

Available online at: <http://ajeet.ft.unand.ac.id/>

Andalas Journal of Electrical and Electronic Engineering Technology

ISSN 2777-0079

AJEET
 Andalas Journal of Electrical and Electronic Engineering Technology

A Compact Review of Industrial Robots: Dynamic Modeling, Control Strategies, and Operational Challenges

Benjamin Kommey, Stephanie Essah, David Dery Kuusofaa, Samuel Boahene Jnr

Department of Electrical and Electronics Engineering, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana

ARTICLE INFORMATION

Received: April 11, 2025

Revised: September 8, 2025

Accepted: November 14, 2025

Available online: November 30, 2025

KEYWORDS

Real-Time Control, Kinematics, Dynamic Modeling, PID Control Algorithm, Sparse Identification of Nonlinear Dynamics

CORRESPONDENCE

E-mail: bkommey.coe@knust.edu.gh

A B S T R A C T

When considering Industrial Robots, one often envisions large-scale mechanical systems integrated within manufacturing facilities, specifically designed to transport objects across assembly lines, perform complex assembly tasks, and handle material manipulation with high precision. These robots are capable of performing repetitive and demanding operations, but if proper safety protocols are not implemented, they can pose serious safety hazards to human workers. While these perceptions are accurate, they do not fully capture the complexity and extensive scope of industrial robotics. This review, supported by an exhaustive analysis of numerous technical reviews, academic papers, and research studies conducted over multiple years, synthesizes these findings to provide a comprehensive and sophisticated understanding of industrial robotics. It covers the intricacies of engineering design, including kinematic and dynamic modeling, as well as the development of advanced algorithms and control systems that enable precise, real-time operations. Additionally, it highlights the critical role these robots play in boosting productivity, ensuring operational efficiency, and maintaining consistency in manufacturing processes. With recent advancements in artificial intelligence, new perspectives have emerged regarding control and modeling strategies, paving the way for the development of cognitive robots capable of autonomous decision-making and adaptive behavior, thus opening exciting prospects for the future of industrial automation.

INTRODUCTION

Although the history of industrial robotics is conventionally traced back to the 1960s, several pivotal developments in automation occurred prior to this era. Notably, Pollard and Roselund's invention of a "programmable" paint-sprayer device in 1938 and Goertz's creation of a tele-operated "manipulator" in 1949 set important precedents. However, the true turning point for industrial robotics was marked by George Devol's ingenuity. In 1954, Devol designed a "Programmable Article Transfer" device, as described in his patent request. This invention became the foundation for Unimate (Figure 1), which is regarded as the first true industrial robot in history [1]. These automated systems have since undergone significant evolution, becoming integral to numerous sectors. The advancement of industrial robots has significantly transformed manufacturing across various sectors, including automotive, aerospace, electronics, and healthcare. This technological evolution has given rise to a diverse range of robot types, including traditional, intelligent, mobile, and collaborative robots. These systems are pivotal in modern manufacturing due to their capability to augment performance, enhance safety, and reduce operational costs.



Figure 1. Unimate robot [101]

As industries strive for increased efficiency and productivity, the integration of robotics has become a paramount strategy. Industrial robots extend beyond manufacturing environments, finding applications in logistics, agriculture, and healthcare. Examples include wheeled robots for warehouse inventory management, robotic surgical arms, and autonomous mobile robots for agricultural tasks. Industrial robots are classified based on design and application, encompassing articulated robots, Selective Compliance Assembly Robot Arm (SCARA) as shown in Fig. 2, delta robots, Cartesian robots, cylindrical robots, polar robots, and autonomous mobile robots [2]. Each category is tailored for specific operations such as material handling, welding, assembly, painting, and inspection. Key performance metrics for these robots include degrees of freedom, payload

Table 1. Comprehensive overview of industrial robots

Generation	Years	Features	Challenges	Benefits	Reference
First	1950-1967	Hydraulic and pneumatic systems, Limited programmability, Heavy lifting	High costs, Inflexibility, Safety concerns	Enabled automation of heavy and repetitive tasks, Reducing human labor and Increasing production efficiency	[5]
Second	1968-1977	Introduction of microprocessors, Improved control systems, Materials handling	Lack of standards, Workforce skills, Managing workflow	Improved precision and flexibility in the manufacturing process, Facilitated the development of more complex automated systems	[6]
Third	1978-1999	Advanced controls, Miniaturized motors, Increase precision, reprogram ability	High initial investment, Complex programming, Limited payload capacity	Enhanced accuracy and efficiency, Reprogram for different tasks, Significantly reduced production cost over time	[7]
Fourth	2000-date	Integration of AI, Collaborative robots, Enhanced sensors, Autonomous operation	Communication in robot swarms, Energy efficiency, Ethical concerns	Increased adaptability and safety, Collaborative capabilities with humans, Real-time data processing and decision making, High energy efficiency	[8]

capacity, reach, precision, sensor integration, and sophisticated control algorithms, enabling them to execute complex tasks with high accuracy and reliability.

Dynamic modeling and control strategies, which are the stars of the show, are integral to the efficacy of industrial robots. Dynamic modeling involves developing mathematical representations of a robot's physical behaviors and interactions with its environment. These models facilitate predictions of system responses to various inputs and support the design of effective control systems [3]. Control strategies—such as PID (Proportional-Integral-Derivative) control, adaptive control, and robust control—are deployed to ensure precise, stable, and efficient robot movements. Through the application of dynamic models and control strategies, industrial robots achieve superior performance and adaptability to diverse operational conditions, rendering them indispensable in modern industrial applications [4]. The development of industrial robots spans several generations, each marked by distinct features, challenges, and benefits as illustrated in Table 1.

The primary objective of contemporary research in industrial robotics is to develop advanced dynamic models that accurately represent the behavior of robots under various operational conditions. This encompasses the design of sophisticated control algorithms capable of managing the intricacies of robotic motion while maintaining high performance levels. Additionally, there is a significant exploration into the application of machine learning techniques to enhance the adaptability and efficiency of robotic systems in real-world scenarios [9][10].



Figure 2. SCARA AdeptoOne robotics [102]

METHOD

While industrial robots have traditional applications within manufacturing environments, their utility is expanding into various fields beyond industry. Their capability to move autonomously and be easily controlled renders them suitable for a multitude of tasks across different sectors. A thorough understanding of the kinematics (movement) and dynamics (forces) involved in robot operations is crucial for effective path control. Prior studies have employed complex mathematical methods to accurately model these aspects, as illustrated in Appendix 1.

Case studies on notable areas of focus are on holonomic robots, which can move in any direction utilizing specialized wheels known as mecanum wheels. Research in this domain has introduced innovative modeling techniques based on virtual work principles, significantly enhancing the path-following capabilities of these robots. Simulations have validated the effectiveness of these models in improving robotic performance. With the

adoption of Wheeled Mobile Robots (WMRs) over the last few decades, there has been a widespread adoption of WMRs across various fields. Critical considerations in the design of these robots include autonomy, mobility, positioning accuracy, and control mechanisms. Kinematic and dynamic models are essential for achieving precise trajectory tracking in WMRs. Researchers have employed Lagrangian formulations to derive dynamic equations that govern the motion of these systems.

In the methodology of this study, a comprehensive analysis of previous studies and research papers was undertaken to understand the state-of-the-art in dynamic modeling and control of industrial robots. Mathematical models and frameworks, such as Lagrange's equations and Newton-Euler methods, were used to develop dynamic models. Simulations were performed to validate the developed models and control strategies, ensuring their efficacy in real-world scenarios. Additionally, the integration of machine learning techniques was explored to enhance the dynamic modeling and control process. In addition, the integration of artificial intelligence, machine learning, and computer vision in industrial robots enables them to adapt to changing environments within traditional manufacturing applications [10].

Challenges in dynamic modelling and control include nonlinearities, uncertainties, real-time adaptation, and integration with smart manufacturing technologies. This comprehensive overview aims to develop accurate dynamic models, design advanced control algorithms, investigate machine learning applications, and evaluate performance through simulations and experimental validation, contributing to next-generation industrial robots with enhanced precision, improved safety, increased energy efficiency, and seamless integration [11]. The robot modeling and design processes are illustrated in a flowchart format in Figure 3.

