ISSN: 3005-9852 *Journal of Robotics Spectrum 2(2024)*

Programming Methods for Industrial Robotics and Expanding Applications

[1]Andrea Bersamin and [2]Eugenie Euskirchen
[1,2] Department of Mathematics, Kenai Peninsula College, Soldotna, AK, United States.
[1]andreaminl@hotmail.com, [2]ugfchn@hotmail.com

Correspondence should be addressed to Andrea Bersamin: andreaminl@hotmail.com

Article Info
Journal of Robotics Spectrum (https://anapub.co.ke/journals/jrs/jrs.html)
Doi: https://doi.org/10.53759/9852/JRS202402001
Received 02 October 2023; Revised from 20 November 2023; Accepted 06 January 2024.

Abstract – Industrial robotics industry is presently experiencing significant growth and is generally recognized as a crucial element within the industrial sector. The technology offered by this system is standardized and well-suited for a wide range of automated operations. This research investigates the industrial robotics industry and its use of standardized technologies to automate diverse operational procedures. This article explores the two primary tactics used in the process of robotization, with the diverse levels of cooperation seen between human beings and robots. The present study examines the control and programming approaches used in the field of information retrieval, together with the notable technological advancements that have arisen within this area. Moreover, it incorporates the many challenges and limitations faced during the installation of automated industrial robot systems. This research places particular emphasis on the use of computer vision-based approaches, deep reinforcement learning techniques, simulations, and synthetic data within the domain of industrial robotics. The article ends by providing an analysis of novel control methodologies and the use of external coordinators in the programming of industrial robots.

Keywords – Industrial Robot, Computer Vision-Based Technologies, Industrial Robot Control System, Automation Systems, Innovative Control Approaches, Open System Architecture.

I. INTRODUCTION

According to the ISO 8373:2012 standard, an Industrial Robot (IR) is characterized as a manipulator that is capable of being automatically controlled, reprogrammed, and used for various purposes [1]. It may be programmed to operate in three or more axes and may be either mobile or fixed, designed specifically for uses in industrial automation. Nevertheless, the aforementioned criteria allows for a notable exception in terms of broader application. The categorization of robots into industrial, service, or other sorts is determined based on their intended use. According to the categorization of human-robot collaboration, an examination of current developments in industrial robotics applications reveals the presence of two primary approaches to robotization: classical and contemporary.

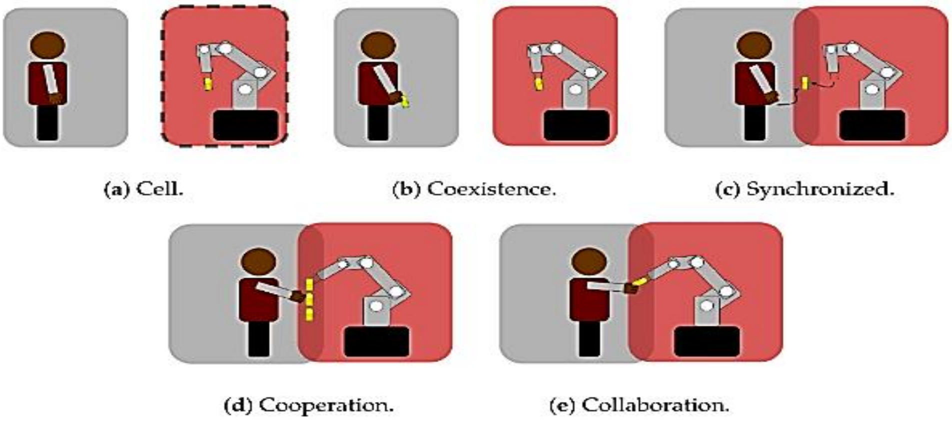

Fig 1. Quantity of Interaction between Humans and Robots.

Fig 2. An Analysis of the Differences between the Working Conditions of (a) Industrial Robots and (b) Collaborative Robots.

In the field of industrial robotics, there are five commonly recognized levels of human-robot cooperation, as seen in **Fig 1**. These levels include: (i) absence of collaboration; (ii) cohabitation; (iii) synchronization; (iv) cooperation; and (v) collaboration. After George Devol was granted the patent for the first industrial robot in 1954, the conventional approach to robotization has advocated for the substitution of human workers with robots in monotonous activities and hazardous work environments [2]. This technique proposes the removal of humans from the robot's workplace (**Fig 2a**). The prohibition of direct collaboration between robots and humans is justified by the possible risks posed to human health and safety.Subsequently, this methodology was further developed to include other dimensions like as precision, dependability, efficiency, and financial considerations.

Industrial robots are often regarded as a fundamental element of competitive production, since they strive to achieve a balance between high productivity, superior quality, and flexibility while minimizing costs. According to data in [3], the number of industrial robot installations exceeded one million in 2007. The automotive sector emerged as the primary user, accounting for almost 60% of these installations. Nevertheless, sophisticated robot technology is expected to play an increasingly crucial role in high-growth sectors such as logistics, health sciences, solar cells, electronics, and food, as well as in developing manufacturing processes like precision assembly, gluing, laser-based processes, and coating. The proportion of robot installations in these sectors has been consistently increasing. The manufacturing of industrial robots and the design, incorporation, and management of robot work cells are distinct engineering activities. To achieve high production volumes, it is essential for a robot design to fulfill the criteria necessary for a diverse range of possible applications. Achieving this objective in practical settings poses challenges, leading to the emergence of many types of robot designs based on factors like as workspace volume, number of robot axes, and payload capacity. These design variations cater to certain application categories including general handling tasks, assembling, machining, palletizing, welding, and painting.

The industrial robotics sector is seeing significant growth, necessitating a comprehensive understanding of control and programming techniques to effectively tackle issues and broaden the scope of industrial robotics applications. The objective of this study is to provide a comprehensive understanding of the present condition of the industry, breakthroughs in technology, and the prospects for future advancements. Through an analysis of control and programming methodologies, this research seeks to ascertain the merits and drawbacks associated with various techniques, and then put forward potential remedies to address any restrictions encountered. The study also emphasizes the significance of human-robot cooperation and the need for standardized programming languages to facilitate the integration of diverse technologies inside industrial facilities. In its whole, this study enhances the existing body of information and comprehension pertaining to industrial robotics, so offering significant insights that are of use to researchers, practitioners, and manufacturers operating within this domain.

The subsequent parts of the paper are arranged in the following manner: Section II presents a review of previous literature works on the concept of robots, and industrial robotics. Section III provides a methodology employed in composing this article. Section IV provides a detailed evaluation of the results, focusing on the concept of industrial robot control systems, and innovative control approaches. Lastly, Section V draws a conclusion on the advancements in control and programming methods for industrial robotics.

