Robotics in Mechatronics Engineering

Md. Tarek Ur Rahman Erin, Ahmed Farhan, Protik Barua, Naheen Ibn Akbar, Rezwan Us Saleheen, and Md Sadatuzzaman Saagoto

Abstract This chapter, "Robotics in Mechatronics Engineering," provides an overview of the area of robotics, including its essential components, the course of history, and integration with mechatronics. This chapter investigates robotic kinematics and dynamics, including topics like location, velocity, motion control, and forward and inverse kinematics. It categorizes robots based on their applications, focusing on industrial, service, and collaborative robots, each with distinct characteristics and use cases. Furthermore, the chapter stresses the importance of robot vision and perception by discussing how robots utilize sensors and cameras to see their surroundings, recognize objects, and apply spatial intelligence, all of which contribute to their autonomy. This comprehensive outline establishes a solid basis for robotics within mechatronics and its numerous implications for various sectors.

Keywords Robots · Robotics · Mechatronics · Sensors · Automation · Controller

1 Introduction

The advancement of contemporary technology has gained a different dimension with robotics and mechatronics engineering in recent years. Mechanical, electrical, and computer engineering are combined in a way that works well to provide the theoretical foundation for robotics and mechatronics engineering. This area of engineering is quite young. The word "mechatronics" emphasizes how important it is to combine mechanical and electrical systems into a particular device. Mechatronics is the field in which engineering science, applied science, and engineering principles come together to facilitate the design, fabrication, and use of devices. As was previously said, robotic technologies may be thought of as a subset of mechatronics. The

Md. T. U. R. Erin
Department of Mechanical Engineering, Textile Engineering College, Chattogram, Bangladesh

A. Farhan · P. Barua (✉) · N. I. Akbar · R. U. Saleheen · M. S. Saagoto
Department of Mechatronics Engineering, World University of Bangladesh, Dhaka, Bangladesh
e-mail: protik.barua@mte.wub.edu.bd

M. M. Rahman et al. (eds.), *Mechatronics*, Emerging Trends in Mechatronics,
https://doi.org/10.1007/978-981-97-7117-2_4

procedure begins with the device's design and ends with the user-built equipment operating without a hitch [1].

1.1 Overview of Robotics

Engineering, science, and technology come together in robotics to create the machine known as a robot. The field of robotics involves the development and use of robotic systems. Robots are mostly used to replace people in a variety of activities. Robot design, building, and operation are the main topics of robotics, together with computer systems for control, feedback, and information processing. It is a machine that has been programmed to carry out the assigned duties and collect data from its environment. It has sensors to perceive its environment and is driven by a central CPU to regulate its motions [2].

1.2 Mechatronics: Integration of Mechanical, Electrical, and Computer Engineering

Mechatronics was first used by the Japanese scientists in the late 1970s to refer to the approach they used while designing the electromechanical systems' subsystems. Mechatronics is the design of goods and manufacturing processes that combine the synergistic elements of electrical, mechanical, system, and control engineering. The integrated and multidisciplinary approach to engineering design is becoming more and more common to design automobiles, machine tools, robotics, cameras, and other devices. We need to incorporate mechanical, electrical, electronic, and control engineering from the outset of the design process in order to create systems that are more affordable, dependable, and adaptable. Systems for mechatronics are developing with an emphasis on efficiency and automation. These systems start to behave more and more like robots as they become more independent. Common illustrations of mechatronics projects include robot controllers, biped walking robots, robotic hands, animatronic hands, and manufacturing plant automation.

Mechatronics brings the areas of technology together involving sensors, actuators and drives, measurement systems, and microprocessor systems, as well as the control system behavior and analysis of system [3]. Figure 1 illustrates these basic elements of a mechatronics system.

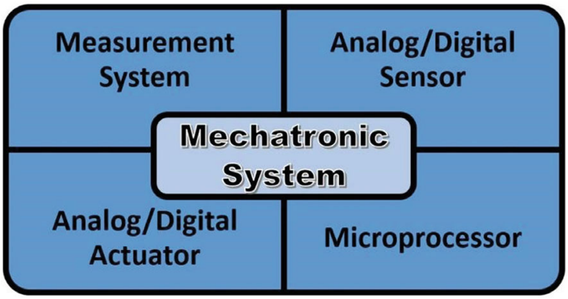


Fig. 1 The major elements of a mechatronic system

1.3 Significance of Robotics in Mechatronics

Robotics and mechatronics are two fields that have a lot in common. Both involve the design and construction of machines that interact with their environment in some way. Both necessitate a profound comprehension of the functioning of these machines, enabling efficient and effective design. Robotics and mechatronics are the points at which mechanical and electrical engineering come together, using computer-controlled systems to make devices smarter and more efficient. Robotic systems assemble mechanical, electrical, and software components to accomplish the desired function.

Robotics has led to the creation of a new field of study called mechatronics, which focuses on the interaction between robots and humans. Today, the manufacturing industry most commonly uses mechatronics, employing robots to perform tasks that would otherwise require significant human time and effort [4].