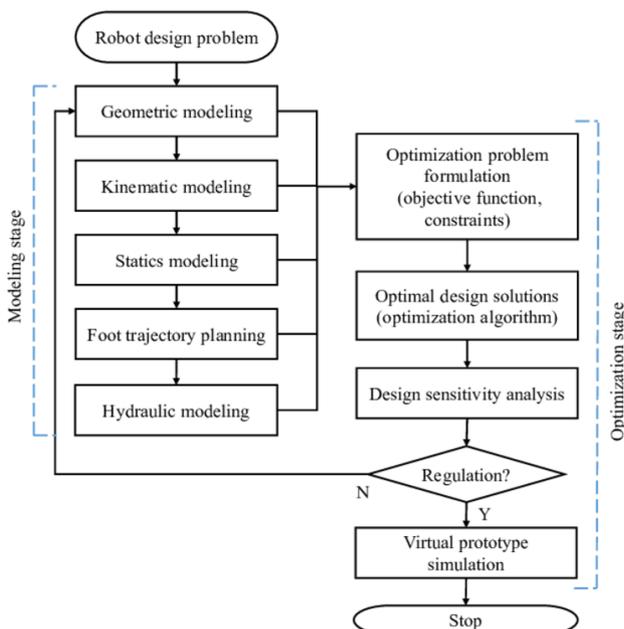


Figure 3. Robot modelling and design process [12].

In recent years, robotic systems have expanded their applications to include domestic, medical, and industrial settings. There is also an increasing demand for robots with high speed and high

precision. Particularly for the manufacturing of large-scale machinery, industrial robots often have low positioning accuracy, which limits their ability to meet the accuracy requirements and hinders the development and manufacture of robots.

The key to solving this problem is to establish an accurate dynamic model of the robot, enabling precise control and enhanced accuracy. Therefore, the study of dynamic parameter identification of IRs is important both in theory and in application [13]. To accurately describe robot motion, it is crucial to develop either the appropriate inverse kinematics or dynamic model. The dynamic model enables consideration of various factors, including mass, moment of inertia, friction forces, centrifugal force, torque, and other physical properties. These models are constructed to gain a deeper understanding of the structure and operation of future mechatronic products. The development of such models becomes increasingly important when dealing with complex systems. It is generally less expensive and simpler to modify virtual prototypes compared to improving physical prototypes that already exist [14].

As shown in Appendix 2, various dynamic modeling strategies have been developed and deployed over the years. The applications, solutions, and possible limitations are also examined [36-41]. While paper [42] identified the stiffness parameters by using the measured natural frequencies. They were then updated using the method of eigen-sensitivity analysis. The frequency content of the measured vibrations demonstrated that the response is dominated by the first three natural frequencies, and the updated dynamic model accurately predicted these frequencies with a precision of 99.04%. According to [43], the Newton-Euler and the Lagrange-Euler algorithms are used to determine the dynamic model of the ABB IRB 120 manipulator. Recognizing the challenge associated with collecting a large amount of data, they resorted to creating a 2000-point dataset including joint positions, angular speeds, angular accelerations, and the corresponding torques.

ML, specifically Multilayer Perceptron Learning (MLP), Artificial Neural Network (ANN), was used to model the dynamics using the synthetic dataset. Separate MLPs were used for the individual joint torques and for the total torque, employing a randomized search for hyperparameter tuning. The trained models were evaluated using the Coefficient of Determination (R²) and the Mean Absolute Percentage Error (MAPE). All individual joint torques had above 0.9, with the first four above 0.95, for the R² and all individual joint torques had MAPE below 2%. The total torque, however, had an R² of about 0.89 and a slightly higher than 2% MAPE.

The Lagrangian is used in [44] to build the robot dynamics equation. After establishing the Lagrangian mathematical model, the robot's numerical value is entered into the Lagrangian method to create the dynamic equation's matrix form. The following is obtained by analyzing, storing, and solving the inertia matrix and expressing the interactions between different joints using a Lagrangian equation:

$$T_i = \left(\sum_{j=\max(i,k)}^n \text{Trace} (U_{jk} J_j U_{ji}^T) \right) \ddot{q}_i(t) + \dot{\theta}^T H_{i,v} \dot{\theta} + \sum_{j=1}^n (-m_j g U_{ji}^T \bar{r}_j) \quad (1)$$

The establishment of an automatic analysis model for mechanical system dynamics involves deriving the robot dynamics equation and solving the dynamic inverse problem, as well as employing a single-joint modeling approach and co-simulation with multiple software packages to obtain a function that depends on acceleration and velocity. As an alternative to the current analytical and numerical reduction strategies, [45] suggested a novel analytic approach for the model reduction based on Fourier series development. It accomplishes this by taking advantage of the fact that each term in the robot's torque vector can be expressed as a linear combination of a function that depends on position and that has benefits like eliminating the need for case-by-case analysis, which makes it easier to automate and more generalizable, as well as identifying a smaller number of crucial factors than numerical methods, which makes the calibration process more effective.

Paper [45] introduces the concept of local dynamic model calibration, demonstrating how ad hoc calibration can enhance the dynamic model's torque forecast accuracy. Throughout three distinct workspace sub-regions, the normalized RSME (Root Mean Square Error) remained consistently below 6%. The modeling and identification of the elasto-dynamics model, where identification procedures can be defined, were covered, and a workable application of the joint stiffness matrix has been presented based on the findings. In axes 1 and 2, the error between the two inertia values was less than 7%; however, in axis 3, it exceeded 16%. The use of a locally calibrated rigid body dynamics model to create a virtual force sensor whose operation relies on measuring extra torques is covered in [45]. In simple terms, different dynamic modeling approaches for robotic systems vary in accuracy, with natural frequency methods achieving up to 99.04% accuracy, while MLP ANN models perform well for individual joints but slightly worse for total torque prediction. This variation in accuracy underscores the importance of selecting appropriate modeling methods for specific industrial applications to ensure optimal performance and efficiency in robotic systems. Table 2 contains several approaches and their performance.

Table 2. Dynamic modelling and performance

Approach	Metrics	Performance
Natural Frequencies	Accuracy	99.04%
MLP ANN (Individual Joints)	R ²	>0.95
MLP ANN (Individual Joints)	MAPE	<2%
MLP ANN (Total Torque)	R ²	0.89
MLP ANN (Total Torque)	MAPE	>2%
Local Calibration	RSME	<6%
Inertia Error (Axes 1&2)	Error	<7%
Inertia Error (Axis 3)	Error	>16%

MLP ANN (Multilayer Perceptron Artificial Neural Network) refers to a type of feedforward neural network composed of multiple layers of interconnected nodes (neurons) that process

and transmit information through the network. MLPs are widely used in machine learning applications due to their ability to learn complex patterns in data and make predictions or classifications based on that learning [46].

RESULTS AND DISCUSSION

The inaugural industrial robots were initially controlled by rigidly programmed logic circuits and relay-based systems. These early control mechanisms exhibited limited flexibility and necessitated extensive manual reconfiguration for varied tasks. While suitable for straightforward, repetitive operations, this approach fell short in accommodating the complexity demanded by modern manufacturing processes [47]. The evolution of control strategies in industrial robotics has been characterized by several pivotal developments: The eighties saw the Introduction of microprocessors and computer-based control systems, enabling the implementation of more intricate and adaptable control algorithms. The development of PC-based control systems and specialized robotic simulation software facilitated the emergence of offline programming and virtual commissioning procedures that began in the 1990s. This was followed by the integration of networked control systems and fieldbus technologies, enhancing intercommunication between robots and other devices within the manufacturing environment. Currently, AI and machine learning are being incorporated into control strategies, enabling robots to autonomously learn from data and progressively enhance their performance over time [48].

The control strategies for industrial robots have consequences such as enhanced versatility, improved accuracy, predictive maintenance, and collaborative robotics (cobots). Therefore, contemporary control systems enable robots to adapt to diverse tasks without extensive reprogramming, sophisticated control algorithms contribute to increased precision and consistency in robotic movements, AI-driven control systems can forecast equipment failures and schedule maintenance to minimize downtime, and control strategies have facilitated the development of robots that can safely coexist with human workers [49]. Industrial robots equipped with advanced control systems are utilized across numerous industries such as manufacturing where tasks such as welding, painting, and assembly benefit from precise control, healthcare where surgical robots and rehabilitation robots rely on sophisticated control systems, logistics where automated guided vehicles and sorting systems require accurate control strategies an agriculture where robots for planting, harvesting, and crop monitoring employ sophisticated control algorithms [50].