II. LITERATURE REVIEW

Since the beginning of the 21st century, as depicted by Kumar, Sudhakar, Samykano, and Jayaseelan [4], there has been a growing focus on the study and development of robots due to advancements in automation. This has led to a surge in their popularity, as well as an increase in the number of applications and improvements being pursued. Significant focus, both domestically and internationally, has been directed into the advancement of robotics technology. Various nations over the globe have presented their individualized strategies for advancement and progress.

According to Wang, Zhang, Liu, Li, and Tang [5], various constraints, such as diverse circumstances on the production floor, a shortage of experienced staff, and the need for functional accuracy and dependability, are impeding the broader use of smart industrial robots. The sustainability of dedicated systems tailored to specific environmental circumstances is no longer viable in the context of smart manufacturing. Various factors, particularly in the context of computer vision-based systems, such as variations in lighting conditions, differences in field of view distances, and the presence of diverse object kinds, may significantly diminish the accuracy of the system. The process of adapting to environmental changes may need adjustments in software or the replacement of physical components. The modification of a specific component should ideally have little or no effect on the functioning of other components in the system. Hence, the quality of modularity, as defined by Gualtieri, Rauch, and Vidoni [6], is considered a fundamental need for the smart industrial robot system.

According to Golnabi and Asadpour [7], the implementation and integration of automated industrial robot systems serve as a substitute for activities characterized by repetition, lack of meaningfulness, high potential for accidents, and low levels of competence that have historically relied on human cognitive abilities and physical agility. The advent of automation has led to the emergence of novel employment opportunities that need a higher degree of creativity. Nevertheless, a notable challenge arises in the form of a scarcity of proficient individuals capable of effectively operating and maintaining sophisticated technology, including intelligent industrial robotic systems. According to Garibaldi [8], the existing skill gap may result in around 2.4 million job vacancies in the United States alone from 2018 to 2028. The operational intricacies of a smart robotic system might prove to be exceedingly intricate, even for proficient operators. The system's maintainability and adaptability to other stated objectives should be achievable without requiring extensive understanding of the underlying target technology.

Poppe [9] argue that despite the potential of computer-vision based solutions in conjunction with industrial robots to effectively address a range of tasks in a dynamic environment, it is necessary to explicitly configure the robot movements. Hence, the method's inability to acquire knowledge via environmental interactions or observational learning constrains its capacity to attain a level of performance comparable to that of humans. Deep Reinforcement Learning (DRL), however, facilitates the ability of robots to acquire knowledge and skills for a given activity without prior knowledge or experience. Currently, it stands as a prominent area of study within the field of robotics. The latest accomplishments have been largely facilitated by the presence of high-performance computing capabilities, extensive datasets, and cutting-edge algorithms. However, in order to acquire proficiency in a task without prior knowledge, a substantial number of attempts are necessary, hence highlighting the issue of sample inefficiency.

The presence of this particular feature, together with the existence of a reality-gap, is impeding the ability of industrial robots to achieve an ideal level of performance when it comes to addressing real-world challenges. Despite the fact that an optimum policy is often pre-existing or may be obtained by an expert in the field of industrial robot control, the bulk of the suggested approaches focus on learning a job from the beginning. The application of expert knowledge in the field of robotics may be achieved via the use of imitation learning techniques. By using various demonstration methods, the reliance on exploratory activities can be minimized. The advantages and disadvantages of each learning method have been extensively discussed by Moreno et al. [10]. The suitability of a particular strategy is mostly contingent upon the complexity and demands of the job at hand. Currently, there is no one approach that can adequately address all grasping circumstances and achieve consistent performance. Nevertheless, by using a combination of learning algorithms, it is possible to optimize the balance between performance and training efficiency.

Simulations and synthetic data may be vital in enhancing the development of intelligent industrial robotic systems across several phases and aspects. The options presented include the ability to generate substantial quantities of data at a reduced cost, expedite the design cycle, and provide a secure and meticulously regulated testing environment. Nevertheless, the aforementioned advantages are accompanied by a number of obstacles that remain unresolved. Within the realm of examined methodologies for controlling smart industrial robots, the obstacles encountered are closely intertwined with the disparities between simulated environments and real-world conditions. Consequently, the efficacy of the system is often compromised when implemented in practical, non-simulated settings, leading to a reduction in accuracy. The challenge of transferring learned models from simulation to reality is influenced by several factors, including physics simulations, virtual representations of objects, reconstruction of sensor data, artificial lighting, and numerous other elements that together constitute the actual world.

In recent years, there has been significant progress in the development of control alternatives and new technologies in both industrial and academic sectors. These advancements aim to address the existing gaps and enhance the range of applications in the area of robotics. These factors enable the attainment of a more adaptable, segmented, non-reliant on certain manufacturers, and accessible framework that can be integrated into various industrial facilities in order to meet a broader spectrum of production requirements. The objective of this study is to provide a comprehensive examination of the control and programming solutions that have undergone extensive research and testing in the field of Industrial Robotics technology.

III. METHODOLOGY

A comprehensive literature review was conducted, including both industrial and scientific domains, in order to achieve the findings presented in this study. The primary objective of the literature study is to discern prevailing concerns and requirements within the domain of IR control and programming. This analysis aims to facilitate conversations about the existing tools and methodologies used in this subject. Hence, in contrast to all-encompassing literature reviews, the current body of knowledge is limited to scientific publications that pertain specifically to the firm field and has the capacity to be implemented in real industrial contexts. In this context, scholarly publications that lack empirical validation or practical application to industrial controllers have been excluded from consideration. The research used the extensively utilized databases Web of science, Google Scholar, and Scopus for data collection and analysis. The initial determination of the keywords to be used in the search process was derived from prior scholarly works that were considered to be relevant to the subject matter. Several examples of topics in the field of robotics are Distributed Control Architecture, “Industrial Robot Control,” “Robot Control,” and “Reconfigurable Control.”

Initially, a total of around 200 articles were chosen and thereafter subjected to processing. The correlation between the number of citations and the significance of an article is not straightforward, particularly for recently published studies, due to the time lag in their dissemination. As a result, the articles underwent scrutiny and evaluation with respect to their pertinence to the topic at hand, rather than being judged just on the number of citations they included. The writers used their discretion to choose the collection of papers, with a particular emphasis on concepts that may be readily applied in industrial assets. Furthermore, the citations used in the publications underwent meticulous verification. In addition to conducting a comprehensive examination of the scientific literature, this study also examined commercially available industrial technology, with a specific emphasis on the leading robot suppliers. The analysis was conducted using internet information such as datasheets, as well as manufacturer-provided documentation such as advanced or particular manuals, which were obtained upon request. The primary focus of the study conducted on each IR controller pertains to the communication protocols, interface choices, and settings, as well as the outwardly available features for the user.