2 Fundamentals of Robotics

The need for automation with industrial robots is still growing in spite of a lack of skilled labor and the shifting of manufacturing to high-wage industrial nations. For the correct design of robot systems, it is essential to understand the many types of kinematic robots, their benefits and drawbacks, the unique motion behavior of industrial robots and how to mathematically describe them, the parts of automation systems, and, of course, safety considerations.

Understanding robotics begins with an understanding of the many types of kinematic robots, their characteristics, and their uses. It focuses on the scientific description of robotic motions as utilized for robot simulation and control, the calculation of rotations, and the application of Denavit-Hartenberg rules to define kinematic chains. It also addresses the fundamentals of controlling motion and path planning, robot programming, and the systematic design of general handling systems which includes interactive, teach-in, and automatic offline programming. It also includes safety equipment, characteristics of safe robot cell design, robot hardware, and the precision of robot-based actions [5].

2.1 Historical Development

Modern culture is driven by technological breakthroughs in fields such as robots, electronics, and software. Every day, technology advances and transforms the way we work and do business. Autonomous robotics and artificial intelligence (AI) technologies are arguably at the forefront of contemporary technological innovation. Undoubtedly, mobile, self-governing robots will be important in the future.

The business of commercial robot has progressed significantly in only fifty years. "Cobots" or collaborative robots can operate alongside human beings. These robots are coined as a result of the progression of robotics in the commercial field. The question of whether autonomous robots will eventually replace humans in jobs or work alongside them is still up for debate [6].

We must look back to the 1940s and 1950s, when robots first appeared, in order to comprehend the effects, they have on various businesses. British neurophysiologist W. Grey Walter is the pioneer of this industrial robot who is frequently credited with creating the two robots, Elmer and Elsie, in the late 1940s. The Grey Walter's Tortoises demonstrated basic autonomous behaviors that were modeled after biological systems. The robots could traverse their environment and even react to stimuli using simple sensors, mimicking human behavior. A brief historical advancement of robot is illustrated in Table 1.

Table 1 Evolution of robots and robotics

Time	Significance	Name of the robot	Inventor
First century A.D. and earlier	Pneumatica and Automata by Heron of Alexandria contain descriptions of over a hundred machineries and automata, such as a steam-powered engine, a coin-operated machine, a wind organ, and a fire engine	None	Heron of Alexandria, Ctesibius, Philo of Byzantium, and others
1206	The first robots with humanoid programming	Four robotic musicians on a boat	Al-Jazari
1495	Humanoid robotic designs	Mechanical knight	Leonardo da Vinci
1738	A mechanical duck with the ability to fly, eat, and urinate	Digesting Duck	Jacques de Vaucanson

(continued)

Table 1 (continued)

Time	Significance	Name of the robot	Inventor
1800s	Japanese mechanical toys with painted surfaces, arrow shooters, and tea services	Karakuri toys	Tanaka Hisashige
1921	The first imagined automata referred to as "robots" first appear in the drama R. U. R	Rossum's Universal Robots	Czech writer Karel Čapek
1930s	Robotic humanoid displayed at the 1939 and 1940	Elektro	Westinghouse Electric Corporation
1949	Basic robots with self-navigation behaviors	Elsie and Elmer	William Grey Walter
1956	Based on George Devol's inventions, the Unimation firm, created by Joseph Engelberger and George Devol, produced the first commercial robot	Unimate	George Devol
1961	Commercial robot installation	Unimate	George Devol
1963	Robot for palletizing	Palletizer	Fuji Yusoki Kogyo
1973	The first six electro-mechanically powered axes of an industrial robot	Famulus	KUKA Robot Group
1975	Universal manipulation arm	PUMA	Victor Scheinman

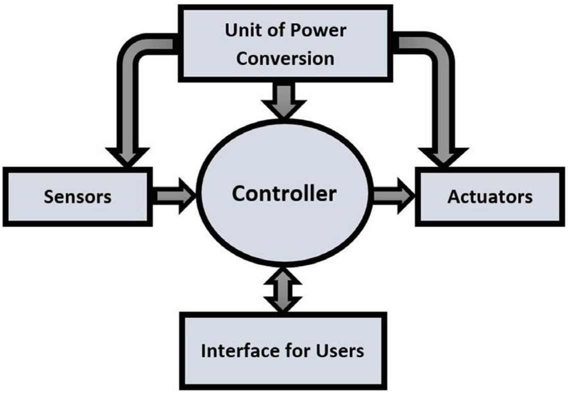


Fig. 2 Key components of robotics

2.2 *Key Components of Robotic Systems*

A robot is usually an electromechanical system consisting of some components like power supplies, sensors, actuators, controllers, and some other components [7]. Figure 2 illustrates the basic components of a robotic system.

The major components are identified by their respective names.