Different control strategies exhibit varying levels of flexibility, precision, and ease of implementation as illustrated in Table 3.

Table 3. Comparative analysis of control strategies [51]

Control Strategy	Flexibility	Precision	Ease of Implementation
Hard-wired Logic	Low	Moderate	High
Microprocessor-based	Moderate	High	Moderate
.PC-based Control	High	High	Moderate
AI/ML-based	Very High	Very High	Low

Best practices for implementing advanced control strategies are assess task requirements: understand specific needs, including precision, flexibility and speed, evaluate system compatibility: ensure compatibility with existing systems and equipment, consider future scalability: choose strategies that can be easily adapted for future needs, invest in training: provide adequate training for operators and maintenance personnel and monitor and optimize: continuously monitor performance and adjust as needed [52].

Application of industrial robot controls

[53] presents dynamic control through the implementation of impedance control strategies, another form of advanced control strategies. It utilizes two industrial robots involved in a peg-in-hole assembly, where one of the robots is controlled to behave as a mechanical impedance. This enables the robot to adapt to environmental variations and compensate for errors. Feedback control loops also monitor the end-effector's orientation and position by responding to errors in real-time, enhancing the robustness of the control system against unmodeled dynamics and disturbances. A quaternion-based approach is also employed to manage the rotational aspect of impedance control, thereby maintaining a geometrically consistent control strategy. [54] states that impedance control reduces mating forces and moments, allowing robots to handle misalignments in the task environment.

In reference [55], the Kinect v2 module was developed to facilitate both offline and online control of an industrial robot through gestures and voice commands, thereby enabling dynamic adaptation. The sensor was employed to capture gesture inputs, while a microphone recorded voice commands. Additionally, a two-threaded application was created in RABID to allow real-time control of the industrial robot. The proper functioning of the application was tested offline using a digital twin, with verification also conducted online on the actual test station. However, noise interference posed a challenge, with the study noting a recognition rate of 91.43% for a 10 dB difference between the command and background noise.

Reference [56] introduces a novel approach to dynamic control within a multi-arm robotic system by implementing adaptive control strategies that treat cross-coupling effects as disturbances to be compensated for, rather than relying on complex mathematical models of arm dynamics or load parameters. The adaptive control laws address both position and force control, enabling the system to adjust controller gains in real-time based on the manipulator's performance in Cartesian space. These laws include a position-position control law where both arms operate under position control with adaptive controllers ensuring end-effector position tracking despite unknown interaction forces; a position-hybrid control law in which one arm employs position control while the other uses hybrid position/force control, allowing for simultaneous management of end-effector motion and applied force on the load; and a hybrid-hybrid control law where both arms utilize hybrid position/force control to manage both position and force on the load concurrently.

In [57], a Model Reference Adaptive Control (MRAC) approach is proposed for the trajectory following control of the robotic manipulator. This is because the MRAC approach is a robust

control strategy that effectively handles distances and non-linearities. A PI-like non-linear control law based on the MRAC approach, which considers the robot's non-linear and actuator dynamics, is developed. This control law was determined by the hyperstability approach of Popov. Simulation results demonstrate the effectiveness of the proposed MRAC algorithm, showing accurate trajectory following and smooth control voltages. The dynamic control of industrial robots, particularly in relation to trajectory tracking, is the main topic of [57]. It is evident that designing control systems is significantly hampered by the dynamic nature of robot arms. These difficulties are caused by non-linear forces resulting from the interaction of the motion of several links, time-varying inertia experienced by each joint as the robot arm moves, and variations in gravitational load on each link. To assess and improve the control algorithm before deploying it on the actual robot, this document describes a procedure for creating an automatic analysis model to simulate robot arm dynamics. Co-simulation using MATLAB and ADAMS, single-joint modeling, and dynamic equation modeling are all steps in the process.

Industrial robots' software and simulation

The proposed method in [61] aims to serve as a versatile tool that can be easily integrated with various industrial robots. To ensure compatibility across multiple robot brands, this approach incorporates libraries and frameworks for robot simulation, motion planning, and control that are commonly used in both industry and research. The method focuses on typical activities of a robotic cell programmer. The dual-arm robot used in this study features two different arms, each equipped with different tools mounted at various orientations relative to the robot's base. Assembly parts have their unique handling requirements [58][61]. Initially, in Process Design and Simulation, the engineer designs the robot process by setting positions in Cartesian space using a VR interface. The designed process is then transmitted to the robot simulation, which checks for possible collisions and generates the robot's trajectory and control schemes. The robot simulation is connected to the VR environment, providing immediate feedback to the user regarding the new robot process. The physical robot only executes the process after validation by the engineer [59][61].

The near real-time control method addresses the challenges of real-time teleoperation, which typically require additional compensation for jerkiness caused by unsteady hands or tracking losses, as well as the need for a high-speed network. A TCP connection is recommended for this near-real-time communication, as it ensures the delivery of an extensive robot process without packet loss, despite a slightly higher delay compared to a UDP connection. The engineer does not have real-time control over the physical robot. Instead, this closed-loop system of process design, execution in a simulated environment, and immersive visualization of the robot's motion provides a realistic experience for the engineer. The trajectories generated by the simulation are calculated using a motion planner integrated into the simulation framework, which includes tools for environment specification, collision detection, visualization, and integrated motion planning algorithms for robot kinematics. The motion planning algorithm used is the Open Motion Planning Library (OMPL) [60].

This approach reduces the time required to program a complete robot procedure. Although multiple iterations of replanning and simulation execution may occur, these interactions happen rapidly. The robot simulation and control module can also host supplementary sensory equipment to assist the engineer during the process design. A vision system automatically localizes the parts to be manipulated, eliminating the need for precise manual positioning by the engineer. For example, in a pick-and-place process, the engineer only specifies the part to be grasped, and the robot automatically handles localization and grasping. The vision system also provides an accurate representation of the environment and various components within the simulation environment [60][61].

Industrial robot programming languages

Industrial control systems employ a range of programming languages, each serving distinct purposes. At the core are specialized languages within the IEC 61131-3 standard, including Ladder Logic, Function Block Diagram, and Structured Text. These were crafted specifically for industrial control and remain prevalent in current practices. However, there has been a noticeable shift towards utilizing more versatile, general-purpose programming languages, such as C++, Python, Java, and C#. These newer languages offer benefits, including access to broader skill sets, robust development tools, and adaptability. Modern approaches often integrate these traditional IEC 61131-3 languages with higher-level programming languages, such as Python or C++. This hybrid approach capitalizes on the strengths of both paradigms - graphical interfaces for troubleshooting and textual languages for complex tasks. The selection of a programming language depends on the specific requirements of each application. While traditional ladder logic remains favored for straightforward control functions, more sophisticated applications are increasingly relying on contemporary programming languages. This evolution reflects the industry's need for flexibility and innovation in addressing complex automation challenges [62][63].

The software architecture leverages Unity for visualizing the simulation environment and VR interface. The robot connection and control functionalities are based on the Robot Operating System (ROS) to ensure compatibility with a wide range of robots. ROS acts as middleware to connect with hardware devices and develop robot control capabilities. Integration of the VR interface with the robot and vision system modules is achieved via a ROS bridge. Unity connects to the ROS bridge server using the ROS Sharp plugin, which models all messages implemented in ROS. This enables engineers to remotely access production shop floor pipeline information and calculate kinematic motions for multiple robots. Information is transmitted to and from the robot controller through a TCP/IP interface [64][65]. A Comau dual-arm manipulator was chosen for this implementation due to its capability to handle a wide range of processes. Controlling a dual-arm robot is complex, akin to managing two individual manipulators [66]. Appendix 3 contains a comparison of industrial robots' controls and applied strategies.

Simulation and synthetic data in industrial robotics

Simulations and synthetic data play crucial roles in developing smart industrial robotic systems across various stages and

aspects. These tools offer several advantages, including generating large amounts of low-cost data, accelerating design cycles while reducing costs, and providing a safe, controlled testing environment [86]. However, implementing simulations and synthetic data also presents challenges that need to be addressed. One significant challenge lies in bridging the gap between simulated environments and real-life scenarios. This discrepancy often results in decreased precision when systems are deployed in real-world settings. Factors contributing to this issue include physics simulations, virtual representations of objects, recreation of sensor data, artificial lighting, and other elements mimicking real-world conditions [86]. Machine learning-based approaches play a crucial role in the control of smart industrial robots, as they are widely used to tackle complex tasks. According to research, more than 80% of the time spent on AI projects is dedicated to data collection and processing [87]. Various data collection techniques exist, ranging from openly available datasets for autonomous driving and traditional applications, such as household object recognition, to more specialized methods for specific industries [88][89][90].

