section 3.3). The culminating section of the chapter focuses on the computation of forward (Section 3.4) and inverse (Section 3.5) kinematic problems for serial-chain manipulators. The discussion of the inverse kinematic problem encompasses both the closed-form solution and the numerical solution.

3.1 Coordinate Transformations

The study of kinematics for robotic manipulators or objects generally focuses on characterizing motions without considering forces and torques. Specifically, the dynamics of the robotic system is neglected in this area of research. In this context, coordinate transformations play a central role [45, 48, 49], referring to mathematical techniques used to describe the position and orientation of rigid bodies in different coordinate frames. These transformations are essential for accurate modeling and control of robot motions because they facilitate the conversion of coordinates from one reference frame to another. This feature provides a method for analyzing and predicting the robot's spatial configuration.

In the context of Euclidean space, the pose of an object includes its position and orientation. For a rigid object, the required pose coordinates are defined by at least six parameters. Many representations of spatial pose use sets of redundant coordinates, where there are auxiliary relations between the coordinates. The number of independent auxiliary relations is the difference between the number of coordinates in the set and the six. The poses of an object are described with respect to the coordinate reference frame, which consists of an origin of the frame i, denoted O_i, and a set of three orthogonal basis vectors, denoted $(\vec{x}_i, \vec{y}_i, \vec{z}_i)$, all fixed within a specific body [45]. The pose of a body is consistently expressed relative to another body, conveyed as the pose of one coordinate frame relative to another.

This section is a compilation of content from various sources, including both scientific articles (see [50, 51, 52, 53]) and book publications (see [45, 48, 49, 54]).

3.1.1 Representation of Position

In a three-dimensional Euclidean space, the position, also called translation, of the origin of the coordinate frame i relative to the coordinate frame j, can be expressed by the vector ${}^{j}\mathbf{p}_i \in \mathbb{R}^3$.

$$^{j}\mathbf{p}_i = \begin{bmatrix} {}^{j}p_i^x \\ {}^{j}p_i^y \\ {}^{j}p_i^z \end{bmatrix}. \tag{3.1}$$

The individual components of this vector are the Cartesian coordinates of the origin O_i in the coordinate frame j, which represent projections of the vector ${}^{j}\mathbf{p}_i$ onto the corresponding axes x, y, and z.

3.1.2 Representation of Rotation

The concept of expressing a rotation in three-dimensional space as a mathematical formulation is considerably broader than the straightforward representation of position. According to Euler's rotation theorem, any arbitrary rotation of a rigid body in a three-dimensional space can be represented as a single rotation about a fixed axis. Based on this statement, we can conclude that any rotation can be described by at least three parameters.

There are several ways of expressing rotation in three-dimensional space, some involving more than three parameters, although all ways of expressing rotation are described by three degrees of freedom. In the field of robotic manipulators, rotation matrices, Euler angles, and quaternions are the most commonly used. Each of these representations will be briefly described in the following sections.

Rotation Matrices

In a three-dimensional Euclidean space, the orientation of the coordinate frame i relative to the coordinate frame j can be denoted by expressing the basis vectors $(\vec{x}_i, \vec{y}_i, \vec{z}_i)$ of the frame i in terms of the basis vectors $(\vec{x}_j, \vec{y}_j, \vec{z}_j)$ of the frame j. In this way, we obtain a so-called rotation matrix ${}^{j}\mathbf{R}_i$, in which each component is the scalar product of the basis vectors of both the i and j coordinate frames, denoted $({}^{j}\vec{x}_j, {}^{j}\vec{y}_j, {}^{j}\vec{z}_j)$.

$${}^{j}\mathbf{R}_i = \begin{bmatrix} \vec{x}_i \cdot \vec{x}_j & \vec{y}_i \cdot \vec{x}_j & \vec{z}_i \cdot \vec{x}_j \\ \vec{x}_i \cdot \vec{y}_j & \vec{y}_i \cdot \vec{y}_j & \vec{z}_i \cdot \vec{y}_j \\ \vec{x}_i \cdot \vec{z}_j & \vec{y}_i \cdot \vec{z}_j & \vec{z}_i \cdot \vec{z}_j \end{bmatrix}. \tag{3.2}$$

The basis vectors are equal to unit vectors, and the scalar product of any two unit vectors is the cosine of the angle between these vectors. The components of this angle are called directional cosines.

The rotation of frame i about the $\vec{x}_j$ axis through an angle θ can be expressed by the rotation matrix with the values

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}, \tag{3.3}$$

while the same rotation about the $\vec{y}_j$ axis can be expressed by the rotation matrix with the values

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \tag{3.4}$$

lastly, the $\vec{z}_j$ axis can be expressed by the rotation matrix with the values.

$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.5}$$

The general three-dimensional rotation is achieved by combining the above rotation matrices into one matrix through matrix multiplication. Since matrix multiplication is non-commutative, it is important to consider the order of the rotations, as follows

$$\mathbf{R}_x\mathbf{R}_y\mathbf{R}_z \neq \mathbf{R}_z\mathbf{R}_y\mathbf{R}_x. \tag{3.6}$$

This statement implies that rotation about the x, y, and z axes does not produce the same rotation as rotation about the z, y, and x axes. This property is consistent with the fact that rotations are also non-commutative.

The rotation matrix ${}^j\mathbf{R}_i$ contains nine elements (see Eq. 3.2), while only three parameters are needed to define the orientation of an object in a three-dimensional Euclidean space. Therefore, there are six additional auxiliary relations between the elements of the matrix. Since the basis vectors of both coordinate frames i and j are mutually orthonormal, the columns of ${}^j\mathbf{R}_i$ formed by the scalar products of these vectors are also mutually orthonormal. A matrix composed of mutually orthonormal vectors is called an orthogonal matrix and has the property that the inverse matrix of rotation is equal to the transpose matrix of rotation.

$${}^j\mathbf{R}_i^{-1} = {}^j\mathbf{R}_i^T. \tag{3.7}$$

This property provides the six auxiliary relations. Three of them require that the column vectors have unit length, and three require that the column vectors be mutually orthogonal.

Euler Angles

The orientation of the coordinate frame i relative to the coordinate frame j can be represented as a vector of three independent coordinates (α, β, γ), commonly known as Euler angles. These coordinates, employed in the Euler angle representation, describe three consecutive rotations about an axis of a moving coordinate frame. As the order of rotation about the axes is essential, the specification of rotations for Euler angles must always be accompanied by an appropriate convention, providing a detailed description of the axes.