IV. RESULTS AND DISCUSSION

Industrial Robot Control System

When formulating a controller for the purpose of facilitating human-robot cooperation, it is essential to take into account two fundamental elements: customizable autonomy and mixed-initiative. These aspects are vital in the successful integration of people into an autonomous control system. The concept of human initiatives and adjustable autonomy involves the dynamic transfer of task control between an automated system and an operator, in order to effectively adapt to the evolving needs of the robotic system. The present work investigates the practical implementation of a collaborative manipulator robot that actively participates in direct physical collaboration with a human operator inside industrial environments, with a specific focus on its application in collaborative assembly tasks. The extant scholarly literature has examined a range of control structures, spanning from rudimentary to intricate. The collaborative control architecture provides a comprehensive perspective on the interaction between robots and people, including both the lower-level aspects such as actuators and sensors, as well as the higher-level control involving cognition and perception. This is shown in **Fig 3**.

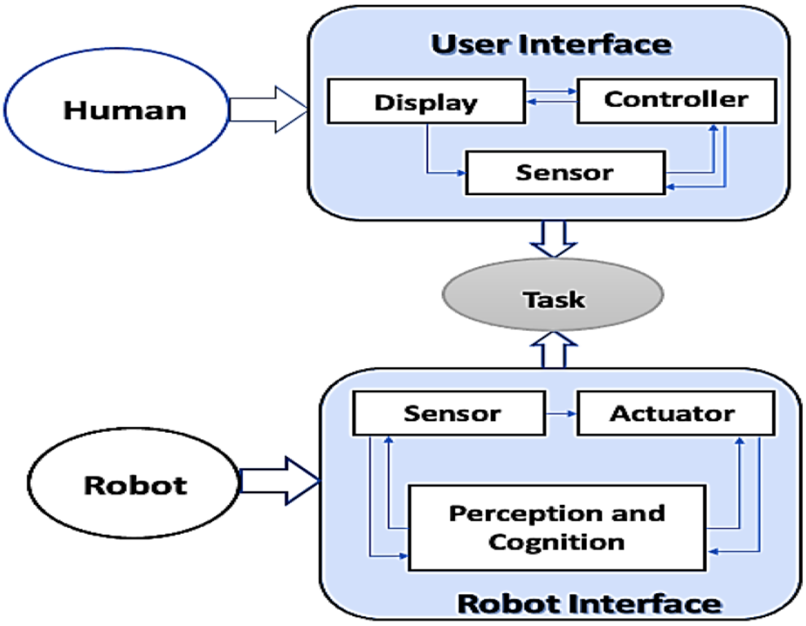

Fig 3. Collaborative Control Architecture Block Diagram.

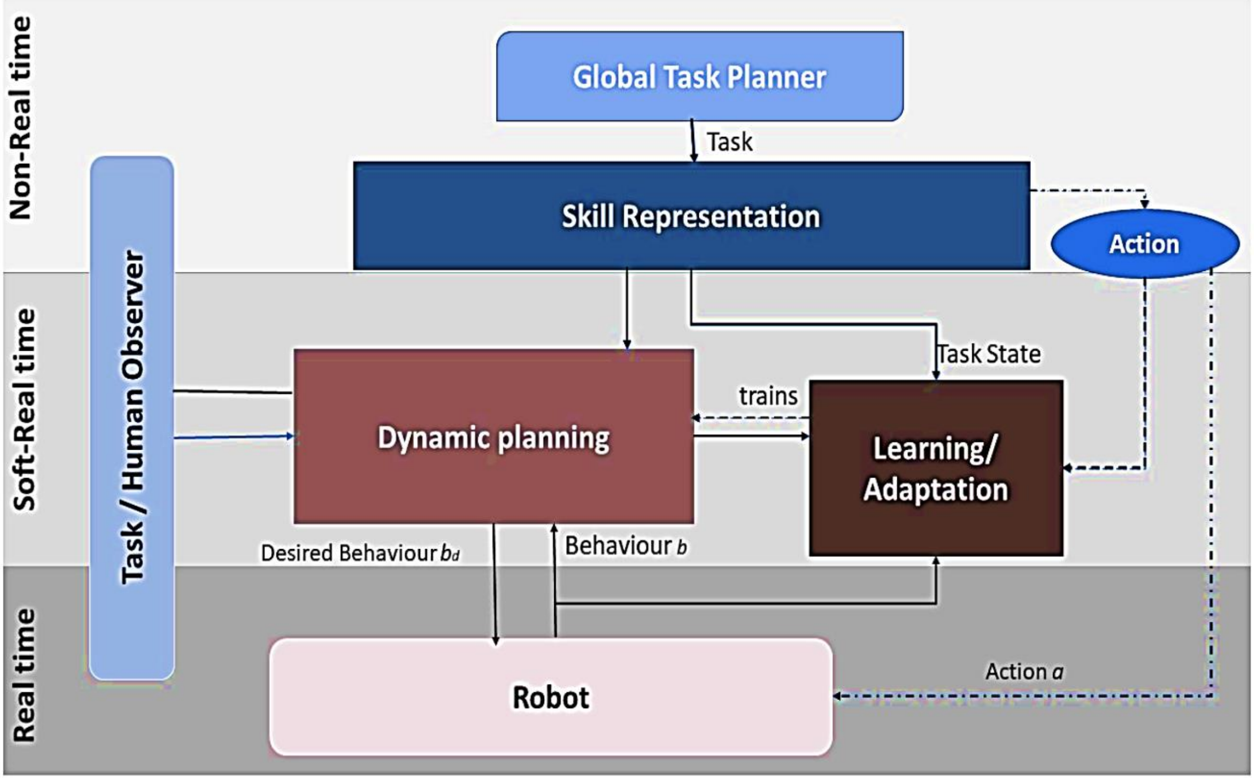

Fig 4. Framework for Cooperative, Interactive Control.

Dimeas and Aspragathos [11] provide an alternative control architecture that offers a more extensive examination of interaction control within the context of human-robot cooperation. The framework in **Fig 4** elucidates the intricate demands and varied approaches pertaining to interaction planning, motion planning, and interaction control. The control architecture under consideration has an atypical nature in contrast to a traditional control architecture, primarily owing to its incorporation of safety-oriented cooperative planning. The control architecture shown in **Fig 4** consists of three distinct abstraction levels, as outlined in **Table 1**.