Smart manufacturing presents unique challenges due to the rapid change in product varieties and the need for algorithms trained on new datasets. This scenario limits the reusability of existing datasets and makes manual labeling methods inadequate due to their time-consuming, expensive nature and potential for human error [91]. Synthetic data generation addresses these issues by utilizing physics and graphics engines to replicate real-life situations, including environmental uncertainties [92]. However, the benefits of synthetic data may not outweigh the reality gap issues in controlled environments and non-complex scenes. To address this, combining real and synthetic data could potentially leverage the efficiency of data collection processes and the precision of trained models. Yet, comprehensive studies on this approach in manufacturing applications are still lacking [92]. In scenarios involving self-learning robots, training in real life while interacting with the environment and using feedback to learn policies presents another set of challenges. This approach can be tedious, expensive, and time-consuming, especially when considering the risks associated with exploratory actions in unstructured environments [86]. These challenges highlight the ongoing efforts to bridge the gap between simulated environments and real-world applications in smart industrial robotics. Researchers continue to explore innovative solutions to address these issues and improve the accuracy and efficiency of robotic systems in manufacturing settings.

VR interface and simulation environment implementation

The simulation environment runs on a PC located on the shop floor, directly communicating with the ROS-based robot simulation and control module. This environment handles communication with the engineer's VR interface, acting as a server. The VR interface functions as a client, providing a window for the user to view both the simulation and the actual shop floor scenario [71][72]. Mirror Networking, an open-source API for Unity, was used to develop this communication channel. This networking API, typically used in MMO games for low-latency synchronization among multiple players, is suitable for teleoperation systems with multiple user instances. An initial empty warehouse was designed as the base simulation environment, capable of hosting various workstations. URDF and

.stl models were imported from ROS to Unity using ROS Sharp functions. The shop floor environment around the robot was designed in CAD software to match physical dimensions, and 3D models were then imported into Unity [73].

Robot programming and control are managed through ROS interfaces and libraries. The MoveIt motion planning framework simulates robot motions to match physical executions, providing collision-free robot trajectory plans. The planning scene, configured to resemble the VR environment, represents the physical setup of the robotic cell. The tf (Transform Frames) library efficiently generates Cartesian trajectories for various assembly operations, tracking relationships between multiple coordinate frames (assembly objects, tools, and robot arms). Actual control of the dual-arm robot is handled by a custom driver developed in accordance with the ROS community interface guidelines, which accommodates various robots [74]. The robot driver operates on a Linux machine, deployed as a ROS node, and interacts seamlessly with the motion planning node and the tf tree. Communication with the robot is established over a dedicated Ethernet network through a TCP connection. Accuracy in robot control is achieved through URDF calibration for each robot. This calibration enables the calculation of unique deviations from the manufacturer's calibration process. Once the URDF matches the physical position, it ensures accurate Cartesian goal planning with the ROS driver [75]. Figure 4 depicts VR imitation learning.

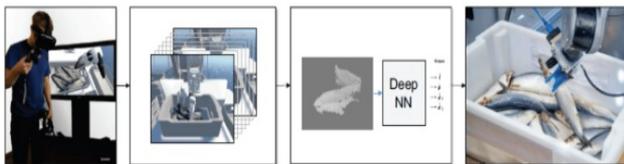


Figure 4. Imitation learning in VR [76].

Smart manufacturing trends

The evolution of smart manufacturing is transforming how new tasks are learned and implemented in factories. As supply chains in smart manufacturing need to adapt, downtime is no longer acceptable for organizations [93]. The COVID-19 pandemic exposed vulnerabilities in automation and had broader economic impacts. However, it also highlighted the benefits of digitization for many factories. The path towards future smart factories is incremental, with acceptance of industrial robots partly dependent on advancements in control methods [93]. The adoption of smart industrial robots is likely to be crucial for productivity growth, as it could reshape global supply chains. A key challenge is striking a balance between human capabilities and automation. To achieve high levels of automation and flexibility, industrial robots should strive for more human-like performance through cognitive robotics [93][94]. If full human-like performance is not possible, collaboration between humans and robots is essential to maximize their potential. This balance extends beyond Industry 4.0 and into the vision of Industry 5.0, where humans and robots work together whenever possible [94].

Industry 5.0 envisions combining human creativity with efficient, smart machines. However, several important issues remain, including acceptance of robots, trust in human-robot collaboration, redesigning workplaces, and proper education and training [94]. Currently, attitudes towards close human-robot

collaboration are ambiguous. However, this may change as more natural ways of communicating with machines develop, and the overall understanding of digitization improves [95]. The future of manufacturing will likely involve greater integration of humans and robots working together effectively. Addressing current challenges and developing more advanced control methods will be crucial for realizing the full potential of smart industrial robots in future factories [96].

Artificial intelligence is revolutionizing industrial robots, granting them advanced capabilities such as autonomous operation, enhanced decision-making, and adaptability to complex tasks. This technology allows robots to process data rapidly, learn from experiences, and optimize workflows, resulting in increased efficiency and reduced operational costs [97]. Key advantages of AI-enhanced robots include increased productivity, whereby robots can operate continuously without fatigue, significantly boosting output [98]; improved precision, whereby AI-driven robots perform tasks with high accuracy, minimizing errors [99]; and enhanced safety, as they can operate in hazardous environments, reducing risks for human workers.

The integration of AI into industrial robots is transforming manufacturing processes across various sectors. These advanced robots are capable of handling complex tasks, adapting to changing circumstances, and working effectively in collaboration with human workers. Their ability to process large volumes of data, learn from experiences, and make informed decisions enables them to optimize production workflows and quality control processes. This leads to improved efficiency, reduced downtime, and enhanced overall manufacturing performance. While AI-driven robots offer numerous benefits, their implementation also presents challenges related to safety, employee training, and cybersecurity. Manufacturers must carefully consider these factors when integrating AI-enhanced robots into their operations. As AI technology continues to advance, we can expect to see even more sophisticated applications of industrial robots in various industries, driving innovation and productivity improvements across the board.

CONCLUSIONS

This study delineates industrial robots and their various types, comparing them by generation based on features, challenges, benefits, and operational durations. It examines dynamic modeling of these robots, highlighting different methods such as stiffness parameter identification, NE and LE techniques, and machine learning with synthetic data, along with their effectiveness through experimental validation. The paper also reviews dynamic control approaches like impedance control. In response to increasing industry demands for performance, flexibility, safety, and cost efficiency—which traditional control methods struggle to meet—it explores smart robotic control and learning strategies to address these challenges. AI-based systems enable robots to perceive, learn, and adapt autonomously, aiming for human-like performance. Despite advancements, challenges like sample inefficiency and generalization persist, necessitating further research. The integration of AI and machine learning signifies a major shift, with ongoing technological refinements promising more sophisticated, adaptable manufacturing robots.