Assuming that the moving frame i and the fixed frame j are initially coincident, the Euler angles of the Z-Y-X convention can be obtained by three successive rotations in the order α, β, γ, corresponding to rotations about the z, y, and x axes of the frame i. In terms of the basic rotation matrices, the resulting rotational transformation of the Z-Y-X convention can be formed as the product of

$$\begin{aligned} {}^{j}\mathbf{R}_i(\alpha,\beta,\gamma) &= \mathbf{R}_z(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_x(\gamma) \\ &= \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{bmatrix}, \end{aligned} \tag{3.8}$$

where c and s are abbreviations of the cosine and sine functions.

However, significant disadvantages of Euler angles, such as susceptibility to gimbal lock and low accuracy in integrating incremental changes over time, have led researchers to use alternative rotation representations, such as quaternions, to effectively represent rotations.

Quaternions

The quaternion representation of orientation, described by W. R. Hamilton in the 19th century, is a mathematical concept used in three-dimensional geometry. Unlike traditional representations such as Euler angles, a quaternion can be expressed as a four-dimensional vector in the form

$$\mathbf{Q} = q_w + iq_x + jq_y + kq_z, \tag{3.9}$$

where q_w, q_x, q_y, $q_z \in \mathbb{R}$, are the four constituents of the quaternion $\mathbf{Q}$, sometimes referred to as Euler's parameters, denoting arbitrary real quantities: positive, negative, or zero. The symbols i, j, k are operators or simply symbols that represent three imaginary quantities.

A quaternion $\mathbf{Q}$, especially a unit quaternion $\|\mathbf{Q}\| = 1$, is typically written as a combination of a scalar and a vector, where the scalar part contains information about the angle of rotation, and the vector part defines the axis of rotation. Quaternions provide a compact and efficient way to perform three-dimensional rotations, avoiding certain mathematical ambiguities and gimbal lock issues associated with other rotation representations.

In terms of the basic rotation matrices, the resulting rotational transformation of a unit quaternion can be formulated as follows

$${}^{j}\mathbf{R}_i = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_w q_z) & 2(q_w q_y + q_x q_z) \\ 2(q_w q_z + q_x q_y) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_w q_x) \\ 2(q_x q_z - q_w q_y) & 2(q_w q_x + q_y q_z) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix}. \tag{3.10}$$

The corresponding rotation matrix ${}^{j}\mathbf{R}_i$ (see Eq. 3.10) is obtained as a rotation about the unit axis in the direction of the vector part of the unit quaternion $(iq_x + jq_y + kq_z)$ by an angle of rotation $2\cos^{-1}(q_w)$.

3.1.3 Homogeneous Transformations

The previous sections addressed the representation of position (Sect. 3.1.1) and orientation (Sect. 3.1.2) separately. However, when computing the pose representation in Euclidean space, it is often convenient to work with a combined representation known as the homogeneous transformation. In this representation, the position vector ${}^{j}\mathbf{p}_i$ and the rotation matrix ${}^{j}\mathbf{R}_i$ are combined in a compact notation. It follows from this statement that the homogeneous transformation allows the combination of both algebraic operations of translation and rotation within a fixed and consistent homogeneous transformation matrix ${}^{j}\mathbf{T}_i$.

$$ {}^{j}\mathbf{T}_i = \begin{bmatrix} {}^{j}\mathbf{R}_i & {}^{j}\mathbf{p}_i \\ \mathbf{0}^T & 1 \end{bmatrix}. \tag{3.11} $$

The matrix ${}^{j}\mathbf{T}_i$ transforms the vectors from the coordinate frame i to the coordinate frame j. On the other hand, the inverse ${}^{j}\mathbf{T}_i^{-1}$ transforms the vectors from the coordinate frame j to the coordinate frame i.

$$ {}^{j}\mathbf{T}_i^{-1} = \begin{bmatrix} {}^{j}\mathbf{R}_i^T & -{}^{j}\mathbf{R}_i^{T}\,{}^{j}\mathbf{p}_i \\ \mathbf{0}^T & 1 \end{bmatrix}. \tag{3.12} $$

The composition of homogeneous 4x4 transformation matrices is achieved by a simple matrix multiplication between the frames 0 and n, which is built from multiple piecewise transformations as described in

$$ {}^{0}\mathbf{T}_n = {}^{0}\mathbf{T}_1\,{}^{1}\mathbf{T}_2 \ldots {}^{n-1}\mathbf{T}_n. \tag{3.13} $$

This equation is suitable for describing the complex kinematic transformation along a serial chain of bodies connected by joints, which is a typical model for robotic manipulators.

The homogeneous transformation of a rotation about a particular axis can be denoted as **Rot**, so the rotation of α about the x axis can be expressed in the form

$$ \mathbf{Rot}_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.14} $$

while, the homogeneous transformation of a translation along a particular axis can be denoted as **Trans**, so the translation of t along the x axis can be expressed in the form

$$ \mathbf{Trans}_{x,t} = \begin{bmatrix} 1 & 0 & 0 & t \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.15} $$

Although homogeneous transformation matrices technically contain sixteen elements, four of them are defined as a vector of zeros or identity values to ensure the proper composition and representation of transformations. The remaining elements are composed of a rotation matrix and a position vector.

3.2 General Representation of Robotic Manipulators

Robotic manipulators generally consist of links (rigid bodies) connected by joints that allow controlled (linear) or free relative motion, forming a so-called kinematic chain [55] (see Fig. 3.2). The joints are usually represented as revolute (rotational motion) or prismatic (translational motion) and serve as a connection between two links. Equally important components in the representation of a robotic manipulator are the base, which is typically connected to the ground and directly linked to the world coordinate system, and the end-effector, which is the last link of the robotic structure and can be used to hold tools.

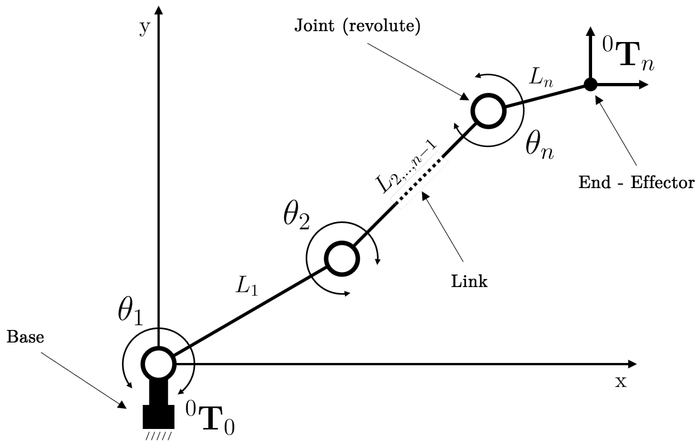

Figure 3.2: A simple representation of an n-link robotic manipulator in two-dimensional space, which describes the basic components of the robotic structure.