- **Power supplies**: The primary function of a power supply unit is to provide power to an electrical load and to transform current of a source into the proper frequency and voltage. As a result, they are occasionally denoted as electric power converters.
- **Feedback devices/Sensors**: A sensing unit is an apparatus that offers data on the external environment. These gadgets belong to a class that is used to watch over a process or operation and then confirm that it actually took place. A sensor is an apparatus that measures and converts a physical quantity into a signal so that an instrument or observer can understand. Returning to the analogy of a human body, sensors are the organs that provide information to the brain (controller). To detect the locations of different joints and linkages, a robot might attach a feedback device and provide data to the controller. The robot's arm may operate basic limit switches or position-measuring tools like encoders, resolvers, tachometers, and/or potentiometers. The feedback data is either digital or analog based on the devices.
- **Controllers**: The robot controller is a software and hardware-based device that is connected to the robot and serves as its brain. The robot part that manages the movements of the entire mechanical system is called the controller. It also uses a range of sensors to gather information from its immediate surroundings. A microprocessor at the heart of the robot's controller connects to the input/output and monitoring devices. The controller issues a command that activates the motion control mechanism, comprising various controllers, actuators, and amplifiers. The controller unit receives the signal from the sensor and drives the actuators. Consider the Arduino Uno, a microcontroller board that relies on the ATmega328.
- **Actuator**: An actuator plays a crucial role in powering a robot's mechanical unit. The energy-conversion components found within robots are called actuators. Actuators' primary job is to transform energy into motion. Sensors are the eyes and ears of a robot, whereas actuators work like muscles. Actuators are tiny motors which are directly attached to the structure of the machine that facilitate movement. Actuators commonly use DC-geared motors. Servo and stepper motors can also be used as actuators.
- **Manipulator**: Figure 3 shows the manipulator of the parts in the robots without direct operator contact using a sequence of link and joint connections. The arm and the robot body make up the two components of the robot manipulator [8]. A machine or robotic mechanism is usually composed of many segments that are either joint or slide in relation to each other. These segments are used for gripping and/or moving things, such as tools or parts, usually in several degrees of freedom. These mechanical connections are driven by actuators.
- **End-Effectors**: An end-effector attaches to the end of a robotic arm to operate. It moves or orients the process or product. Specialized tasks like welding, measuring, marking, drilling, cutting, painting, and cleaning utilize it [9].

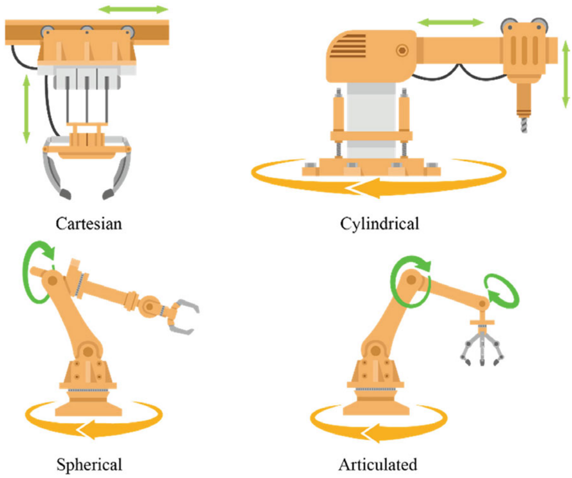


Fig. 3 Different types of workspaces

2.3 Robot Joints

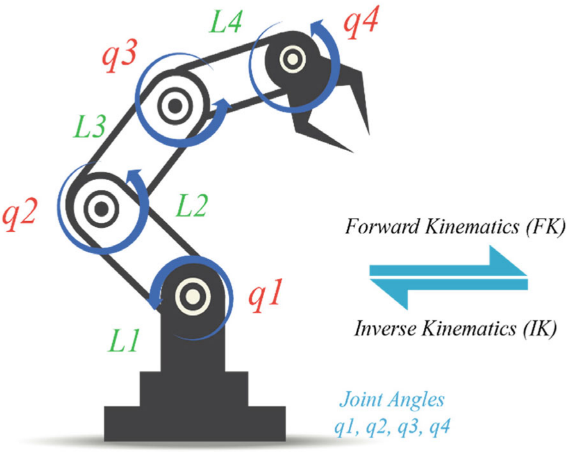
Because they enable the connections to move in a variety of ways, the robot joints are the most crucial component. Joints come in five main categories.

- **Linear Joint:** A linear joint is represented by the symbol L. This kind of joint has the ability to move both slidingly and translationally.
- **Orthogonal Joint:** Symbol "O" is used to represent the orthogonal joint. It has certain similarities with the linear joint. The input and output connections will move at right angles, which is the sole change.
- **Rotational Joint:** Symbol R can also be used to symbolize a rotating joint that can rotate along the axis that runs parallel to the axes of the arms.
- **Twisting Joint:** It is known as a twisting joint, joint T. The movement of this joint between the input and output links is twisting. The rotation axis and the output link's axis are vertically aligned throughout this operation. The output link rotates with the relation to the input link.
- **Revolving Joint:** The V joint is the popular name for the revolving joint. The input link is parallel to the rotating axes and the output link axis is perpendicular to the rotational axis. The output link rotates around the input link like a twisting joint [10].

2.4 Classification of Robots in Mechatronics Context

There are many different sizes and forms of robots as shown in Fig. 4. They can perform a variety of arm manipulations and motion systems. Most commercially

Fig. 4 Kinematics of robot



available industrial robots have five basic configurations based on their physical characteristics. There are, nonetheless, a number of other kinds of robot setups.