REFERENCES

- [1] A. Gasparetto and L. Scalera, "A Brief History of Industrial Robotics in the 20th Century," *Advances in Historical Studies*, vol. 8, pp. 24-35, 2019, doi: 10.4236/ahs.2019.81002.
- [2] "Modelling and Control of Industrial Robots," *IEEE Trans. Ind. Electron.*, vol. 66, no. 10, pp. 8134-8143, 2019.
- [3] Y. Zhang et al., "Industrial Robots: A Review," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 1823-1832, 2019.
- [4] S. K. Singh et al., "Computer Vision-based Control of Industrial Robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 2, pp. 661-671, 2019.
- [5] Á. E. Eiben and J. Eilers et al., "Robot Evolution: Ethical Concerns," [Online]. Available: [rontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2021.744590/full]. [Accessed: Jan 17, 2025].
- [6] A. E. Eiben, "Real-World Robot Evolution: Why Would it (not) Work?" 2021. [Accessed: Jan 17, 2025].
- [7] D. A. Sofge et al., "Challenges and Opportunities of Evolutionary Robotics," [Online]. Available: [https://arxiv.org/abs/0706.0457] [Accessed: Jan 17, 2025] 2007
- [8] S. Chang et al., "Evolution pathways of robotic technologies and applications in construction," [Online]. Available: [https://scholars.cityu.edu.hk/en/publications/evolution-pathways-of-robotic-technologies-and-applications-in-construction%28a03b5131-0cb1-44e0-bba1-043a5be5f583%29.html?utm_]. [Accessed: Jan 17, 2025] 2022
- [9] A. Dzedzickis et al., "Advanced Applications of Industrial Robotics: New Trends and Possibilities," [Online]. Available: [https://www.mdpi.com/2076-3417/12/1/135]. [Accessed: Jan 17, 2025] 2021
- [10] H. M. Alwan, "Dynamic Analysis Modeling of a Holonomic Wheeled Mobile Robot with Mecanum Wheels Using Virtual Work Method," [Accessed: Jan 17, 2025] 2020
- [11] M. R. Islam et al., "Dynamic Modeling of Industrial Robots using Machine Learning," *IEEE/ASME Trans. Mechatronics*, vol. 24, no. 2, pp. 538-547, 2019.
- [12] P. Hopkinson, "A New Approach to the Development and Maintenance of Industrial Sequence Logic," [Accessed: Jan 17, 2025] 2018
- [13] Y. Li et al., "Dynamic Parameter Identification of Six-axis Industrial Robot Based on Improved Genetic Algorithm," [Accessed Jan 17, 2025] 2022
- [14] M. Stania et al., "Dynamic Modeling of Wheeled Industrial Robots," [Accessed: Jan 17, 2025] 2017
- [15] T.G.Cumming et al., "Closed and Open Systems: Organizational," [Online]. Available: [URL]. [Accessed: Jan 17, 2025].
- [16] G.Kiswanto et al., "Fundamental Aspects in Designing Vibration Assisted Machining: A Review," 2014
- [17] A. Szpicer et al., "Computational Fluid Dynamics Simulation of Thermal Processes in Food Technology and Their Applications in the Food Industry," 2024
- [18] M. P. Groover, "Automation, Production Systems, and Computer-Integrated Manufacturing," [Publisher], year.
- [19] [1]" [Journal Name], vol.[x], no.[y], pp.[xx-xx], year.
- [20] S. Raczynski, "Discrete Event Simulation vs Continuous System Dynamics," 2022
- [21] J. Chen et al., "Convex Polytopic Modeling for Flexible Joints Industrial Robot using TP-model Transformation," in *Proc IEEE Int Conf Inf Autom.*, Hailar, China, July 2014.
- [22] V. Nguyen et al., "Active vibration suppression in robotic milling using optimal control," 2025
- [23] P. Bazzoli, "Modeling and Control for Precision Robotic Machining," 2024
- [24] S. Ricciardi and B. Siciliano, "Robot Dynamics and Control," [Publisher], year.
- [25] J. J. Craig, "Introduction to Robotics Mechanics and Control," [Pearson], 2018.
- [26] R. G. Fenton and R. L. Norton, "Kinematics, Dynamics, and Design of Machinery," [Publisher], year.
- [27] "Robust data-driven dynamic model discovery of industrial robots," [Non-Linear Dynamics], [2024].
- [28] K. Ogata, "System Dynamics: Modeling and Simulation," [Pearson], 2003
- [29] H. K. Khalil, "Nonlinear Control Systems: An Introduction," [Publisher], year.
- [30] S. K. Tso et al., "Optimization Techniques in Robotics," [Journal Name], vol.[x], no.[y], pp.[xx-xx], year
- [31] T. Mori, "Force Control of Robot Manipulators," [Publisher], year.
- [32] Edited by J. M. Martins et al., "Virtual Sensors in Robotics," [Publisher], year.
- [33] P. F. Lima et al., "Experimental validation of model predictive control stability for autonomous driving," [Pergamon-Elsevier Science Ltd, 2018.], vol.[81], no.[y], pp.[244-245], A. Larsson and O. Grönlund, "Comparative Analysis of the Inverse Kinematics of a 6-DOF Manipulator," [Accessed: Jan 18, 2025] 2018
- [34] "Dynamic Modelling of Differential-Drive Mobile Robots using Lagrange and Newton-Euler Methodologies," [Online]. Available: [https://www.hilarispublisher.com/open-access/dynamic-modelling-of-differentialdrive-mobile-robots-using-lagrange-and-newtoneuler-methodologies-a-unified-framework-2168-9695.1000107.pdf]. [Accessed: Jan 18, 2025] 2013
- [35] K. M. Lynch and F. C. Park, *Modern Robotics*, [Online]. Available: [https://hades.mech.northwestern.edu/index.php/Modern_Robotics?utm_]. [Accessed: Jan 18, 2025] 2017
- [36] M. A. Roehrl et al., "Modeling System Dynamics with Physics-Informed Neural Networks Based on Lagrangian Mechanics," [Online]. Available: [https://www.researchgate.net/publication/341785437_Modeling_System_Dynamics_with_Physics-Informed_Neural_Networks_Based_on_Lagrangian_Mechanics]. [Accessed: Jan 18, 2025] 2020
- [37] W. Khalil, "Dynamic Modeling of Robots using Recursive Newton-Euler Techniques," [Online]. Available: [https://hal.science/hal-00488037/document]. [Accessed: Jan 18, 2025] 2011
- [38] A. De Luca, "Dynamic model of robots: Newton-Euler approach," [Online]. Available: [https://www.diag.uniroma1.it/~deluca/rob2_en/06_NewtonEulerDynamics.pdf]. [Accessed: Jan 18, 2025].