3.2.1 Degree of Freedom

The Degree of Freedom (DoF) refers to the number of independent motions or variables through which a rigid object can move within a defined space. The number of DoF is, therefore, equal to the dimension of the configuration space. A rigid object in two-dimensional space has 3 DoF (two for translation and one for rotation), while in three-dimensional space it has 6 DoF (three for translation and three for rotation), as we can see in Fig. 3.3.

In robotics, the concept of degrees of freedom (DoF) is critical to understanding and designing robotic systems, particularly robotic arms and manipulators. DoF refers to the number of independent motions a manipulator can perform, determined by the total number of joints and the constraints imposed by the robot's design [46].

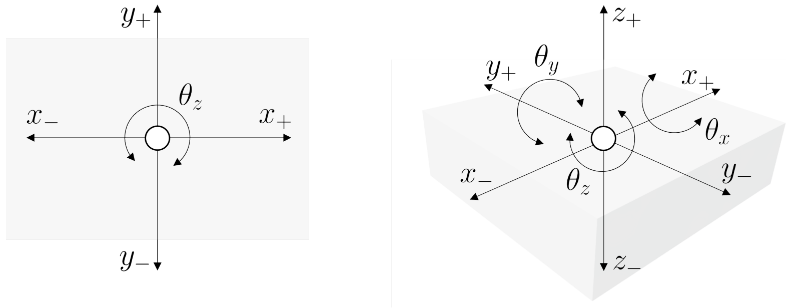

Figure 3.3: The number of degrees of freedom of the rigid body in two-dimensional (left) and three-dimensional (right) space.

Based on this statement, we can categorize the degrees of freedom in a robotic system into two main parts: translational and rotational. The translational part involves linear motion along a specific axis, allowing the robot to move in a straight line. On the other hand, the rotational part involves motion about a specific axis, allowing the robot to change its orientation.

A higher degree-of-freedom increases the redundancy of available variable joint configurations, leading to mathematical challenges such as singularities and suboptimal extrema [56]. Robotic manipulators with fewer DoFs are considered under-articulated, while those with more than six DoFs are described as over-articulated, or more precisely, kinematically redundant.

3.2.2 Types of Joints

Joints are crucial components of the robotic structure that enable robots to achieve movement and flexibility. They are designed to mimic the range of motion found in natural limbs, providing the robot with the ability to perform diverse tasks. There are various types of robot joints available to define either translational or rotational motion of the individual segments. The most common types of robot joints include revolute, prismatic, helical, cylindrical, universal, and spherical [46].

The revolute joint, also known as a hinge joint, enables rotational motion about a specific axis. In contrast, the prismatic joint, or linear joint, facilitates translational motion along a designated axis. The helical joint, commonly referred to as a screw joint, allows simultaneous rotation and translation along a screw axis. All of these types of joints provide one degree-of-freedom.

Moving to joints with two degrees of freedom, the cylindrical joint permits independent translation and rotation about a single fixed axis. Similarly, the universal joint, composed of two revolute joints with orthogonal axes, also boasts two degrees of freedom. This joint is frequently employed to transmit motion between non-collinear axes, showcasing its versatility.

Last but not least, the spherical joint, often known as a ball-and-socket joint, stands out with its three degrees of freedom, functioning much like the human shoulder joint.

This design allows for a wide range of motion and finds applications in various mechanical systems.

3.2.3 Workspace

The workspace of a general-purpose robotic manipulator refers to the total volume covered by the end-effector during all possible motions. This space is determined by the geometry of the manipulator and the movement limits of each joint. The workspace is commonly categorized into two types [45, 55]: reachable workspace, representing the total set of points accessible by the end-effector, and dexterous workspace, which is a subset of these points allowing arbitrary end-effector orientations.

Most serial-chain robotic manipulators exhibit regional and orientational structures [45]. The regional structure, formed by inner joints, determines the end-effector's position in space, while the orientational structure, comprised of outer joints, determines its orientation. The volume of the regional workspace can be computed using the known geometry and joint motion limits of the serial-chain robot manipulator [45] or through a simplified version employing the Monte Carlo method [57].

3.3 Denavit-Hartenberg Convention

The geometric representation of a robotic mechanism is defined by attaching coordinate frames to a specific link within the structure. Although the coordinate frames can be positioned anywhere, it is advantageous for both clarity and computational efficiency to adhere to a mathematical convention for frame placement within the links. Jacques Denavit and Richard Hartenberg introduced this fundamental convention in 1955 [58], known as the Denavit-Hartenberg (DH) standard method. This method was subsequently modified by Etienne Dombre and Wisama Khalil [59] and is referred to as the Denavit-Hartenberg (DH) modified method. While the standard method exhibits ambiguities when applied to closed or tree-structured robots, the modified method allows a unified description of complex and serial structures in articulated mechanical systems [60]. It is important to note that, despite the differences, the transformation matrices derived from both conventions will yield the same end-effector position and orientation.

In both forms, determining the relative location of one coordinate frame to another requires only four parameters, in contrast to the six parameters needed to describe any transformation in Euclidean space. The parameters, based on the Denavit-Hartenberg (DH) convention, include two link parameters: the link length a_i and the link twist α_i, along with two joint parameters: the joint offset d_i and the joint angle θ_i.

The reduction in the number of parameters is achieved by strategically placing the origins and axes of the coordinate frames. Specifically, the x axis of one frame is positioned to intersect both perpendicularly and tangentially with the z axis of the subsequent coordinate frame.