There are various types of industrial robots, as outlined below.

2.4.1 Cartesian/Rectangular/Gantry Configuration (PPP)

This configuration uses three perpendicular slides for the construction of the x, y, and z axis. The robot can operate within a rectangular work envelope by moving three slides relative to one another. They are also known as Rectangular or Cartesian configuration robots.

2.4.2 Cylindrical Configuration (RPP)

These robots make use of a slide-moving vertical column. This robot may travel radially with respect to the column by fastening an arm to the slide. The robot may retrieve a cylindrical work envelope by rotting the column.

2.4.3 Spherical/ Polar Configuration (RRP)

The robot utilizes a telescopic arm with the ability to raise or lower itself around a horizontal pivot point. A rotating base mounts the pivot point, enabling the robot to move vertically. The robot can move its arm within a spherical enclosure.

2.4.4 Articulated/ Revolute/ Jointed Arm/ Anthropomorphic Configuration (RRR)

It is made up of two straight halves, the elbow and shoulder joints which revolve along horizontal axes that represent the human upper and forearm. The shape of the work envelope is not uniform.

2.4.5 SCARA Configuration (RRP)

This particular kind of the jointed arm robot rotates the elbow and shoulder joints along vertical axes rather than horizontal ones. Its cylindrical work envelope, which is far greater than that of any other arrangement, offers significant vertical stiffness for a number of crucial jobs [11]. Robots have several types, like Cartesian, spherical, articulated, and cylindrical [12].

3 Work Envelop/Work Space

The region encompassing all points in the surrounding space that the robot arm can reach is referred to as the work envelope (also called the robot reach or work volume). It is one of the most crucial factors to take into account when choosing a good robot since the application space must remain within the chosen robot's reach. The collection of places that a robot manipulator's end-effector can reach is known as its workspace. Stated differently, a robot's workspace is the area where its mechanism operates. Workspaces have different categories, for example, cylindrical, Cartesian, articulated, and spherical [13]. These categories of workspaces are shown in Fig. 3.

- **Cartesian Work Envelope**: This is a rectangular prism that represents a Cartesian configuration envelope. Its work envelope is devoid of dead zones, allowing the robot utilizing it to freely operate the highest payload it is capable of handling without experiencing any strain. This kind of work envelope is present in gantry or Cartesian robots.
- **Cylindrical Work Envelope**: This arrangement, as its name implies, is cylindrical in shape and has a hollow center, which significantly restricts the robotic arm's range of motion. This arrangement leaves a sizable dead zone both inside and outside the robot frame. This kind of envelope is most appropriate for short, simple operations requiring little movement performed by robots.
- **Spherical/ Polar Work Envelope**: The volume situated between two half spheres is swept up by this arrangement. The robotic arm's maximum angular rotations in both the vertical and horizontal axes are restricted by this design. Unfortunately, these physical constraints result in large, noticeable dead zones above and below the concerned robot.

- **Articulated/ Revolute Work Envelope**: Usually, this unusual arrangement has an envelope larger than the available floor area. In essence, the robotic arm is supported by a large, open work envelope that may be either completely closed or partially opened. In terms of work envelope arrangements, it is among the most intricate.
- **SCARA Work Envelope**: This arrangement is unique to SCARA robots and usually resembles a kidney or heart. Its distinctive hollow circle, which runs directly through the center, facilitates simpler lateral motions and offers a very wide horizontal plane of operation. Vertical motions are, however, limited since there is very little room remaining for upward motion [14].

Dead Zone: Often, the robot arm cannot reach certain areas within the working envelope. These areas are known as dead zones.

4 Kinematics and Dynamics

The geometry and algebra of robot motion are studied in robotic kinematics. It explains the relationship between a robot's joints' position, orientation, velocity, and acceleration, as well as their links to each other and the environment. Typically, the robot kinematics have two types: forward and inverse. Given the joint angles, forward kinematics calculates the end-effector pose (the position and orientation of the robot's tool or sensor). Inverse kinematics computes the joint angles given the desired end-effector pose. Robot kinematics is essential for planning and executing robot motions, such as reaching, grasping, or manipulating objects.

Robot dynamics is the study of the forces and torques that affect robot motion. It describes how mass, inertia, friction, gravity, and external forces influence the robot's acceleration and energy. Robot dynamics can be classified into two categories: direct and inverse. Direct dynamics computes the joint torques given the joint angles and accelerations. Given the joint angles and torques, inverse dynamics computes the joint accelerations. Robot dynamics is important for designing and controlling robot systems, such as balancing, stabilizing, or optimizing performance [15].

4.1 Basics of Robot Movement Control/Trajectory of a Robot

Trajectory planning is critical for robotic applications and automation in general. The capacity to produce trajectories with provided characteristics is a critical element to ensuring substantial outcomes in terms of quality and convenience of accomplishing the required motion, especially at the high operating speeds needed in many applications. [16]. A trajectory is a robot's configuration as a function of time. There are four types of robot trajectory and motion control. They are mentioned below.