- [39] M. Lutter et al., "A Differentiable Newton Euler Algorithm for Multi-body Model Learning," [Online]. Available: [https://arxiv.org/abs/2010.09802]. [Accessed: Jan 18, 2025] 2020
- [40] "Robot Dynamics Lecture Notes - ETH Zurich," [Accessed: Jan 18, 2025] 2017
- [41] H. Jalali and F. Rafieian, "Model Identification and Dynamic Analysis of Ship Propulsion Shaft Lines," *J. Theor. Appl. Vibration Acoust.*, vol. 1, no. 2, pp. 85-95, 2015.
- [42] S. Baressi Šegota et al., "Dynamics Modeling of Industrial Robotic Manipulators: A Machine Learning Approach Based on Synthetic Data," *Math.*, vol. 10, no. 7, p. 1174, 2022, doi: 10.3390/math10071174.
- [43] "An Approach to Dynamic Modelling of Industrial Robots Based on 3D CAD Technique," [Online]. Available: [https://www.researchgate.net/publication/370913183_An_Approach_to_Dynamic_Modelling_of_Industrial_Robots_Based_on_3D_Cad_Technique]. [Accessed: Jan 18, 2025] 2023
- [44] F. Janabi-Sharifi et al., "Cosserrat Rod-Based Dynamic Modeling of Tendon-Driven Continuum Robots: A Tutorial," Available: [https://www.researchgate.net/publication/351298901_Cosserat_Rod-Based_Dynamic_Modeling_of_Tendon-Driven_Continuum_Robots_A_Tutorial]. [Jan 18, 2025] 2021
- [45] J. A. Lôbo Marques and S. J. Fong, "Artificial neural network-based approaches for computer-aided disease diagnosis and treatment," [Online]. Available: [https://oa.mg/work/10.1016/b978-0-323-85751-2.00008-6]. [Accessed: Jan 18, 2025] 2022
- [46] "Force Control of Robotic Manipulators: Theory and Applications," [Online]. Available: [URL]. [Accessed: Jan 18, 2025].
- [47] P. Bilancia et al., "An Overview of Industrial Robots Control and Programming Approaches," [Online]. Available: [https://www.mdpi.com/2076-3417/13/4/2582?utm_]. [Jan 18, 2025] 2023
- [48] J. Arents et al., "Smart Industrial Robot Control Trends, Challenges and Opportunities within Manufacturing," [Online]. Available: [URL]. [Jan 18, 2025] 2021
- [49] C. Zhang et al., "Recent Progress in Robot Control Systems: Theory and Applications," [Online]. Available: [URL]. [Jan 18, 2025] 2022
- [50] A. Mazhar et al., "Robust Control Approaches and Trajectory Planning Strategies for Industrial Robotic Manipulators in the Era of Industry 4.0: A Comprehensive Review," [Online]. Available [URL] [Jan 18 2025] 2009
- [51] P. Bilancia et al., "An Overview of Industrial Robots Control and Programming Approaches," [Online]. Available: [URL]. [Jan 18, 2025] 2013
- [52] "Impedance Control: An Overview," [Online]. Available: [URL]. [Jan 18, 2025] 2014
- [53] F. Caccavale et al., "Six-DOF Impedance Control of Dual-Arm Cooperative Manipulators," *IEEE Trans. Mechatronics*, vol. 13, no. 6, pp. 688–699, Dec. 2008, doi: 10.1109/TMECH.2008.2005043
- [54] W. Kaczmarek et al., "Industrial Robot Control by Means of Gestures and Voice Commands in Off-Line and On-Line Mode," *Sensors*, vol. 20, no. 21, p. 6358, 2020, doi: 10.3390/s2021
- [55] M. Folgheraiter et al., "Adaptive Reflex Control for an Artificial Hand," in *Proc.SYROCO*, 2003
- [56] R.Y.J.Tam and S.C.A.Thomopoulos, "Model Reference Adaptive Control of Robotic Manipulators using a Modified Output Error Method," Southern Illinois University, Carbondale, IL 62901, Tech.Rep.TR-SIU-EE-04-91, 1991.
- [57] S.Tarbouriech, "Dual-Arm control strategy in industrial environments," [Online]. Available: [URL]. [Jan 18, 2025] 2004
- [58] T.Hoebert et al., "A Framework for Enhanced Human–Robot Collaboration during Disassembly Using Digital Twin and Virtual Reality," [Online]. Available: [URL]. [Jan 18, 2025] 2014
- [59] A.Bonci et al., "Robot Operating System 2 (ROS2)-Based Frameworks for Increasing Robot Autonomy: A Survey," [Online]. Available: [URL]. [Jan 18, 2025] 2015
- [60] S. Chen and J. T. Wen, "Industrial Robot Trajectory Tracking Control Using Multi-Layer Neural Networks Trained by Iterative Learning Control," *Robotics*, vol. 10, no. 1, p. 50, 2021, doi: 10.3390/robotics10010050. 2001
- [61] E. Babaian et al., "ROS2Unity3D; High-Performance Plugin to Interface ROS with Unity3d engine," [Online]. Available: [URL]. [Jan 18, 2025] 2005
- [62] "ROS-Sharp (https://github.com/siemens/ros-sharp)," [Online] 2013
- [63] C. Wolf and S. W. Floyd, "Strategic Planning Research: Toward a Theory-Driven Agenda," [Online] 2019
- [64] M. S. Reed et al., "Evaluating impact from research: A methodological framework," *Y4:0*. 2013
- [65] , "[Title]," *Journal Name*, vol x, no y (if applicable), pp xxx-xxx (if applicable), 2012
- [66] S. Tzafestas and G. Stavrakakis, "Model Reference Adaptive Control Of Industrial Robots With Actuator Dynamics," *IEEE Trans. Ind. Electron.*, vol. 19, no. 14, pp. 123-130, Dec. 1986.
- [67] J. Liang, "Research on Industrial Robot Trajectory Tracking Control System based on PID Feedforward Algorithm," 2022.
- [68] M. Balsells et al., "Autonomous Robotic Reinforcement Learning with Asynchronous Human Feedback," [2011]. M. Balsells et al., "Autonomous Robotic Reinforcement Learning with Asynchronous Human Feedback," [2011].
- [69] M. Saveriano et al., "Dynamic movement primitives in robotics: A tutorial survey," [2014].
- [70] A. Norouzi and C. R. Koch, "Robotic Manipulator Control Using PD-type Fuzzy Iterative Learning Control," 2019.
- [71] J. Nayak et al., "Applications of Robotics in Industry Using Advanced Mechanisms," 2020.
- [72] E. Kurdoglu et al., "Development of Augmented Reality Application for Teaching Mechanism and Robotics in ROS," [2012].
- [73] Mirror Networking, "Open source networking for Unity," [Year]. Available: [https://mirror-\[2002\]](https://mirror-[2002]).
- [74] MoveIt Motion Planning Framework, [2012].
- [75] R. D. Atkinson, "Robotics and the Future of Production and Work," Information Technology and Innovation Foundation, Washington, DC, USA, 2019.
- [76] J. S. Dyrstad et al., "Teaching a robot to grasp real fish by imitation learning from a human supervisor in virtual

- reality," *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Madrid, Spain, 2018, pp. 7185–7192.
- [77] T. Brogardh, "Robot Control Overview: An Industrial Perspective," [2012].
- [78] M. G. Heimann, "Model-based Feedforward Control in Industrial Robotics," [2013].
- [79] P. Mesmer and M. N., "Challenges of Linearization-based Control of Industrial Robots with Cycloidal Drives," 2021.
- [80] A. H. M. Shaheen, "Real-time control of industrial robots by multiple microcomputers," 1983.
- [81] G. Ferretti et al., "Decoupling Force and Motion Control in Industrial Robots," *Control Eng. Pract.*, vol. 1, no. 3, pp. 345-357, 1993.
- [82] S. Tzafestas and G. Stavrakakis, "Model Reference Adaptive Control of Industrial Robots with Actuator Dynamics," [2015].
- [83] F. Caccavale et al., "Industrial Robots for Automation," [2016].
- [84] T. Gold et al., "Model Predictive Interaction Control for Industrial Robots," [2017].
- [85] H. Choi et al., "On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward," *Proc. Natl Acad Sci.*, vol. 118, no.e1907856118, 2021.
- [86] Cognilytica, "Data Engineering, Preparation, and Labeling for AI," 2019.
- [87] Y. Roh et al., "A Survey on Data Collection for Machine Learning: A Big Data—AI Integration Perspective," *IEEE Trans.Knowl.Data Eng.*, vol. 33, no .5, pp .1328–1347 , [2020].
- [88] É.N.Laflamme et al., "Driving Datasets Literature Review," arXiv:1910.11968 , [2003].
- [89] J.Vanschoren et al., "OpenML: Networked science in machine learning," *ACM SIGKDD Explor. Newsl.*, vol .15 , no .2 , pp .49–60 , [2013].
- [90] M.Bertolini et al., "Machine Learning for industrial applications: A comprehensive literature review," *Expert Syst.Appl.*, vol .175, p .114820, [2020].
- [91] S.Levine et al., "Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection," *Int.J.Robot.Res.*, vol .37 , no .4-5 , pp .421–436 , [2024].
- [92] J.M.de Vet et al., "Impacts of the COVID-19 Pandemic on EU Industries," European Parliament, Strasbourg, France, [2023].
- [93] R.D.Atkinson, "Robotics and the Future of Production and Work," Information Technology and Innovation Foundation, Washington, DC, USA, [2019].
- [94] K.A.Demir et al., "Industry 5.0 and human-robot co-working," *Procedia Comput.Sci.*, vol .158 , pp .688–695 , [2018].
- [95] C.Bröhl et al., "Human–robot collaboration acceptance model: development and comparison for Germany, Japan, China and the USA," *Int.J.Soc.Robot.*, vol .11 , no .3 , pp .709–726 , [2009]
- [96] A.Lawrence, "AI's Impact on Robots in Manufacturing," 2024
- [97] Innovate Energy Latest News, "How Robots and AI are Redefining Efficiency, Safety, and Sustainability," 2024.
- [98] Redress compliance.com, "Top 10 Real-World of AI Use in Industrial Robots." Available: <https://redresscompliance.com/category/ai/2010>
- [99] N. Kshetri et al., "Opinion Paper: 'So what if ChatGPT wrote it?' Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research practice and policy," vol .71, Aug. 2023, p .102642
- [100] Figure 1 (spectrum.ieee.org): IEEE Spectrum. [Online]. Available: <https://spectrum.ieee.org> [Accessed on January 19, 2025].
- [101] Figure 2 (www.scirp.org): SciRP. [Online]. Available: <https://www.scirp.org> [Accessed on: January 19, 2025].