In the standard DH convention, each homogeneous transformation ${}^{i-1}\mathbf{T}_i$ is represented as a product of four basic transformations about particular axes

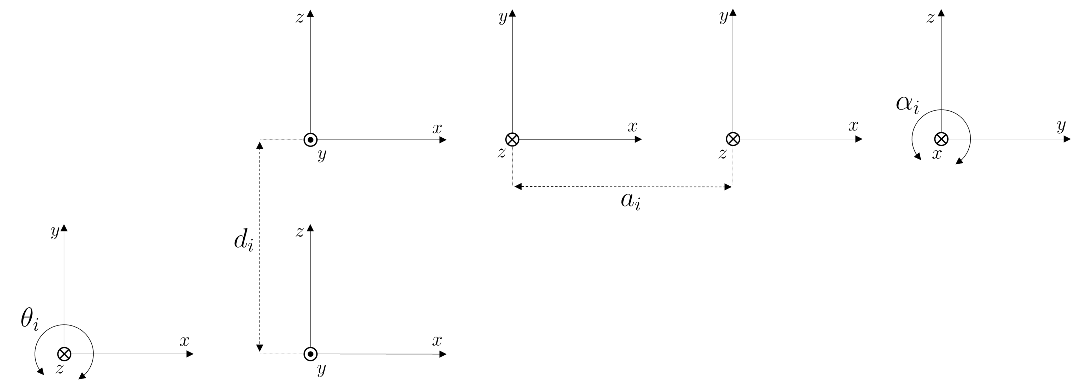

Figure 3.4: Illustration of the Denavit-Hartenberg convention defined by the four parameters a_i, α_i, d_i, and θ_i.

$$\begin{aligned} {}^{i-1}\mathbf{T}_i &= \mathbf{Rot}_{x,\alpha_i}\mathbf{Trans}_{x,a_i}\mathbf{Rot}_{z,\theta_i}\mathbf{Trans}_{z,d_i} \\ &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned} \tag{3.16}$$

and in the modified DH convention, similar to the previous case, but with the modified transformation rules

$$\begin{aligned} {}^{i-1}\mathbf{T}_i &= \mathbf{Rot}_{x,\alpha_{i-1}}\mathbf{Trans}_{x,a_{i-1}}\mathbf{Rot}_{z,\theta_i}\mathbf{Trans}_{z,d_i} \\ &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & a_{i-1} \\ s_{\theta_i}c_{\alpha_{i-1}} & c_{\theta_i}c_{\alpha_{i-1}} & -s_{\alpha_{i-1}} & -d_i s_{\alpha_{i-1}} \\ s_{\theta_i}s_{\alpha_{i-1}} & c_{\theta_i}s_{\alpha_{i-1}} & c_{\alpha_{i-1}} & d_i c_{\alpha_{i-1}} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \tag{3.17}$$

The matrix ${}^{i-1}\mathbf{T}_i$ is a function of a single variable, with three of its four parameters determined by the joint type and geometry. In the case of a prismatic joint aligned along the z axis, only the joint offset d_i varies, while all other parameters remain constant. Conversely, for a revolute joint revolving around the z axis, only the joint angle θ_i is variable, and the remaining parameters are fixed.

3.4 Forward Kinematics

The computation of the forward kinematics problem involves determining the homogeneous transformation matrix ${}^0\mathbf{T}n$ for a serial-chain manipulator, represented by Equation 3.18. This matrix describes the end-effector's position and orientation relative to the base. The computation utilizes joint variables $\theta 1, \ldots, n$ in a recursive process, where coordinate transformations are performed along each link, leading from the base frame to the end-effector frame.

It is crucial to note that the frame attached to the end-effector is commonly known as the tool frame. This frame, which maintains a constant displacement in both position and orientation, plays a significant role in the kinematic analysis. Importantly, the tool transformation matrix must be added as the last element in the chain.

$$f(\theta_{1,\ldots,n}) = {}^{0}\mathbf{T}_{n}. \tag{3.18}$$

The forward kinematics problem is critical to the development of robot coordination algorithms because the orientation of joints is typically measured by sensors integrated into the design of a particular joint. In practice, solving the forward kinematics problem involves computing the transformation between the tool and the station frames [45]. This process is straightforward for a serial chain because the transformation describing the pose of the end-effector relative to the base is obtained by simply concatenating transformations between frames fixed in adjacent links of the chain, as explained using the Denavit-Hartenberger convention in Sect. 3.3. It is important to note that the calculation of forward kinematics always provides a unique solution.

3.5 Inverse Kinematics

The computation of the inverse kinematics problem, expressed in Eq. (3.19), for a serial-chain manipulator involves identifying joint variables denoted as $\theta_{1,\ldots,n}$. These variables correspond to a given homogeneous transformation matrix ${}^{0}\mathbf{T}_{n}$, representing the end-effector's position relative to the base. Unlike the forward kinematics problem, determining this set of joint parameters is not straightforward. Equation (3.13) underscores the necessity of solving systems of non-linear equations to address the challenges posed by the inverse kinematics problem.

$$\theta_{1,\ldots,n} = f^{-1}({}^{0}\mathbf{T}_{n}). \tag{3.19}$$

Addressing the characterization of an identical Cartesian target with different joint configurations poses a challenging issue, potentially resulting in zero to infinite solutions. While algebraic solutions can be obtained for simpler geometries, their applicability diminishes for more complex cases. Consequently, numerous methods are based on numerical solutions [61, 62, 63, 64]. Although analytical solutions may prove useful in specific cases [65, 66], a numerical approach generally offers greater universality. However, it is crucial to recognize that numerical solutions come with challenges, including suboptimal extremes, computational cost, and the potential to encounter singularities.

It is important to highlight that singularities present a significant challenge in the field of robot kinematics. Singular points lead to an immediate reduction in degrees of freedom during the resolution of motion postures. To address this challenge, a strategic approach involves transforming the task into a workspace region where configurations with singularities are avoided [67, 68]. This transformation helps maintain a higher level of flexibility in the robot's movements. For instance, this may involve adjusting the angles of the joints. Such strategies can effectively mitigate the effects of singularities on robot kinematics.

3.5.1 Closed-Form Solutions

The methods for obtaining closed-form solutions in inverse kinematics problems strongly depend on the geometry of robotic mechanisms and are, therefore, not sufficiently general. However, the solutions obtained are exact and can provide all existing joint configurations for a particular reachable Cartesian pose. Although these methods have a lower computational cost, they are typically only available for specific geometries, primarily for non-redundant manipulators with fewer degrees of freedom. The complexity rapidly increases with each additional degree-of-freedom. The necessary conditions for a closed-form solution of a non-redundant six-degree-of-freedom robotic mechanism are as follows [69]:

(a) Three consecutive revolute joint axes intersect at a common point, as in a spherical wrist.

(b) Three consecutive revolute joint axes are parallel.