4.1.1 Point-To-Point Control Robot (PTP)

A time-optimal movement in three dimensions between two specified points is called a point-to-point movement (PTP movement). Robot axes are traveling from the present location (M1) to the target point (M2) in synchrony, which causes the end-effector's trajectory to curve. PTP motions are utilized for fast positioning, and then a targeted operation or controlled path motion starting at the target location is applied (M2). The PTP robot has the ability to move between locations. The locations are stored in the control memory [17].

4.1.2 Continuous-Path Control Robot (CP)

It is possible for the CP robot to go along the predetermined course. The robot may halt at any point in the controlled path with CP from a single control. Every point along the path needs to be explicitly stored in the robot's control memory. The most basic example of this kind of robot is one that moves in a straight line [18]. Several robots that use continuous pathways can also be trained to follow a smooth curve. In these cases, the controller unit stores a large number of discrete point locations along the path in memory (teach-in), and the programmer manually leads the robot arm along the desired path.

4.1.3 Controlled-Path Robot

Controlled-path robots are capable of producing very accurate pathways with varying geometry, including circles, straight lines, and interpolated curves. Almost every site on the given path may get high accuracy. Just the start and finish locations and the path definition function need to be kept in the robot's control memory [19].

4.1.4 Stop-To-Stop

The system is open-loop. The controller does not know the position or velocity. The on/off commands are stored by the controller as valve states. The final trip is also adjusted via a mechanical device.

4.2 Kinematic Analysis

The link between a robot's endpoint motion and joint motion is known as kinematics. It is a basic and traditional topic in robotic system that analyzes the link between the joint coordinates of a robot and its spatial arrangement. Figure 4 illustrates the kinematics of a robot. When calculating the location of a gripper in space, creating a mechanism to move a tool from one position to another position, or determining if a robot's motion will collide with an obstruction, kinematics can provide extremely precise results [20].

Characteristics including singularity manifolds, joint restriction, collision avoidance, and redundancy are all covered by robotic kinematics. For this reason, kinematic analysis is essential to a robot manipulator's positioning duty. The kinematics of the robot arms often divide into forward and inverse solutions.

4.2.1 Forward Kinematics

The end-effector placement of a robot is determined by computational processes. The processes employ combined sensors and mathematical algorithms to pinpoint its position. A manipulator is made up of serial links that are connected to one another from the base frame through the end-effector by prismatic or revolute joints. The location and orientation of the end-effector are determined by forward kinematics using the joint variables.

4.2.2 Inverse Kinematics

In inverse kinematics, the robot's motion to a target point is calculated using kinematic equations. For example, automated bin picking on a production line involves a robotic arm that moves precisely between bins and manufacturing units from one location to another. The grabbing end of the robot arm is called the end-effector. The robot configuration is a list of joint positions that do not deviate from any limitations the robot has and fall inside the position limits of the robot model.

4.3 Dynamic Analysis and Control

The link between the accelerations that a robot mechanism produces and the forces exerted on it is the focus of robot dynamics. Robot dynamics models the robot mechanism as a rigid-body system and applies rigid-body dynamics to robotics. Robot dynamics has two main problems, forward dynamics and inverse dynamics. Figure 5 represents these two problems of robot dynamics.

Forward dynamics is also referred to as "direct dynamics," or just "dynamics." It is mostly employed in simulation. Numerous applications of inverse dynamics exist, such as the online control of robot forces and movements, robot mechanism design, trajectory design, and optimization, and as a building block for some forward-dynamics methods [21].

Another type of dynamics is hybrid dynamics, which determines the unknown forces and accelerations given the forces at certain joints and the accelerations at others. This system can often flow and jump at the same time.

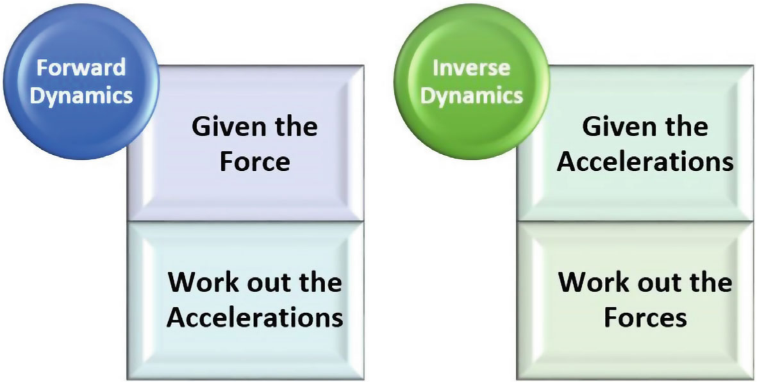


Fig. 5 Problems of robot dynamics

5 Programming and Control

Robot control is an essential technology that enables a robot to move precisely and adaptively. In order to ensure that the robot can do its mission and achieve the required transient and steady state, the formulation of the robot control issue entails defining the joint torques that the joint actuators will create. Robot controls are divided into two types based on the restrictions of the environment: motion control in unconstrained space and interaction control in restricted space.

5.1 Programming Paradigms for Robotics

Programming is a key aspect of any robotic system. It transmits vital information to an industrial robot, allowing it to function and perform specific tasks. Programming robots to perform their tasks can take various forms.