Table 1. Schematic Representation of a Shared Control System

Non-Real-Time Layer:	The highest degree of abstraction in the architectural framework applies to the topmost layer, which is responsible for formulating a comprehensive plan for the entire assignment of the robot. The present proposal is formulated by capitalizing on the robot's repertoire of abilities in an offline capacity.
Soft Real-Time Layer:	The second level of abstraction is responsible for the dynamic execution and update of global plans. The process of achieving this outcome involves selecting the most appropriate action by considering factors such as the environmental state, present task state, human state, and behavior state.
Real-Time Layer:	The lowest layer of the control hierarchy is known as the low-level control layer. This layer is responsible for receiving the intended action and behavior and immediately transmitting these to the robot for the purpose of task execution. The anticipated response may be modified in the presence of reflexive actions triggered by unintended circumstances or collision occurrences.

The control architecture exhibits human involvement across several levels of abstraction. The collection of human observer statements entails the aggregation of all pertinent information and knowledge pertaining to humans within the secondary layer, which operates in a soft real-time manner. This accumulated data may then be used for the purpose of strategic decision-making within the lower layer. The implemented control architecture effectively guarantees the preservation of human safety during physical interactions in cooperative and interactive activities using a collaborative robot.

The aforementioned procedures are generally executed inside the confines of the IR cabinet, wherein the other electrical together with the servo drives constituents are situated. The technique of programming the robot job and control logic using offline or online methods is a long-standing and widely used approach in the industrial sector. However, the use of inflexible methodologies gives rise to certain practical challenges (as shown in **Table 2**), which significantly affect the performance of intelligent robots. Consequently, these limitations restrict their utilization in contemporary, completely adaptable manufacturing systems.

Table 2. Issues Impacting IR Performance

Time-waste	In order to implement any desired alteration in a robotic job, sequence, or cycle, it is necessary to interrupt the ongoing operation of the plant. This interruption entails manually modifying the program of the robot, followed by downloading the updated program onto the robot controller. Subsequently, prior to restoring the IR to its optimal functionality, it is necessary for the controller to allocate additional time for system reinitialization and execute an initial movement at an extremely reduced speed. Consequently, this process unavoidably results in delays within the production system.
Code redundancy	In contemporary manufacturing processes, a considerable number of robotic systems may be used, whereby each individual robot is equipped with its own distinct code that is run autonomously at a localized level. In the majority of instances, comparable jobs, specifically operations inside the working cycle of an industrial robot (IR), are replicated across several scripts and thereafter installed on each IR controller.
Limitations of programming languages	The absence of a universally accepted programming language that is independent of manufacturers poses challenges when integrating several robot technologies in a single manufacturing facility, unlike machine tool programming which relies on G-code. The linguistic characteristics of robot manufacturers, such as KUKA, ABB, Fanuc, and Stäubli, exhibit variations in terms of complexity, syntax, and semantics, necessitating the involvement of professionals with particular expertise. Furthermore, it is essential for robot programmers to use fundamental instructions and libraries.
Limited motion control	The implementation of complex robot paths, such as those necessary for precision assembly, gluing, or robot machining, is not effectively achievable through conventional programming methods. Traditional programming primarily offers limited motion instructions, typically restricted to step-to-step movements in the circular/liner space motions of the end-factor of the robot. In theory, it is possible to divide the intended non-trivial trajectory into smaller segments and then use conventional motion functions.
No disturbs compensation	The presence of undesirable dynamic effects has a significant impact on the position accuracy of industrial robots (IRs), making them unsuitable for precision manufacturing jobs. These effects cannot be adequately addressed or mitigated using conventional control and programming methods. The key factor contributing to this constraint is the limited accessibility of users to low-level motion control techniques that are implemented within the trajectory generator.
Outdated parts	Industrial Robots (IR) controllers consist of several outdated technical components, including the primary central processing unit (CPU), such as the Pentium 4 processor with a single core, and its related system of operation, such as Windows XP. Furthermore, it is worth noting that the storage capacity on the controller is significantly restricted, sometimes amounting to as little as 25 MB. Consequently, it is essential to consistently verify the size of newly acquired things, such as programs and data files.

In order to address the aforementioned limitations and fulfill the need for collaborative efforts, reconfiguration, and fast adaptation of information retrieval (IR) activities, significant progress has been achieved over the last decade. This progress has been seen in academic research endeavors as well as via the efforts of robot manufacturers, who have introduced more adaptable options for controlling IR processes.

Innovative Control Approaches

Table 3 provides a comprehensive overview of creative control methods aimed at enhancing the capabilities of IR technology. The table includes a detailed description of these solutions, along with relevant examples sourced from existing literature. The aforementioned methodologies have been developed with the intention of facilitating seamless and efficient integration into current production environments. This is achieved by minimizing the need for extensive modifications or replacements of pre-existing hardware and software components.

Table 3. Creative Regulatory Methods for Extending IR Capacities

External coordinator (Fig 5a)	This feature enables the transmission of orders using an external device to the robot, like a personal computers or programmable logic controllers, whereby the robot program is composed in a universally applicable programming language. The translator software on the robot controller operates in a cyclical manner to systematically translate the instructions it receives into a language that is particular to the robot.
External trajectory generator (Fig 5b)	The execution of trajectory planning involves the utilization of an external device, which can be in the form of either a robust personal computer connected to a programmable logic controller or a real-time board. The motion profile generated by the computation process and the feedback from the robot are effectively transferred through I/O robot module, thanks to the implementation of fast-cyclic communication.
Open controller (Fig 5c)	The need to replace the current system with a novel open system may arise as a result of some highlighted concerns, primarily stemming from the restricted accessibility to the lowest levels of control in most controllers of IR. This new system would provide the opportunity for a fully customized control design, enhancing the operational flexibility of the robot and aligning with the requirements of Industry 4.0.

The following Sections provide a comprehensive overview and analysis of the solutions proposed by both industrial firms and academic researchers. These solutions will be thoroughly examined and elucidated.

External Coordinator

The utilization of sequential instruction sets in robot controllers offers a user-friendly programming interface for operators, hence obviating the necessity for a vendor-specific robot programming language. Robotic systems commonly undergo programming within a higher-level environment, such as a Programmable Logic Controller (PLC) editor or the Python programming language, that functions on an external device. The comprehensive guidelines are thereafter converted to the robot controller in a sequential fashion. The robot controller has an interpreter that is specifically designed to execute instructions written in its native language. In order to fulfill this objective, it is possible to use a conventional personal computer (PC) since there is no need for instantaneous contact with the robot in order to convey instructions and access or modify variables, structures, and registers.

The Fanuc PC Developer's Kit (PCDK) is classified under the first category. The utility described facilitates the data exchange and commands between an external PC and Fanuc robot controller, namely the R-J3iB and R-J3 models. The activities performed by the PCDK include the manipulation of variables and numeric registers via reading and writing operations, the execution of tests, the configuration and adjustment of inputs and outputs, the retrieval and modification of locations, as well as the loading and saving of programs. The customized programs are executed on an external personal computer (PC) which establishes communication with the robot controller across a network using designated handshaking protocols. In recent times, there has been an introduction of novel technologies by robot manufacturers, which enable the dynamic streaming of a series of instructions to the robot via an external Programmable Logic Controller (PLC). Examples of such technologies are mxAutomation by motoLogix, KUKA, by uniVAL, YASKAWA, by PLC Motion Interface by Fanuc.