APPENDICES

Appendix 1. Aspects of industrial robots

Aspect	Description
Dynamic Modeling	A tool to represent and understand the behavior of systems over time, focusing on the interactions and evolution of components.[15]
Time-Dependent Behavior	Captures changes in a system's state due to internal and external factors, such as material movement, machining vibrations, or joint torques.[16]
Types of Systems	Continuous Systems: Variables change smoothly over time, such as fluid flow or thermal processes. Discrete Systems: Events occur at specific times, causing abrupt state changes, like in manufacturing processes or robotic actions.[17][18][20]
Mathematical and Computational Tools	Differential Equations: Used to describe relationships between variables and their rates of change, ideal for continuous systems. Discrete Event Simulation: Models system behavior by simulating individual events and their impacts, suitable for discrete systems. Machine Learning: Neural networks trained on data to predict the dynamic behavior of complex systems, offering a data-driven modeling approach [19][20][35]
Model Calibration and Validation	Identifying Key Parameters: Determining the most influential factors governing the system's dynamics. Estimating Parameter Values: Using experimental data and optimization techniques to ascertain parameter values. Validating Model Predictions: Comparing model outputs with real-world measurements to evaluate accuracy and reliability. Applications in Manufacturing and Robotics: Predicting and Mitigating Vibrations in Machining: Dynamic models can predict and mitigate vibrations in cutting tools and tool holders, enhancing precision and stability. Designing and Optimizing Manufacturing Systems: Analyzing and optimizing production lines and material handling systems to identify bottlenecks, reduce costs, and improve efficiency. Improving Robot Control: Dynamic models are essential for advanced control strategies, enabling compensation for nonlinearities, disturbances, and flexibility, resulting in precise and efficient robotic motion control [21].
Applications in Manufacturing and Robotics	Predicting and Mitigating Vibrations in Machining: Dynamic models can predict and mitigate vibrations in cutting tools and tool holders, enhancing precision and stability. Designing and Optimizing Manufacturing Systems: Analyzing and optimizing production lines and material handling systems to identify bottlenecks, reduce costs, and improve efficiency. Improving Robot Control: Dynamic models are essential for advanced control strategies, enabling compensation for nonlinearities, disturbances, and flexibility, resulting in precise and efficient robotic motion control.[22]
Key Takeaways	Dynamic modeling captures time-dependent system behavior. Various mathematical and computational tools are used for creating dynamic models. Calibration and validation with real-world data are crucial for ensuring model accuracy. Dynamic modeling is vital for optimizing manufacturing systems and improving robotic control.[23]
Highlighted Equations	Angular Velocity of a Link: It represents the rate of change of angular displacement with respect to time. The equation typically describing the angular velocity of a link considering previous link rotation and joint velocity might take a form similar to: $\omega = d\theta/dt = \dot{\theta}$: ω is the angular velocity θ is the angular displacement $d\theta/dt$ is the derivative of angular displacement with respect to time $\dot{\theta}$ is the time derivative of angular displacement Equation describes the angular velocity considering previous link rotation and joint velocity. Linear Velocity of a Link: Similarly, linear velocity represents the rate of change of linear displacement with respect to time. The equation for linear velocity considering previous link motion and joint velocity might be expressed as: $v = dx/dt = \dot{x}$: v is the linear velocity x is the linear displacement dx/dt is the derivative of linear displacement with respect to time \dot{x} is the time derivative of linear displacement This equation accounts for how the linear velocity of a link depends on its own motion and the velocity of the preceding link Equation (21) presents the linear velocity considering previous link motion and joint velocity. These kinematic principles are essential for accurate motion capture and developing effective

Aspect	Description
Dynamic Models, Forces, and Torques	<p>control strategies. They form the foundation for understanding and analyzing the motion of multi-link systems, which is crucial in fields such as:</p> <p>Robotics: For planning and executing precise movements of robotic arms or legs.</p> <p>Computer Animation: For creating realistic character movements in video games and simulations.</p> <p>Biomechanics: For studying human movement and gait analysis. Mechanical Engineering: For designing and optimizing mechanisms and machines. [24-26]</p> <p>Cutting Tool Vibrations: Dynamic modeling predicts and mitigates vibrations in cutting tools, crucial for robotic machining. The model incorporates stiffness parameters of the joints in the holder structure, influencing the tool's dynamic response. Accurate identification of these parameters through experimental measurements enables the model to predict vibration responses during machining.</p> <p>Robot Manipulator Dynamics: Machine learning is employed for dynamic modeling of industrial robotic manipulators, utilizing both Newton-Euler (NE) and Lagrange-Euler (LE) algorithms. These models calculate joint torques by considering various forces and torques, with the NE algorithm determining joint speeds, accelerations, forces, and momenta [27][35]</p>
Importance of Dynamic Modeling	<p>Predicting System Behavior: Allows engineers to anticipate and address potential issues such as vibrations or instabilities.</p> <p>Improving Robot Control: Incorporating dynamic models into control strategies enables compensation for nonlinearities and disturbances, enhancing motion control precision and efficiency.</p> <p>Optimizing System Design: Guides the design and optimization of robotic systems, improving performance, reducing energy consumption, and enhancing safety</p>
Additional Insights	<p>Virtual Force Sensing: Develops virtual force sensors by estimating expected torques based on robot motion and attributing residual torque to external forces acting on the end-effector. [28-32]</p> <p>Model Reduction: Simplifies models while retaining essential characteristics, reducing computational burdens.</p> <p>Friction Modeling: Accurate friction modeling is crucial for capturing robot dynamics and influencing motion control accuracy and stability.</p> <p>Experimental Validation: Comparing model predictions with real-world measurements ensures accuracy and reliability.[33]</p>
The Role of Kinematic Modeling in Industrial Robotics	<p>Kinematics as a Prerequisite for Dynamics: Kinematics deals with the geometry and motion of objects without considering forces. It also involves understanding joint relationships of robots, providing a framework for dynamic models.</p> <p>Denavit-Hartenberg Convention: This convention systematically assigns coordinate frames to each robot link and provides parameters describing their relative positions and orientations, crucial for deriving transformation matrices</p> <p>Forward and Inverse Kinematics:</p> <p>Forward Kinematics: Determines the robot's end-effector position and orientation from joint angles.</p> <p>Inverse Kinematics: Finds the joint angles required to achieve a desired end-effector pose, fundamental for motion planning and control</p> <p>Jacobian Matrix: Relates joint velocities to end-effector velocities, crucial for control and analysis tasks</p> <p>Kinematics for Trajectory Generation: Kinematics is vital for generating feasible robot trajectories within workspace constraints [34]</p>

Appendix 2 Aspects of industrial robots

Model	Developer	Applications
Lagrange's Method	Joseph-Louis Lagrange, publishing <i>Mécanique Analytique</i> in 1788	Used to derive equations of motion for robotic systems. Uses the principle of least action to derive the equations of motion based on kinetic and potential energy. High computational cost for complex systems. Use of symbolic computation and software tools.
Forward Dynamics	Various researchers. Notable contributors include Roy Featherstone and David Orin, who developed efficient algorithms for robot dynamics	<p>Developed efficient algorithms for simulating robotic systems. Involves determining the motion of a system (e.g., a robot or a human body) based on the applied forces and torques. It answers the question: "Given the forces and torques, what will be the resulting motion?". Computational Complexity: Simulating the motion of complex systems can be computationally intensive</p> <p>Model Accuracy: Ensuring the model accurately represents the real-world system can be difficult. Efficient Algorithms: Development of algorithms like the recursive Newton-Euler algorithm has improved computational efficiency.</p> <p>Improved Modeling Techniques: Using more accurate models and incorporating real-world data can enhance model accuracy</p>