The most effective methods for obtaining closed-form solutions involve analytical techniques that utilize specific geometric features of particular mechanisms. Generally, closed-form solutions can only be obtained for six-degree-of-freedom systems with a special kinematic structure characterized by a large number of geometric parameters, as described in Section 3.3.

3.5.2 Numerical Solutions

The numerical methods for solving inverse kinematics problems are not dependent on the geometry of the robotic mechanisms, unlike the closed-form solutions discussed in the previous section (see Sect. 3.5.1). Therefore, they can be applied to any kinematic structure, regardless of the number of degrees-of-freedom. Since numerical methods require the approximation of derivatives, they are computationally more expensive than methods based on closed-form solutions. However, it is possible for them to operate directly in a joint space. This simplifies the use of joint limits and allows for greater flexibility in including additional objectives.

In general, the numerical solution of the discussed problem provides the relationship between the joint velocities and the corresponding end-effector linear and angular velocity. In other words, it is desirable to express the linear velocity of the end-effector $\dot{\mathbf{p}}_e$ and the angular velocity of the end-effector $\dot{\omega}_e$ as functions of the joint velocities $\dot{\theta}$.

$$\dot{\mathbf{p}}_e = \mathbf{J}_P(\theta)\dot{\theta}, \tag{3.20}$$

$$\dot{\omega}_e = \mathbf{J}_O(\theta)\dot{\theta}, \tag{3.21}$$

where $\mathbf{J}_P \in \mathbb{R}^{3\times n}$ represents the translational part of the manipulator that relates the contribution of the joint velocities to the linear velocity of the end effector, while $\mathbf{J}_O \in \mathbb{R}^{3\times n}$ represents the rotational part of the manipulator that relates the contribution of the joint velocities to the angular velocity of the end effector.

In compact form, equations (3.20) and (3.21) can be written as

$$\mathbf{v}_e = \begin{bmatrix} \dot{\mathbf{p}}_e \\ \dot{\omega}_e \end{bmatrix} = J(\theta)\dot{\theta}, \tag{3.22}$$

which represents the general numerical kinematics equation. The vector $\mathbf{v}_e$ is the spatial velocity of the end-effector expressed in an arbitrary frame, $\dot{\theta}$ is an n-dimensional vector composed of the joint velocities, and $J(\theta) \in \mathbb{R}^{6\times n}$ is the so-called Jacobian matrix of non-linear functions of θ, expressed relative to the same coordinate frame as the spatial velocity $\mathbf{v}_e$. The Jacobian matrix is a matrix of first-order partial derivatives of each joint variable, providing a linear approximation of the resulting end-effector velocities in Cartesian space.

$$J(\theta) = \begin{bmatrix} \mathbf{J}_P \\ \mathbf{J}_O \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{P_1} & \cdots & \mathbf{J}_{P_n} \\ \vdots & \ddots & \vdots \\ \mathbf{J}_{O_1} & \cdots & \mathbf{J}_{O_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial p_e^x}{\partial \theta_1} & \cdots & \frac{\partial p_e^x}{\partial \theta_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \omega_e^z}{\partial \theta_1} & \cdots & \frac{\partial \omega_e^z}{\partial \theta_n} \end{bmatrix}. \tag{3.23}$$

It can be observed from Equation (3.23) that the i-th column of the Jacobian matrix corresponds to the i-th joint of the robot manipulator. Depending on whether the i-th joint is prismatic or revolute, it takes one of two forms [55, 47], as follows

$$J_i(\theta) = \begin{bmatrix} \mathbf{J}_{P_i} \\ \mathbf{J}_{O_i} \end{bmatrix} = \begin{cases} \begin{bmatrix} p^z_{e_{i-1}} \\ 0 \end{bmatrix} \rightarrow \text{prismatic joint} \\ \begin{bmatrix} p^z_{e_{i-1}} \times (\mathbf{p}_{e_i} - \mathbf{p}_{e_{i-1}}) \\ p^z_{e_{i-1}} \end{bmatrix} \rightarrow \text{revolute joint} \end{cases} \tag{3.24}$$

To solve the linear system of equations in the joint velocities obtained by the decomposition, expressed in Eq. (3.22), into its component equations when $\mathbf{v}_e$ is known, it is necessary to invert the Jacobian matrix as follows

$$\dot{\theta} = J(\theta)^{-1}\mathbf{v}_e. \tag{3.25}$$

However, the numerical approach is only feasible if the Jacobian matrix J is invertible. This requires the matrix to be square-shaped and have a non-zero determinant. It is clear from that statement that redundant manipulators will not be invertible because the Jacobian matrix will not be square. Although the robotic mechanism is not in a singularity configuration, but only in a near-singular configuration, the values of J^{-1} become very large, leading to very high joint space velocities and instability in the numerical solution approach. Various methods are employed to address the aforementioned shortcomings, specifically to reduce instability and enhance the usability of redundant robotic structures. These methods will be discussed further.

Jacobian Transpose Method

The numerical computation of the inverse kinematics problem can be achieved using the Jacobian transpose method, which involves using the transpose of the Jacobian matrix instead of its inverse [70, 71]. This method modifies the general equation (3.25) of numerical inverse kinematics solutions as follows

$$\dot{\theta} = \alpha J(\theta)^T \mathbf{v}_e, \tag{3.26}$$

for some appropriate scalar α, to minimize the new value of the vector after the update. The method of expressing the parameter α was described in [72], where the relationship between the desired change $\mathbf{v}_e$ and an approximated change of the end-effector, defined as $J(\theta)J(\theta)^T\mathbf{v}_e$, was used. Assuming that $J(\theta)J(\theta)^T\mathbf{v}_e$ is the real change, α is chosen to be as close as possible to $\mathbf{v}_e$ as follows

$$\alpha = \frac{\left\langle \mathbf{v}_e, J(\theta)J(\theta)^T\mathbf{v}_e \right\rangle}{\left\langle J(\theta)J(\theta)^T\mathbf{v}_e, J(\theta)J(\theta)^T\mathbf{v}_e \right\rangle}. \tag{3.27}$$

It is important to note that the transpose method generates smooth motion but often suffers from slower convergence.