One significant advantage is in the ability to execute all programming operations via an exterior PLC, enabling the centralized management of robots and other devices inside the manufacturing facility. In this scenario, the PLC serves as the exterior coordinator, facilitating streamlined control and coordination. The technologies primarily include of two key components: (i) a server model operational on robotic controller, which remains in a state of readiness to receive orders from a coordinator program executing on the PLC and the external PLC. The use of programmable logic controller (PLC) libraries offered by robot suppliers enables the encapsulation of the initial robot orders and parameters into the appropriate data format. These encapsulated data are then sent to the server by fieldbus, User Datagram Protocol (UDP) or Internet Protocol/Control Protocol (TCP/IP). In this context, the incoming packets are analyzed and then performed, resulting in the transmission of specified messages and parameters back to the Programmable Logic Controller (PLC). The utilization of a singular, centralized, resilient programming habitat enhances the adaptability of manufacturing processes, streamlines the initiation of large-scale facilities, and diminishes the expenses associated with their setup and upkeep by minimizing the need for interventions and specialist staff.

The KUKA company is a leading global manufacturer of industrial robots and automation solutions. The emergence of mxAutomation is discussed by Munz, Braumann, and Brell-Çokcan [12], wherein Andrade and Vinces [13] demonstrate the applciation of visual-programming habitats to facilitate the automated production of KUKA Robot Language (KRL) code, which is later transferred to the robot. The workflow that relies on file-based processes introduces temporal gaps between the establishment of file parameters and their execution, hence emphasizing the need of establishing a direct connection between the robot and the external computer. mxAutomation is regarded as a prospective option due to its capability to facilitate the continuous flow of KUKA instructions and feedback data. In [14], a case study is presented involving the use of two KUKA robots. Specifically, an LBR iiwa collaborative robot and an articulated Agilus robot are employed to perform remote-controlled collaborative tasks aimed at supporting human workers in the task of lifting large objects.

Chang [15] highlight the need of attaining an appropriate size for the buffer used by commands. In accordance with the aforementioned concept, Upchurch, Kuby, Zoldak, and Barranda [16] have put forward comparable solutions centered upon personal computers (PCs), hence exploring the potential for leveraging a wider range of programming languages. In their study, Seth [17] introduced a MATLAB toolbox designed for the purpose of programming Kawasaki and KUKA industrial robots. The transmission of data between the interpreter and the MATLAB program on the robot controller is facilitated by means of data communications, such as serial or TCP/IP connections. The paper discusses a Python-based system that utilizes event-based logic to effectively use Adept robots in dispersed contexts. The communication protocol used is IP/TCP, and instructions are sent as string information. As a component of the open-sourced Robot Operating System (ROS) effort, the ROS-industrial project offers interfaces and libraries that enable the operation of industrial robots (IRs) from various manufacturers via TCP/IP connection from an external personal computer. Nevertheless, it is evident from the official website (https://rosindustrial.org/) that a significant number of these solutions are now in the developmental phase and do not include the whole of safety conditions necessary for effectively managing an industrial robot.

The open-source utility JOpenShowVar, as described by Sanfilippo, Hatledal, Zhang, Fago, and Pettersen [18], offers an alternative methodology that does not rely on instruction streaming. This software utility enables users to conveniently access and alternative variables within a KUKA organizer, thereby obviating the necessity for supplementary proprietary programs. The system's architecture adheres to a server/client model, wherein JOpenShowVar operates as the client on a PC, while KUKAVARPROXY performs as the server on the robot controller. The transmission of data between the server and client takes place using the TCP/IP protocol. As a result, the current technology does not guarantee real-time availability of the robot's data, rendering it unsuitable for streaming point-to-point trajectories. The use of the KUKA Control Toolbox (KCT) as described in [19] enables the execution of the aforementioned tasks, including diagnostics, kinematics computation, visualization, and motion planning for KUKA robots, using a collection of MATLAB functions. In contrast to the preceding coordinators, the KCT maintains its objective of remotely controlling an IR. However, it differs in its approach by using point streaming instead of streaming of instruction, as shown in the generators that are actual-time trajectory discussed in Section IV/3.

External Trajectory Generator

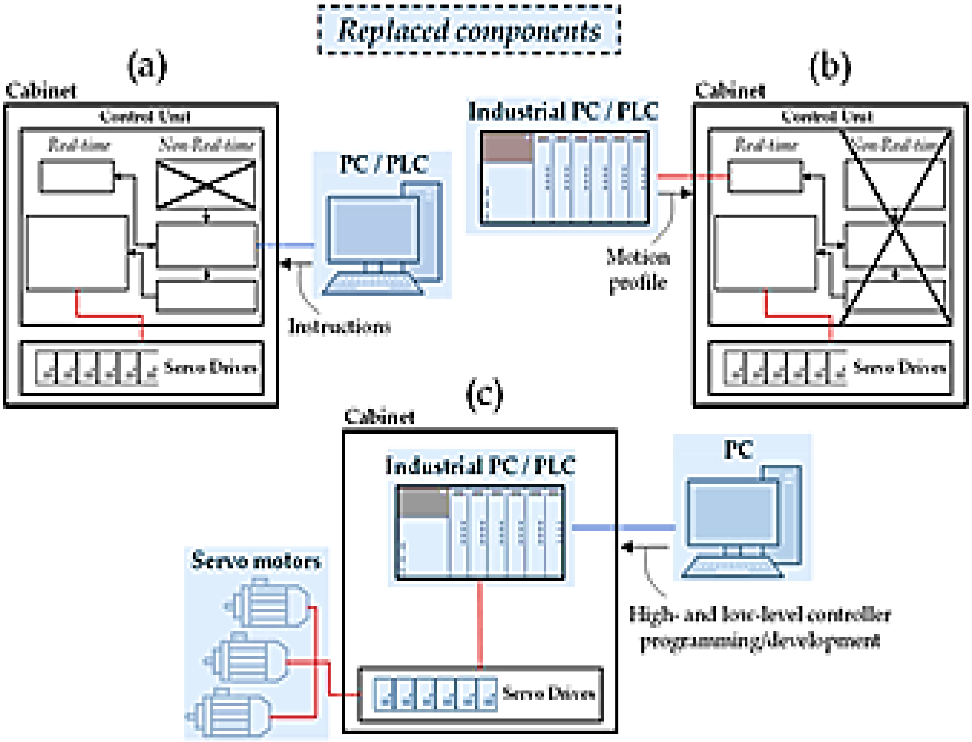

Fig 5. IR Control Schematic Innovative Approaches: External Trajectory Generator.