Model	Developer	Applications
Inverse Dynamics	Various researchers including Maciej Pikuliński, Paweł Malczyk, and Ronald Aarts, who later worked on data-driven approaches	<p>Developed methods for solving inverse problems using data-driven techniques and machine learning. Determines the forces and torques required to achieve desired motion.</p> <p>Ill-Posed Problems: Inverse problems can be ill-posed, meaning small errors in data can lead to large errors in the model parameters.</p> <p>Data Requirements: Accurate inverse models require high-quality data, which can be difficult to obtain.</p> <p>Data Quality: The accuracy of the model heavily depends on the quality of the input data.</p> <p>Computational Cost: Inverse dynamics modeling can also be computationally expensive, especially for large systems. Data-driven approaches, regularization techniques.</p>
Newton-Euler Method	Various researchers	<p>Developed for solving inverse and forward dynamics problems in robotic systems. Uses Newton's laws of motion, especially Newton's Second Law and Euler's equations to derive the equations of motion for robotic systems.</p> <p>Complexity in solving equations for multi-link robots. Simplified models and computational algorithms</p>
Screw Method	Sir Robert Stawell Ball, Felix Klein and Sophus Lie in the late 19th and early 20th centuries.	<p>Model spatial mechanisms using screw theory. Uses screw theory to model the kinematics and dynamics of robotic systems. Complexity in modeling spatial mechanisms. Use of advanced computational tools.</p>
Recursive Newton-Euler (RNEA)	J. M. Luh, R. E. Paul, and S. H. Shim in 1980	<p>Improved version of Newton-Euler for better computational efficiency. Recursive algorithm to solve the Newton-Euler equations for robotic systems and for calculating the inverse dynamics of robotic mechanisms with tree-like structures. Handling of high degrees of freedom. Parallel processing and optimized algorithms.</p>
Data-Driven Modeling (eg. SINDy)	Steven L. Brunton, J. Nathan Kutz, and Eureka Kaiser. Their work was first published in 2016	<p>uses sparse regression to infer nonlinear dynamical systems from measurement data. The resulting models are inherently interpretable and generalizable. Uses data to derive dynamic models without relying on physical equations. SINDy performs a sparsity-promoting regression (such as LASSO) on a library of nonlinear candidate functions of the snapshots against the derivatives to find the governing equations. Capturing the complex, nonlinear behavior of dynamic systems can be challenging. When the underlying dynamics are not fully understood, it can be difficult to build accurate models. High-quality, sufficient data is essential for building reliable models, but obtaining such data can be difficult. Handling large datasets and complex algorithms requires significant computational power. Additionally, models might overfit the training data, hindering their ability to generalize to new data.</p>
Physics-Informed Neural Networks (PINNs)	Raissi, Perdikaris, and Karniadakis in 2019	<p>use the knowledge of physical laws to guide the learning process of neural networks. Useful in scenarios where data is scarce or noisy. Uses neural networks to solve partial differential equations with physical constraints. Requires significant computational resources; may need fine-tuning. Enhances predictions in complex systems with physical constraints.</p>
Simulink/Multibody Modeling:	MathWorks	<p>Developed as a tool for simulating and modeling dynamic systems. Uses block diagrams to model and simulate dynamic systems. Requires learning curve; can be complex for large systems. Widely used in engineering for system design and analysis.</p>
Deep Reinforcement Learning-Assisted Teaching	Various Researchers	<p>Uses reinforcement learning to optimize robot control. Uses reinforcement learning algorithms to optimize robot control policies. Adaptability in diverse environments. Advanced algorithms for real-time adaptability.</p>
Graph Theory-Based Method	Various Researchers	<p>Uses graph theory to model and analyze dynamic systems. Uses graph theory to model the relationships and interactions within dynamic systems. Complexity in modeling large-scale systems. Use of advanced computational tools and algorithms.</p>
Bio-Inspired Dynamic Modeling	Various Researchers	<p>Inspired by biological systems to develop dynamic models. Uses principles from biology to model dynamic systems. May not be applicable to all engineering systems. Effective for systems where biological principles are relevant.</p>

Model	Developer	Applications
Real-Time Adaptive Modeling	Various Researchers	Developed to adapt models in real-time based on changing conditions. Models that adapt in real-time to changing conditions and data. Requires real-time data processing and adaptation algorithms. Enhances model accuracy and responsiveness in dynamic environments.
Integrated and Sensory Dynamic Modeling	Various Researchers	Combines sensory data with dynamic models to improve predictions and control. Integrates sensory inputs with dynamic models for enhanced predictions. Requires accurate sensory data and integration algorithms. Improves control and prediction in systems with sensory feedback.
Soft Robotics Modeling	Prothick Kumar Shingo et al.	Uses compliant and deformable materials for adaptive forces. Uses finite element methods to model the behavior of soft robotic systems. Non-linear material response and infinite degrees of freedom. Real-time control techniques and advanced materials.
Quantum Computing-Based Modeling	Future Researchers	Use of quantum computing to solve complex dynamic models. Uses quantum algorithms to solve complex dynamic models in robotics. High computational requirements. Development of quantum algorithms and hardware.

Appendix 3. Control and applied strategies of industrial robots [64-71]

Control Strategy	Ref.	Technology	Effect and limitations
Open loop control Close loop control	[64]	An open-loop system operates without using feedback. It follows a predetermined sequence of operations based on the input signal. Typically includes an input device, a controller, and an actuator Uses feedback to compare the actual output with the desired output and adjusts the control action accordingly. Components: Includes sensors, a controller, an actuator, and a feedback loop	These systems are easier to design and implement because they lack feedback mechanisms, making them generally more cost-effective and faster in operation since they don't require processing feedback signals. Lack of Adaptability: Cannot compensate for disturbances or changes in the environment. Accuracy: Less accurate since it doesn't adjust based on output. Reliability: More prone to errors and variations Accuracy: Provides high accuracy by continuously adjusting based on feedback. Adaptability: Can compensate for disturbances and changes in the environment. Stability: Enhances system stability by correcting deviation More complex to design and implement due to the feedback mechanisms. Cost: Generally, more expensive to build and maintain. Latency: Can introduce delays due to the processing of feedback signals
Variable-Gain Iterative Learning Control Iterative Learning Control (ILC)	[65]	Adjusts control gains iteratively based on the performance of previous tasks. Learns from previous executions to improve performance on subsequent ones.	Improves accuracy and efficiency over repeated tasks. Requires extensive data and iterative computation. Enhances performance through learning from repetition. Requires repetitive tasks and data collection.
PID / Adaptive control	[66]	Feedback control loop calculating error value to apply corrections. Adjusts controller parameters in real-time based on system changes and uncertainties	Widely used for maintaining process variables like temperature and speed. Enhances system performance in the presence of uncertainties and varying conditions Can be difficult to tune; may not handle nonlinear dynamics. Requires accurate parameter estimation; can be complex to implement
Point-to-Point Control (PTP) Continuous Trajectory Control (CTC) Computed Torque Control (CTC)	[67]	Regulates the position and orientation at specific discrete points. Controls position and orientation along a predefined path. Uses computed torques to follow a desired trajectory.	Precise positioning for pick-and-place tasks. Limited to discrete points; not suitable for continuous paths. Smooth, continuous motion for tasks like welding and painting. Requires accurate path planning; computationally intensive. Accurate trajectory tracking; effective for complex dynamics. Requires precise model and tuning.
Force (Torque) Control	[68]	Controls the force or torque applied by the robot's end effector.	Essential for tasks requiring interaction with the environment. Complex to implement; requires precise force measurement.

Control Strategy	Ref.	Technology	Effect and limitations
Model Predictive Control (MPC)		Uses a model to predict future behavior and optimize control actions.	Handles multivariable control problems effectively. Computationally expensive; requires accurate model
Sliding Mode Control		Uses discontinuous control signals to force the system to "slide" along a predefined surface	Robust against system uncertainties and disturbances. Can induce chattering; requires precise modeling.
Nonlinear Control	[69]	Handles systems with nonlinear behaviors and dynamics.	Effective for complex and nonlinear systems. Mathematically complex and computationally intensive.
PD-PI + Fuzzy Logic		Combines PID control with fuzzy logic to handle uncertainties.	Handles nonlinearities and uncertainties effectively. Can be complex to design and implement.
Intelligent Control		Uses AI and machine learning for adaptive control.	Enhances adaptability and learning capabilities in complex tasks. Depends heavily on data quality and quantity.
Cognitive Robotics		Combines AI with cognitive science to enable robots to understand and interact with their environment.	Facilitates higher-level reasoning and interaction capabilities. Highly complex and requires significant computational resources.
Collaborative Robots (Cobots)	[70]	Designed to safely work alongside humans.	Enhances productivity and safety in collaborative environments. Requires advanced safety features and human-robot interaction protocols.
Advanced AI Integration		Integrates advanced AI techniques for higher-level decision making and control.	Provides highly adaptable and intelligent control capabilities. Complexity in integration and requires high computational power.
Autonomous Mobile Robots (AMRs)	[71]	Robots capable of navigating and performing tasks autonomously.	Increases efficiency and flexibility in dynamic environments. Requires advanced navigation and perception systems.
Humanoid Robots		Robots designed to mimic human form and functions.	Capable of performing a wide range of tasks in human environments. Complex design and control requirements; high computational demand.