5.1.1 Online-Programming Method

- **Teach Method**:

With this technique, the robot is guided through a sequence of locations and their memory is retained by the use of a teaching necklace. Teaching in robotics refers to the act of directing and logging these locations, or coordinates, in space. Most contemporary industrial robots available today have a teach pendant that enables anybody, not only engineers, to operate the robot and train it to carry out certain tasks. Custom apps installed into a specialized tablet or touchscreen device constitute modern teaching pendants. Currently, it is the most user-friendly and favored method for reprogramming and programming industrial robots.

The following are some of the reasons why this strategy is preferred above others:

Precision: It is simpler to make sure that the robot arm performs as intended when the operator can enter extremely accurate points or coordinates into the teaching pendant.

Safety: It is simpler to guarantee that the robot arm and other moving elements remain inside a designated area for safety reasons because of the accuracy. For robots equipped with safety stops or collision detection systems, this is especially crucial.

Sense of intuition: Contemporary tablet-based instructional systems are so natural that industrial robots may not even need to be reprogrammed by robotics professionals. A teaching pendant with controls to drive the robot in many coordinate systems is used to manually drive the robot to the required places.

The only drawback of the teaching method is its limited ability to issue multiple instructions at once. The technique does not allow for the insertion of new commands to increase efficiency or adapt to changing circumstances; it simply trains the robot a predetermined set of motions.

- **Play-Back/Hand-Guiding/Lead-Through Programming**:

Lead-through programming, or hand-guiding, involves physically directing the robot arm over a sequence of points and axes to "teach" it how to accomplish a specific task. For example, a robot can be instructed by an operator on how to reach for and utilize a tool on a workstation. The robot will continue the same route on its own after it has stored the instructions in its memory.

Smaller modern robots, specifically designed to work alongside human operators, prefer hand-guiding. Indeed, the robot can train and perform intricate tasks like painting, sorting items, precision welding, and engraving, among others.

Lead-through or hand-guiding offers several advantages, which are as follows:

- Intuitiveness.
- Safety.
- Collaboration.
- Great for complex tasks or movements.

However, as a programming tool, hand guidance falls short when accuracy is needed. As such, this approach is not designed for human cooperation and is not feasible for big industrial robots. For contemporary collaborative robots utilized in smaller industrial settings, this approach is the most popular.

5.1.2 Offline-Programming Method

Using offline programming, robots or programmers may generate programs and route data straight from CAD models of the items they are handling. Writing instructions on a different system and testing them on virtual versions of industrial robots is the process of robotic programming. The written and verified instructions are uploaded into the robot's memory. Before implementing the software in real life, the manufacturer needs to create it and test a lot of times. Offline programming techniques usually work best for complicated applications where human programming would take a long time. The following are a few benefits of offline programming:

- More complexity is possible.
- More effective robot operation.
- Supports precision.
- Safety is ensured.

Due of its technical nature, offline programming is not recommended for non-engineers who are not familiar with low-level robotics programming or coding. All programming techniques are still widely used today in the majority of businesses. Depending on the type of robot and its programming, the industry can select a technique [22].

5.2 Control Techniques in Robotics

5.2.1 Force Control in Robotics

The capacity to manage the physical touch between a robot and its surroundings is a basic need for a manipulation job to be successful. Sadly, basic motion control is insufficient since modeling errors and unknowns can raise the contact force and make the interaction unstable, particularly in stiff surroundings. A robotic system needs force feedback and control in order to function reliably and safely in human presence, as well as to exhibit robust and varied behavior in unstructured situations. Maintaining a desired force or limiting the contact forces might be the control aim along the directions where interaction occurs while ensuring a desired motion in the other directions. The ability to regulate the force that a machine or a robot manipulator applies to an item or its surroundings is known as force control. During handling, regulating the contact force helps to avoid harming people, the machine, and the processed products. By preserving a constant contact force, it helps minimize wear and correct faults in manufacturing activities [23].

5.2.2 Cartesian Control

A Cartesian control robot has the ability to move a robotic manipulator arm along linear Cartesian axes. To put it simply, the robot arm can move in a straight line in the following directions: forward, back, left, right, up, and down. Certain robotic systems, referred/known as "Cartesian Coordinate Robots," are limited to moving linearly along predetermined axes. An excellent illustration of this is a CNC machine, or the claw game at your neighborhood arcade [24].

6 Software Platforms for Robot Control

Industrial robot programming and control require a symbiotic relationship between software intelligence and precise hardware. A robotics software platform is a software bundle that provides the following characteristics to simplify the programming of different robotic devices:

- an integrated programming and service execution environment;
- a collection of reusable parts;
- an environment for debugging and simulation;
- a package containing "drivers" for the majority of robotics hardware;
- a package containing common features like computer vision, navigation, or robotic arm control.