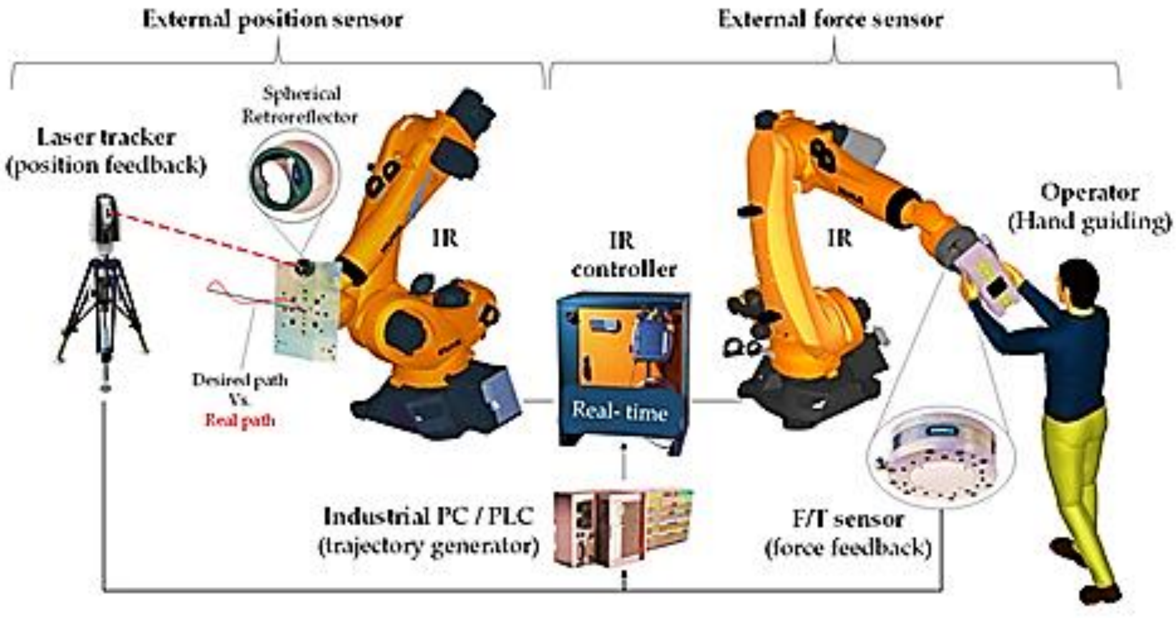

Fig 6. Force-Based Guided and Closed-Loop Real-Time Position Motions.

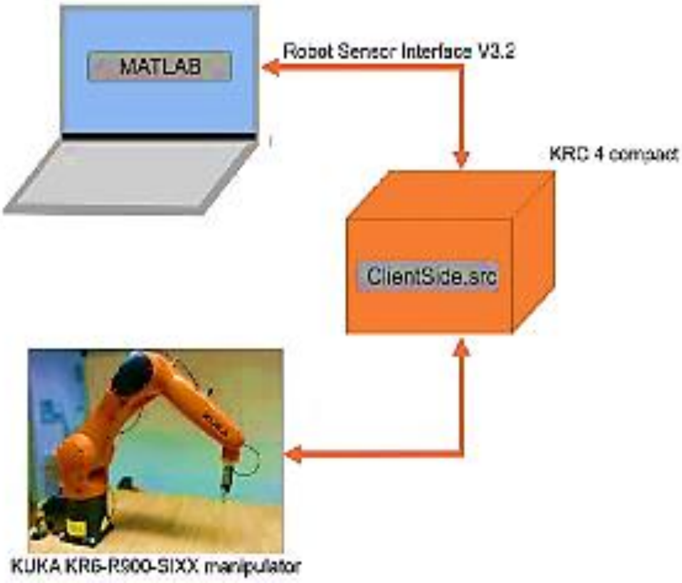

Fig 7. General Functioning RoBO-2L.

There exist scenarios whereby precise manipulation is necessary, particularly in industrial contexts that need the imposition and monitoring of predefined geometric trajectories and/or motion regulations at the terminal actuator. Some illustrative instances are robotic machining, the manipulation of fragile products, accurate assembly, and inspection. In these instances, the utilization of an external coordinator is rendered ineffective as it solely facilitates the execution of the robot's inherent motion directives, such as circular/linear and joint movements. Furthermore, it enhances the efficiency of the internal trajectory planning module by employing hidden proprietary methods to calculate the motion profiles. A frequently proposed approach, which is highly regarded for its minimal impact on the robotic system, entails including an external trajectory generator. To progress within this specific trajectory, it is imperative to provide the IR controller with suitable real-time interfaces that facilitate externally guided movements. Efficient and dependable cyclic communication with external devices, such as an industrial PC or PLC, is crucial in practical applications. This requirement is illustrated in **Fig 5b**, where the external device is responsible for trajectory calculation.

Based on the architectural framework, the external position data obtained via a closed-loop technique that includes additional sensors (such as force or position sensors, as shown in **Fig 6**), requires periodic transmission to the robot controller at a preset frequency. The unpredictability of a robot's mobility in activities aided by sensors might arise from hardware problems or inaccuracies in signal specification, such as sensor malfunction. The absence of coherent algorithms and meticulously calibrated parameters included in the first robot controller leads to the emergence of this unpredictability.

Therefore, the adoption of appropriate safety standards and security checks is crucial in order to effectively mitigate such hazards.

Currently, there exists a variety of commercially available technology packages that provide real-time control capabilities for robots. The technologies discussed above include the Fanuc Dynamic Path Modification (DPM), the ABB EGM (External Guided Motion), and the KUKA. There are two methodologies that may be used for the purpose of facilitating data sharing with the RSI. The first strategy entails the use of Ethernet in conjunction with the IP/UDP protocol, whilst the subsequent choice includes the utilization of a fieldbus together with the I/O robot controller system protocol. The RSI System demonstrates two distinct cycle rates, namely 12 milliseconds and 4 milliseconds, respectively. The transmission of positional data to the RSI module is a fundamental need in each iteration, since the received signals have a direct influence on the robot's motion. In order to mitigate any injury and adhere to the physical constraints of the servo drive, the RSI system also observes and, if necessary, reduces the designated correction value. The ABB EGM system provides the same set of functions, but it is divided into two distinct working modes. The two subjects of inquiry are (i) route correction in electronic gaming machines (EGMs) and (ii) position guiding in EGMs.