Newton-Raphson Method

The numerical computation of the inverse kinematic problem is often solved using the Newton-Raphson method [46]. A common implementation involves taking advantage of the Moore-Penrose pseudoinverse [73], leading to what we will refer to as the pseudoinverse method. This method is defined by the equation as follows

$$\dot{\theta} = J(\theta)^{\dagger}\mathbf{v}_e, \tag{3.28}$$

where $J(\theta)^{\dagger} = J(\theta)^T(J(\theta)J(\theta)^T)^{-1}$ represents the pseudoinverse of $J(\theta)$ with the same dimensions. This definition holds for all matrices $J(\theta)$, even those that are not square or do not have full row rank.

The pseudoinverse method is a technique used to enhance the stability and usability of redundant robotic structures where ordinary inversion is not feasible [72]. In simple terms, it helps address the challenges posed by having more degrees of freedom than necessary. However, it is important to note that the pseudoinverse method can exhibit instability in nearly singular configurations. In these situations, the system may experience jerky movements, lacking the desired smoothness and stability. Additionally, it may encounter difficulties in finding feasible solutions due to the peculiarities of these configurations.

Gauss-Newton Method

The Gauss-Newton method is a numerical approach employed to enhance the solvability and compatibility of the inverse kinematics problem, particularly in the context of redundant robotic manipulators [74, 63]. This method utilizes the Jacobian matrix $J(\theta)$ to approximate the Hessian matrix $\mathbf{H}$. The approximation is necessary because computing the exact Hessian matrix can be computationally expensive, and often, it is not necessary to obtain the exact Hessian.

$$\dot{\theta} = (\mathbf{H})^{-1} J(\theta)^T \mathbf{W}_e \mathbf{v}_e, \tag{3.29}$$

where $\mathbf{W}_e = \mathrm{diag}(\mathbf{w_e})(\mathbf{w_e} \in (\mathbb{R}^+)^n)$ is a diagonal weighted matrix that prefers the corresponding error term, and $\mathbf{H} = J(\theta)^T \mathbf{W}_e J(\theta)$ describes the approximate Hessian matrix.

However, equation (3.29) is only valid if $\mathbf{H}$ is invertible, as described in [75]. This issue can be resolved, as mentioned earlier, by using the Moore-Penrose pseudoinverse applied to the aforementioned part ($\mathbf{H}^{-1} \rightarrow \mathbf{H}^\dagger$).

Levenberg-Marquardt Method

The Levenberg-Marquardt method, also known as the Damped Least Squares (DLS) method, is designed to address issues associated with the pseudoinverse method, particularly when dealing with near singularities in the Jacobian matrix, denoted as $J(\theta)$. It provides a numerically stable approach for selecting the parameter $\dot{\theta}$.

$$\dot{\theta} = (\mathbf{A})^{-1} J(\theta)^T \mathbf{W}_e \mathbf{v}_e, \tag{3.30}$$

where $\mathbf{W}_e = \mathrm{diag}(\mathbf{w}_e)$ $(\mathbf{w}_e \in (\mathbb{R}^+)^n)$ is a diagonal weighted matrix that prefers the corresponding error term, and $\mathbf{A}$ is defined as follows

$$\mathbf{A} = J(\theta)^T \mathbf{W}_e J(\theta) + \mathbf{W}_n, \tag{3.31}$$

where $\mathbf{W}_n = \mathrm{diag}(\mathbf{w_n})(\mathbf{w_n} \in (\mathbb{R}^+)^n)$ is a diagonal damping matrix. The damping matrix guarantees that $\mathbf{A}$ is both non-singular and positive definite. As is evident, the performance of the method is highly dependent on the selection of $\mathbf{W}_n$.

The diagonal damping matrix w_n was first described as a constant by Wampler [76]. A little later, Chan and Lawrence [77] proposed a damped least-squares method where w_n directly depends on the error value E. Last but not least, Sugihara [63] proposed a robust method for obtaining the variable, as follows

$$\mathbf{W}_n = E\mathbf{1}_n + \mathrm{diag}(\tilde{\mathbf{w}}_n), \tag{3.32}$$

where the biasing value $\tilde{\mathbf{w}}_n$ ensures the computation converges sufficiently robustly, with $\tilde{\mathbf{w}}_n = 1.0 \times 10^{-4}$.

CHAPTER 4

Motion Planning and Control

Text ...

4.1 Joint Space and Operational Space

In the field of robotic manipulators, two configuration spaces are distinguished for defining the movement and control of robotic systems: joint space and operational space. These spaces represent different perspectives on the movement of robotic manipulators and provide distinct frameworks for analysis and control.

Joint space refers to the configuration space of a robot, where the state of the robot is defined by the positions, velocities, and accelerations of its individual joints. Each joint corresponds to a degree-of-freedom, and the combination of joint values determines the overall configuration of the robot. In joint space, the focus is on controlling and coordinating the movements of each joint independently. This space is often characterized by variables such as joint angles or joint displacements.

Operational space, also known as task space or Cartesian space, represents the space in which the end-effector of the robot operates. The position and orientation of the end-effector define the operational space, and the focus is on achieving specific tasks or objectives in this space. Operational space allows for a more intuitive and task-oriented approach to control, as it directly deals with the position and orientation of the end-effector in the environment.

CHAPTER 5

Versatile Intelligent Robotic Workstation in the Context of Industry 4.0

Text ...

CHAPTER 6

Conclusion

"If you want to improve something, you must first understand it. The combination of theoretical and practical knowledge is not an option, it is a must."

— Roman Parak

Text ...

Bibliography

[1] H. Kagermann, W. Wahlster, and J. Helbig, *Recommendations For Implementing The Strategic Initiative Industrie 4.0: Final Report of the Industrie 4.0 Working Group.* Ulrike Findeklee: Acatech – National Academy of Science and Engineering, 2013.

[2] H. Kagermann, W.-D. Lukas, and W. Wahlster, "Industrie 4.0: Mit dem internet der dinge auf dem weg zur 4. industriellen revolution," *VDI nachrichten*, vol. 13, 2011.

[3] A. Sharma and B. J. Singh, "Evolution of industrial revolutions: A review," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 11, pp. 66–73, 2020.

[4] P. N. Stearns, *The industrial revolution in world history.* Routledge, 2020.

[5] H. Mohajan, "The first industrial revolution: Creation of a new global human era," 2019.

[6] H. Mohajan, "The second industrial revolution has brought modern social and economic developments," 2019.

[7] J. Mokyr and R. H. Strotz, "The second industrial revolution, 1870-1914," *Storia dell'economia Mondiale*, vol. 21945, no. 1, 1998.