Below are the software platforms that stand out in the industry for their robustness, flexibility, and user-friendly interfaces, making them perfect for your robotic programming and control needs:

- **Robotics IDEs**: Integrated Development Environments (IDEs) are essential for writing and testing code. A good IDE for industrial robotics offers a comprehensive suite of tools tailored to the specific requirements of robot programming. It should include features like syntax highlighting, code completion, and debugging tools. Additionally, it should support the programming languages commonly used in robotics, such as Python, $C++$, or proprietary scripting languages. The user interface should be intuitive, allowing users to visualize the robot's movements and operations using the provided code.
- **Simulation Software**: Before deploying code to a physical robot, simulation software can save time and reduce risk by allowing you to visualize and test the programs in a virtual environment. This type of software should accurately replicate the physics and constraints of a real-world robot. It often includes features like collision detection and real-time feedback, enabling you to refine the code and ensure it operates as intended before running it on actual hardware.
- **Motion Planning**: Motion planning software is pivotal for defining the trajectories and actions of industrial robots. The software must calculate optimal paths while avoiding obstacles and ensuring the safety of human workers. Look for platforms that allow for easy parameter adjustment and can handle complex tasks like picking, placing, and assembling with precision. The best motion planning tools offer a balance between simplicity for quick tasks and the depth needed for more intricate operations.
- **Control Frameworks**: Motion planning software is pivotal for defining the trajectories and actions of industrial robots. The software must calculate optimal paths while avoiding obstacles and ensuring the safety of human workers. Look for platforms that allow for easy parameter adjustment and can handle complex tasks like picking, placing, and assembling with precision. The best motion planning tools offer a balance between simplicity for quick tasks and the depth needed for more intricate operations.

- **Collaborative Robotics**: With the rise of collaborative robots (cobots) that work alongside humans, software platforms need to ensure safety and ease of interaction. These platforms should support features like teach pendants, which allow operators to manually guide a robot through movements and then replicate those movements automatically. The software should also include safety protocols and be compliant with industry standards to prevent accidents.
- **Customization Tools**: Lastly, customization tools are vital for tailoring the software platform as needed. Whether it's integrating with existing systems or developing custom algorithms for unique tasks, the software should offer flexibility. It should provide APIs (application programming interfaces) or SDKs (software development kits) that allow for the extension of functionality and seamless integration with other tools and systems [25].

7 Perception, Localization, and Mapping

A variety of fundamental technological components are needed by autonomous and assistive systems, such as cars, trucks, and airplanes, in order for them to operate dependably and securely. Having a strong sense of one's surroundings is crucial for many types of mobility and work environments in order to identify safe zones and prevent accidents.

Multimodal sensors are required for this, in addition to data analysis techniques like machine learning and image processing. Another approach uses visual locality determination techniques or other positioning data from satellites to determine the location on a particular map. On the other hand, map data is frequently lacking or unavailable. Research has concentrated on creating sensors and techniques for autonomous systems in a variety of application domains, location determination and mapping for support systems, and safe surroundings detection [26].

7.1 Sensor Technologies for Perception

In an artificial system, the notion of perception gives an alternative method of awareness by categorizing the pertinent aspects of an event that occurs in the surroundings [27]. The multimodal sensor data from cameras (RGB, IR, and multispectral), laser scanners, radars, time-of-flight, or ultrasonic sensors provides the basis for the perception of the environment. In order to provide techniques for data processing, analysis, and fusion that can extract pertinent information for tasks such as object categorization, tracking, localization determination and orientation, collision avoidance, and scene understanding at a higher level of abstraction, researchers performed study. In addition to 2D mapping of the environment based on point clouds produced by laser scanners, stereo camera systems, or radar sensors, 3D modeling of the surroundings is necessary for location determination, navigation, and proper item handling by machines (such as cranes or forklifts). Adverse weather conditions, such as persistent rain, haze, soil, or dust, provide a considerable obstacle. Light-based sensors cannot handle such situations. Therefore, in order to preserve crucial aspects of vision, we must substitute physical modalities such as radar. Every sensor modality has advantages and disadvantages, and combining data is required to make up for the disadvantages.

7.2 Object Recognition and Localization Methods

Accurately locating oneself and maintaining direction on an environment reference map is a prerequisite for numerous support systems as well as the dependable operation of automated machinery or vehicles. GNSS, or satellite navigation systems, commonly offer this data. Accuracy can be improved by using correction signals from reference stations supplied by the Earth. Satellite masking makes GNSS alone unreliable, especially in metropolitan areas, mountainous locations, or on corporate property. As a result, other techniques like SLAM (simultaneous localization and mapping) or local radio-based systems must be added to or replaced. For example, for vehicles and machines, precise mapping with semantic content segmentation and dynamic updating is necessary to record object-specific navigation and operational contexts. The maps can be made and altered by sensors like cameras, laser scanners, or imaging radar sensors that are mounted on equipment, cars, or infrastructural structures on the property [28].

7.3 Challenges in Robot Perception

Robots' issues in perceiving the world are not unlike from those faced by humans. First off, robots may meet scenarios where their sensors may not function as well as they should, just like humans do when they cannot see in the dark. This kind of sensory deterioration in a robotic system can be brought on by any circumstance that exceeds a sensor's operational thresholds. Whatever the reason, this leaves one without the knowledge required for accurate perception.