The first strategy focuses on optimizing the robot's movement along a predetermined trajectory, which is created externally and sent at a pace of 4 milliseconds. However, it is important to recognize the presence of a possible delay of around 20 milliseconds inside the control process. On the other hand, the latter method allows for immediate adjustments to the trajectory within the sensor dataset. In the provided context, it is vital to implement alterations inside the robot's coordinate system at consistent intervals of 48 milliseconds. In addition, Fanuc has included a dual operating mode that encompasses two unique modalities: the inline DPM and the modal DPM. The first option is suitable for applications that need ongoing adjustments along a predetermined trajectory, including a cycle rate of 8 milliseconds and minimized control delay. Regarding the inline dynamic programming matrix (DPM), it is important to remember that each discrete motion segment experiences a modification at its final destination via the use of an offset value. In direct opposition, the Relative Strength Index (RSI) demonstrates a greater degree of efficacy in comparison to other systems as a result of its ability to implement adaptive adjustments at a frequency of up to 250 Hz. Nevertheless, it is crucial to acknowledge that the closed-loop system's bandwidth exhibits variability at the individual level. Therefore, it is crucial to include considerations of communication delays and control lags in practical evaluations.

In [20], an expanded iteration called RoBO-2L as a solution specifically designed to address the limitations of previous iterations of robot controllers. The remote computer, on which MATLAB is executed, establishes communication with the robot controller via the UDP/IP protocol. In the RoBO-2L programming language, there exists a function named "startConnection(PORT).m" that is used for the establishment of a connection. At the beginning of this project, the startConnection function initiated the execution of an external application to facilitate network connectivity. The planned movement was written by MATLAB into a text file that was then read by an external software. The external application then transferred the contents of the file as a string to the controller. Currently, StartConnection is an external .dll file implemented in C# that handles the network connection in the background, as seen in **Fig 7**. A significant issue arose in the first iterations, when MATLAB attempted to edit the text file concurrently with the network software. As a consequence of these access breaches, instances of time outs and MATLAB crashes were seen. In order for RoBO-2L to function properly, it is necessary for the RSI context to include the Ethernet Object. The interpretation of data by the Object is facilitated with the use of standardized XML Files. Data requests, such as the present condition of the joints, are sent in the form of XML files.

Shu, Gharaaty, Xie, Joubair, and Bonev [21] presented an additional application that utilizes a C-Track 780 from Creaform. They effectively showcased their dynamic posture correction systems, achieving IR pose precision of 0.050 mm. Regarding the compensation of online way during robot traversal, relevant solutions have been implemented on MABI, ABB, and KUKA robots. It is worth noting that these implementations have successfully reduced positioning errors by around 90%. The significance of these findings lies in their potential to expand the use of IR thermography to the realm of large-scale machining processes. In pursuit of this objective, notable outcomes have also been achieved via the use of force-based compensatory measures, as shown in [22]. The use of torque/force sensors that are directly attached to the end-factor of robots has been widely implemented across several domains [25]. These applications include activities such as automated component handling and the facilitation of collaborative operations, such as operator hand-guiding.

These examples demonstrate the benefits of exercising control over the robot in accordance with particular requirements that are depending on the application at hand. The use of sensors with the ability to provide guidance or protection to robots during their motion enables the substitution of conventional machinery often employed in manufacturing assembly lines by industrial robots [26]. Nevertheless, it is important to reiterate that the user has the responsibility of adjusting several parameters, including velocity, acceleration, and position, while also setting the interpolation algorithms. Failure to detect any details may result in not only rendering future IR use impractical but also presenting significant hazards to individuals. Experiments conducted using non-optimized algorithms revealed the occurrence of issues, such as vibrations, during the mobility of robots. Two significant factors to consider are the restricted range of functions and output/input capabilities, as well as the substantial financial commitment required for purchasing packages.

Open Controller

Although the control methods discussed in Section IV/1 and Section IV/2 provide promising results, it is significant to note that access to the lowest layers of control is still limited for the majority of IRs. Hence, several studies have suggested the substitution of the initial apparatus (depicted in **Fig 5c**) and the integration of an accessible, completely customizable control system. Significant advancements in this field were achieved using the Open Modular Architecture Control (OMAC) from USA, Open System Architecture for Control within Automation (OSACA) systems from Europe, and Open System Environment for Controllers (OSEC) from Japan. Most of the advancements use a PC architecture, as seen in [23]. The PC is known for its open nature, versatility, and extensive customization options.

Chung [24] presented a cost-effective system of control that may be used in outdated IR devices. The control architecture included an interface and a conventional PC using a Field-programmable Gate Array (FPGA). The PC assumes the role of executing motion interpolation, performing kinematics calculation, as well as facilitating the transmission of joint position references and reception of current positions. The primary function of the Field-Programmable Gate Array (FPGA) is to interpret and process feedback position data, perform calculations for control output and commutation signals, and then transfer this processed data to the actuators. In pursuit of the same objective, the original controller for the robot is substituted with an embedded industrial CompactRIO controller that is designed using LabVIEW.

V. CONCLUSION

This research has examined control and programming methods to enhance the capabilities and functionality of industrial robots in the context of Industry 4.0 production systems. The primary challenges associated with the first Industrial Robot (IR) are to the complexities involved in reutilizing and reconfiguring codes of program, as well as the too restrictive control schematic. These limitations significantly impede the development of customizable and adaptable processes. The aforementioned concerns have stimulated the curiosity around novel control and programming options that are being offered and debated in the current study. Initially, a comprehensive review of the existing literature was conducted, which was supplemented by the identification of the most up-to-date firm technologies. The chosen literature examined and assessed many novel control options, which were then classified into three primary categories: open controllers, external trajectory generators, and external coordinators.

The works within each category were condensed and organized into comparison tables, which included summaries of their breadth, technical aspects, purpose, strengths, and gaps/limitations. The verification process has shown that the efficacy of each option is contingent upon the specific robot application and the resources of software and hardware that are accessible. Additionally, it has been noted that additional measures are required in order to effectively apply these novel solutions inside the sector. There is a need for industrial technology to make significant advancements in achieving the necessary level of interoperability. In contrast, academic studies need a more thorough exploration of the validation process and the evaluation of the suggested controllers' resilience in real-world situations derived from the industrial context. This finding demonstrates the need of fostering strong teamwork in order to expedite the process of commissioning and implementing novel solutions inside contemporary production systems.

Data Availability

No data was used to support this study.

Conflicts of Interests

The author(s) declare(s) that they have no conflicts of interest.

Funding

No funding agency is associated with this research.

Competing Interests

There are no competing interests.

References

[1]. T. Haidegger et al., “Applied ontologies and standards for service robots,” Robotics and Autonomous Systems, vol. 61, no. 11, pp. 1215–1223, Nov. 2013, doi: 10.1016/j.robot.2013.05.008.