[8] K. Schwab, *The fourth industrial revolution.* Currency, 2017.

[9] M. Xu, J. M. David, S. H. Kim, *et al.*, "The fourth industrial revolution: Opportunities and challenges," *International journal of financial research*, vol. 9, no. 2, pp. 90–95, 2018.

[10] P. Thomas and D. Nicholas, "The fourth industrial revolution: Shaping new era," *Journal of International Affairs*, vol. 72, no. 1, pp. 17–22, 2018.

[11] J. Rymarczyk *et al.*, "Technologies, opportunities and challenges of the industrial revolution 4.0: theoretical considerations," *Entrepreneurial business and economics review*, vol. 8, no. 1, pp. 185–198, 2020.

[12] C. O. Klingenberg, M. A. V. Borges, and J. A. do Vale Antunes Jr, "Industry 4.0: What makes it a revolution? a historical framework to understand the phenomenon," *Technology in Society*, vol. 70, p. 102009, 2022.

[13] A. Rojko, "Industry 4.0 concept: Background and overview.," *International journal of interactive mobile technologies*, vol. 11, no. 5, 2017.

[14] S. I. Tay, T. Lee, N. Hamid, and A. N. A. Ahmad, "An overview of industry 4.0: Definition, components, and government initiatives," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 10, no. 14, pp. 1379–1387, 2018.

[15] M. Rüßmann, M. Lorenz, P. Gerbert, M. Waldner, J. Justus, P. Engel, and M. Harnisch, "Industry 4.0: The future of productivity and growth in manufacturing industries," *Boston consulting group*, vol. 9, no. 1, pp. 54–89, 2015.

[16] D. Palka and J. Ciukaj, "Prospects for development movement in the industry concept 4.0," *Multidisciplinary Aspects of Production Engineering*, vol. 2, pp. 315–326, 09 2019.

[17] Y. Uygun, "The fourth industrial revolution-industry 4.0," *Available at SSRN 3909340*, 2021.

[18] G. Erboz, "How to define industry 4.0: Main pillars of industry 4.0," *Managerial trends in the development of enterprises in globalization era*, vol. 761, pp. 761–767, 2017.

[19] F. Sherwani, M. M. Asad, and B. S. K. K. Ibrahim, "Collaborative robots and industrial revolution 4.0 (ir 4.0)," in *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*, pp. 1–5, IEEE, 2020.

[20] A. Oke and F. A. P. Fernandes, "Innovations in teaching and learning: Exploring the perceptions of the education sector on the 4th industrial revolution (4ir)," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 6, no. 2, p. 31, 2020.

[21] A. A. Shahroom and N. Hussin, "Industrial revolution 4.0 and education," *International Journal of Academic Research in Business and Social Sciences*, vol. 8, no. 9, pp. 314–319, 2018.

[22] M. Hermann, T. Pentek, and B. Otto, "Design principles for industrie 4.0 scenarios: A literature review," 01 2015.

[23] J. Ortiz, W. Marroquin, and L. Cifuentes, *Industry 4.0: Current Status and Future Trends.* 03 2020.

[24] R. Hall, S. Schumacher, and A. Bildstein, "Systematic analysis of industrie 4.0 design principles," *Procedia CIRP*, vol. 107, pp. 440–445, 2022. Leading manufacturing systems transformation – Proceedings of the 55th CIRP Conference on Manufacturing Systems 2022.

[25] S. Vaidya, P. Ambad, and S. Bhosle, "Industry 4.0 – a glimpse," *Procedia Manufacturing*, vol. 20, pp. 233–238, 2018. 2nd International Conference on Materials, Manufacturing and Design Engineering (iCMMD2017), 11-12 December 2017, MIT Aurangabad, Maharashtra, INDIA.

[26] P. Brooks, "Ethernet/ip-industrial protocol," in *ETFA 2001. 8th International Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No. 01TH8597)*, vol. 2, pp. 505–514, IEEE, 2001.

[27] J. Feld, "Profinet-scalable factory communication for all applications," in *IEEE International Workshop on Factory Communication Systems, 2004. Proceedings.*, pp. 33–38, IEEE, 2004.

[28] A. L. Dias, G. S. Sestito, A. C. Turcato, and D. Brandão, "Panorama, challenges and opportunities in profinet protocol research," in *2018 13th IEEE International Conference on Industry Applications (INDUSCON)*, pp. 186–193, IEEE, 2018.

[29] W. Mahnke, S.-H. Leitner, and M. Damm, *OPC unified architecture*. Springer Science & Business Media, 2009.

[30] OPC Foundation, "Opc unified architecture." `www.opcfoundation.org/`. Online. Accessed on 7 October 2023.

[31] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital twin in manufacturing: A categorical literature review and classification," *Ifac-PapersOnline*, vol. 51, no. 11, pp. 1016–1022, 2018.

[32] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: Enabling technologies, challenges and open research," *IEEE access*, vol. 8, pp. 108952–108971, 2020.

[33] A. Parrott and L. Warshaw, "Industry 4.0 and the digital twin," *Deloitte Insights*, 2017.

[34] F. Pires, A. Cachada, J. Barbosa, A. P. Moreira, and P. Leitão, "Digital twin in industry 4.0: Technologies, applications and challenges," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, vol. 1, pp. 721–726, IEEE, 2019.

[35] T. R. Wanasinghe, L. Wroblewski, B. K. Petersen, R. G. Gosine, L. A. James, O. De Silva, G. K. Mann, and P. J. Warrian, "Digital twin for the oil and gas industry: Overview, research trends, opportunities, and challenges," *IEEE access*, vol. 8, pp. 104175–104197, 2020.

[36] C. Weidemann, N. Mandischer, F. Kerkom, B. Corves, M. Hüsing, T. Kraus, and C. Garus, "Literature review on recent trends and perspectives of collaborative robotics in work 4.0," *Robotics*, vol. 12, p. 84, 06 2023.

[37] K. C. Haug, T. Kretschmer, and T. Strobel, "Cloud adaptiveness within industry sectors–measurement and observations," *Telecommunications policy*, vol. 40, no. 4, pp. 291–306, 2016.

[38] K. Witkowski, "Internet of things, big data, industry 4.0–innovative solutions in logistics and supply chains management," *Procedia engineering*, vol. 182, pp. 763–769, 2017.