Challenges can develop in repetitive situations, where a lack of environmental distinguishing features can lead to confusion and make it difficult for the robot to correctly detect its location. We refer to this as perceptual aliasing when the surroundings have a maze-like appearance or resemble locations the robot has visited previously. Similar to this, when a robot receives insufficient data from its sensors to determine whether or not it is moving in relation to another item, it may experience perceptual ambiguity. According to theory, observability is the system's capacity to precisely determine its state via sensor observations.

8 Conclusion

The fundamentals of robotics are a broad field that requires both theoretical knowledge and practical experience to apply effectively. Simply put, robotics encompasses everything from robot conception to mechanical design, manufacture, and control, enabling them to perform a sequential set of tasks autonomously or semi-autonomously. Robotics is now being used in many industries, including atomic energy, automotive, defense, pharmaceutical, and textile. According to experts, there could come a time when robots surpass human intelligence. Currently, robots are able to bend an elbow, rotate the wrist, and rotate the base of an arm. It may eventually seem, feel, and behave like a human. The robot can react to its environment naturally because of its realistic-looking skin and hair.

References

1. Grimheden M, Hanson M (2005) Mechatronics—the evolution of an academic discipline in engineering education. Mechatronics 15(2):179–192
2. Wong C, Yang E, Yan XT, Gu D (2017). An overview of robotics and autonomous systems for harsh environments. In 2017 23rd International Conference on Automation and Computing (ICAC) (pp. 1–6). IEEE.
3. Bolton W (2003) Mechatronics: electronic control systems in mechanical and electrical engineering. Pearson Education
4. Ollero A, Boverie S, Goodall R, Sasiadek J, Erbe H, Zuehlke D (2006) Mechatronics, robotics and components for automation and control: IFAC milestone report. Annu Rev Control 30(1):41–54
5. Poole HH (2012) Fundamentals of robotics engineering. Springer Science & Business Media
6. Stone WL (2018) The history of robotics. In: Robotics and automation handbook (pp. 8–19). CRC Press
7. Brooks RA (1991) New approaches to robotics. Science 253(5025):1227–1232
8. https://www.seminarsonly.com/electronics/Robotics.php
9. Badaan SR, Stoianovici D (2011) Robotic systems: past, present, and future. Robotics in genitourinary surgery. Springer, London, London, pp 655–665
10. Melchiorri C, Kaneko M (2016). Robot hands. Springer Handbook of Robotics 463–480
11. Dobra A (2014) General classification of robots. Size criteria. In: 2014 23rd International Conference on Robotics in Alpe-Adria-Danube Region (RAAD) (pp. 1–6). IEEE
12. Müller-Wilderink H (2021) Automation of offline programming for assembly and welding processes in CATIA/DELMIA using VBA
13. Niku SB (2020) Introduction to robotics: analysis, control, applications. John Wiley & Sons
14. Williams B (2019) An introduction to robotics. Ohio University, Mechanical Engineering
15. McInnis BAYLISSC, Liu CK (1986) Kinematics and dynamics in robotics: a tutorial based upon classical concepts of vectorial mechanics. IEEE Journal on Robotics and Automation 2(4):181–187
16. Gasparetto A, Boscariol P, Lanzutti A, Vidoni R (2012) Trajectory planning in robotics. Math Comput Sci 6:269–279
17. Ghazaly MM, Teo TH, Vivek A, Regeev L, Kartikesu A, Vijayan L, Ali MAM (2016) Point-to-point (PTP) control performances of an upper limb robotic arm. Jurnal Teknologi 78(7)
18. Uchiyama T, Terada H, Mitsuya H (2010) Continuous path control of a 5-DOF parallel-serial hybrid robot. J Mech Sci Technol 24:47–50

19. Gupta A, Arora S, Westcott J (2016) Industrial automation and robotics: An introduction. Berlin, Boston: Mercury Learning and Information. https://doi.org/10.1515/9781683922896
20. Bär GF, Weiß G (2006) Kinematic analysis of a pentapod robot. J Geom Graph 10(2):173–182
21. Kosuge K, Furuta K (1985) Kinematic and dynamic analysis of robot arm. In: Proceedings. 1985 IEEE International Conference on Robotics and Automation vol. 2, pp. 1039–1044. IEEE
22. Zhou Z, Xiong R, Wang Y, Zhang J (2020) Advanced robot programming: A review. Current Robotics Reports 1:251–258
23. Gorinevsky DM, Formalskii AM, Shneider AY (1997) Force control of robotics systems
24. Salem FA (2015) The role of control system/algorithm subsystems in mechatronics systems design. J Multidiscip Eng Sci Technol (JMEST) 2(10): 2717–2732, *ISSN: 3159–0040, JMESTN42351005*
25. Han S, Kim MS, Park HS (2012) Open software platform for robotic services. IEEE Trans Autom Sci Eng 9(3):467–481
26. Cadena C, Carlone L, Carrillo H, Latif Y, Scaramuzza D, Neira J, Leonard JJ (2016) Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Trans Rob 32(6):1309–1332
27. Biel L, Wide P (2000) Active perception for autonomous sensor systems. IEEE Instrum Meas Mag 3(4):27–30
28. Kefalea E (1998) Object localization and recognition for a grasping robot. In: IECON'98. Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society (Cat. No. 98CH36200) vol. 4, pp. 2057–2062. IEEE