[39] P. K. R. Maddikunta, Q.-V. Pham, B. Prabadevi, N. Deepa, K. Dev, T. R. Gadekallu, R. Ruby, and M. Liyanage, "Industry 5.0: A survey on enabling technologies and potential applications," *Journal of Industrial Information Integration*, vol. 26, p. 100257, 2022.

[40] S. Huang, B. Wang, X. Li, P. Zheng, D. Mourtzis, and L. Wang, "Industry 5.0 and society 5.0—comparison, complementation and co-evolution," *Journal of manufacturing systems*, vol. 64, pp. 424–428, 2022.

[41] A. Akundi, D. Euresti, S. Luna, W. Ankobiah, A. Lopes, and I. Edinbarough, "State of industry 5.0—analysis and identification of current research trends," *Applied System Innovation*, vol. 5, no. 1, p. 27, 2022.

[42] K. A. Demir, G. Döven, and B. Sezen, "Industry 5.0 and human-robot co-working," *Procedia computer science*, vol. 158, pp. 688–695, 2019.

[43] S. Nahavandi, "Industry 5.0—a human-centric solution," *Sustainability*, vol. 11, no. 16, p. 4371, 2019.

[44] Z. Lv, "Digital twins in industry 5.0," *Research*, vol. 6, p. 0071, 2023.

[45] B. Siciliano and O. Khatib, *ed. Springer Handbook of Robotics.* Berlin, Heidelberg: Springer-Verlag, 2016.

[46] K. M. Lynch and F. C. Park, *Modern robotics.* Cambridge University Press, 2017.

[47] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control.* Springer Publishing Company, Incorporated, 1st ed., 2008.

[48] S. LaValle, "Planning algorithms," *Cambridge University Press google schola*, vol. 2, pp. 3671–3678, 2006.

[49] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in Python*, vol. 146. Springer Nature, 2023.

[50] J. Diebel *et al.*, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.

[51] P. R. Evans, "Rotations and rotation matrices," *Acta Crystallographica Section D: Biological Crystallography*, vol. 57, no. 10, pp. 1355–1359, 2001.

[52] A. Lerios, "Rotations and quaternions," *Standford University,[online],[accessed Oct. 23, 2012], available from World Wide Web:< http://citeseerx. ist. psu. edu/viewdoc/summary*, 1995.

[53] S. W. Shepperd, "Quaternion from rotation matrix," *Journal of guidance and control*, vol. 1, no. 3, pp. 223–224, 1978.

[54] A. Renfrew, "Introduction to robotics: Mechanics and control," *International Journal of Electrical Engineering & Education*, vol. 41, no. 4, p. 388, 2004.

[55] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control.* John Wiley & Sons, 2020.

[56] S. Starke, N. Hendrich, and J. Zhang, "Memetic evolution for generic full-body inverse kinematics in robotics and animation," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 406–420, 2018.

[57] A. Peidró, Ó. Reinoso, A. Gil, J. M. Marín, and L. Payá, "An improved monte carlo method based on gaussian growth to calculate the workspace of robots," *Engineering Applications of Artificial Intelligence*, vol. 64, pp. 197–207, 2017.

[58] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," 1955.

[59] W. Khalil and E. Dombre, *Modeling identification and control of robots.* CRC Press, 2002.

[60] E. Dombre and W. Khalil, *Robot manipulators: modeling, performance analysis and control.* John Wiley & Sons, 2013.

[61] S. Chiaverini, B. Siciliano, and O. Egeland, "Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator," *IEEE Transactions on control systems technology*, vol. 2, no. 2, pp. 123–134, 1994.

[62] S. R. Buss, "Introduction to inverse kinematics with jacobian transpose," *Pseudoinverse and Damped Least Squares methods*, p. 19, 2004.

[63] T. Sugihara, "Solvability-unconcerned inverse kinematics by the levenberg–marquardt method," *IEEE transactions on robotics*, vol. 27, no. 5, pp. 984–991, 2011.

[64] W. Suleiman, F. Kanehiro, and E. Yoshida, "Infeasibility-free inverse kinematics method," in *2015 IEEE/SICE International Symposium on System Integration (SII)*, pp. 307–312, IEEE, 2015.

[65] S. KuCuk and Z. Bingul, "The inverse kinematics solutions of industrial robot manipulators," in *Proceedings of the IEEE International Conference on Mechatronics, 2004. ICM'04.*, pp. 274–279, IEEE, 2004.

[66] R. Konietschke and G. Hirzinger, “Inverse kinematics with closed form solutions for highly redundant robotic systems,” in *2009 IEEE International Conference on Robotics and Automation*, pp. 2945–2950, IEEE, 2009.

[67] P. Donelan, *Kinematic singularities of robot manipulators.* INTECH Open Access Publisher, 2010.

[68] B. Siciliano, “Kinematic control of redundant robot manipulators: A tutorial,” *Journal of intelligent and robotic systems*, vol. 3, pp. 201–212, 1990.

[69] M. T. Mason, *Mechanics of robotic manipulation.* MIT press, 2001.

[70] A. Balestrino, G. De Maria, and L. Sciavicco, “Robust control of robotic manipulators,” *IFAC Proceedings Volumes*, vol. 17, no. 2, pp. 2435–2440, 1984.

[71] W. A. Wolovich and H. Elliott, “A computational technique for inverse kinematics,” in *The 23rd IEEE Conference on Decision and Control*, pp. 1359–1363, IEEE, 1984.

[72] S. R. Buss, “Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods,” *IEEE Journal of Robotics and Automation*, vol. 17, no. 1-19, p. 16, 2004.

[73] M. Meredith and S. Maddock, “Real-time inverse kinematics: The return of the jacobian,” 2004.

[74] J. E. Dennis Jr and R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations.* SIAM, 1996.

[75] K. Anderson and J. Angeles, “Kinematic inversion of robotic manipulators in the presence of redundancies,” *The International Journal of Robotics Research*, vol. 8, no. 6, pp. 80–97, 1989.

[76] C. W. Wampler, “Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 1, pp. 93–101, 1986.

[77] S. K. Chan and P. D. Lawrence, “General inverse kinematics with the error damped pseudoinverse,” in *Proceedings. 1988 IEEE international conference on robotics and automation*, pp. 834–839, IEEE, 1988.

Appendix A: Activities Related to Doctoral Studies

Text ...

Appendix B: List of Scientific Publications

Text ...

Appendix C: Source Codes

"Active participation within the open-source community, not only as a user but also as a contributor, is essential to ensuring continued growth."

— Roman Parak

Text ...