[2]. G. O’Regan, “Unimation,” in Springer eBooks, 2015, pp. 219–223. doi: 10.1007/978-3-319-21464-1_34.

[3]. “Topic: Industrial robots worldwide,” Statista, Jun. 07, 2023. https://www.statista.com/topics/1476/industrial-robots/

[4]. N. M. Kumar, K. Sudhakar, M. Samykano, and V. Jayaseelan, “On the technologies empowering drones for intelligent monitoring of solar photovoltaic power plants,” Procedia Computer Science, vol. 133, pp. 585–593, Jan. 2018, doi: 10.1016/j.procs.2018.07.087.

[5]. S. Wang, C. Zhang, C. Liu, D. Li, and H. Tang, “Cloud-assisted interaction and negotiation of industrial robots for the smart factory,” Computers & Electrical Engineering, vol. 63, pp. 66–78, Oct. 2017, doi: 10.1016/j.compeleceng.2017.05.025.

[6]. L. Gualtieri, E. Rauch, and R. Vidoni, “Emerging research fields in safety and ergonomics in industrial collaborative robotics: A systematic literature review,” Robotics and Computer-Integrated Manufacturing, vol. 67, p. 101998, Feb. 2021, doi: 10.1016/j.rcim.2020.101998.

[7]. H. Golnabi and A. Asadpour, “Design and application of industrial machine vision systems,” Robotics and Computer-Integrated Manufacturing, vol. 23, no. 6, pp. 630–637, Dec. 2007, doi: 10.1016/j.rcim.2007.02.005.

[8]. A. Garibaldi, “5 Numbers That Explain The Talent Gap — And What Leaders Can Do About It,” Forbes, May 10, 2023. [Online]. Available: https://www.forbes.com/sites/guild/2023/05/10/5-numbers-that-explain-the-talent-gap---and-what-leaders-can-do-about-it/

[9]. R. Poppe, “A survey on vision-based human action recognition,” Image and Vision Computing, vol. 28, no. 6, pp. 976–990, Jun. 2010, doi: 10.1016/j.imavis.2009.11.014.

[10]. D. Moreno et al., “Fundamental studies in Design-by-Analogy: A focus on domain-knowledge experts and applications to transactional design problems,” Design Studies, vol. 35, no. 3, pp. 232–272, May 2014, doi: 10.1016/j.destud.2013.11.002.

[11]. F. Dimeas and N. A. Aspragathos, “Online Stability in Human-Robot Cooperation with Admittance Control,” IEEE Transactions on Haptics, vol. 9, no. 2, pp. 267–278, Apr. 2016, doi: 10.1109/toh.2016.2518670.

[12]. H. Munz, J. Braumann, and S. Brell-Çokcan, “Direct Robot Control with mxAutomation: A New Approach to Simple Software Integration of Robots in Production Machinery, Automation Systems, and New Parametric Environments,” in Springer eBooks, 2016, pp. 440–447. doi: 10.1007/978-3-319-26378-6_35.

[13]. R. G. V. Andrade and L. Vinces, “A Comparative Analysis of Kinematics of Industrial Robot KUKA KR 60–3 Using Scientific Computing Languages,” 2020 Asia Conference on Computers and Communications (ACCC), Sep. 2020, doi: 10.1109/accc51160.2020.9347897.

[14]. V. Weistroffer, F. Keith, A. Bisiaux, C. Andriot, and A. Lasnier, “Using Physics-Based digital twins and extended reality for the safety and ergonomics evaluation of cobotic workstations,” Frontiers in Virtual Reality, vol. 3, Feb. 2022, doi: 10.3389/frvir.2022.781830.

[15]. Y.-F. Chang, “Buffered DDA command generation in a CNC,” Control Engineering Practice, vol. 11, no. 7, pp. 797–804, Jul. 2003, doi: 10.1016/s0967-0661(02)00187-9.

[16]. C. Upchurch, M. Kuby, M. A. Zoldak, and A. Barranda, “Using GIS to generate mutually exclusive service areas linking travel on and off a network,” Journal of Transport Geography, vol. 12, no. 1, pp. 23–33, Mar. 2004, doi: 10.1016/j.jtrangeo.2003.10.001.

[17]. A. K. Seth, “A MATLAB toolbox for Granger causal connectivity analysis,” Journal of Neuroscience Methods, vol. 186, no. 2, pp. 262–273, Feb. 2010, doi: 10.1016/j.jneumeth.2009.11.020.

[18]. F. Sanfilippo, L. I. Hatledal, H. Zhang, M. Fago, and K. Y. Pettersen, “Controlling KUKA Industrial Robots: Flexible Communication Interface JOPENShowVAR,” IEEE Robotics & Automation Magazine, vol. 22, no. 4, pp. 96–109, Dec. 2015, doi: 10.1109/mra.2015.2482839.

[19]. F. Chinello, S. Scheggi, F. Morbidi, and D. Prattichizzo, “KUKA Control Toolbox,” IEEE Robotics & Automation Magazine, vol. 18, no. 4, pp. 69–79, Dec. 2011, doi: 10.1109/mra.2011.942120.

[20]. J. Golz, T. Wruetz, D. Eickmann, and R. Biesenbach, “RoBO-2L, a Matlab interface for extended offline programming of KUKA industrial robots,” 2016 11th France-Japan & 9th Europe-Asia Congress on Mechatronics (MECATRONICS) /17th International Conference on Research and Education in Mechatronics (REM), Jun. 2016, doi: 10.1109/mecatronics.2016.7547117.

[21]. T. Shu, S. Gharaaty, W. Xie, A. Joubair, and I. A. Bonev, “Dynamic path tracking of industrial robots with high accuracy using photogrammetry sensor,” IEEE-ASME Transactions on Mechatronics, vol. 23, no. 3, pp. 1159–1170, Jun. 2018, doi: 10.1109/tmech.2018.2821600.

[22]. S. C. Jagers, J. Martinsson, and S. Matti, “The impact of compensatory measures on public support for carbon taxation: an experimental study in Sweden,” Climate Policy, vol. 19, no. 2, pp. 147–160, May 2018, doi: 10.1080/14693062.2018.1470963.

[23]. C. Brecher, A. Verl, A. Lechler, and M. Servos, “Open control systems: state of the art,” Production Engineering, vol. 4, no. 2–3, pp. 247–254, Apr. 2010, doi: 10.1007/s11740-010-0218-5.

[24]. C. A. Chung, “A cost-effective approach for the development of an integrated PC-PLC-robot system for industrial engineering education,” IEEE Transactions on Education, Jan. 1998, doi: 10.1109/13.728266.

[25]. S. P. Jadhav, A. Srinivas, P. Dipak Raghunath, M. Ramkumar Prabhu, J. Suryawanshi, and A. Haldorai, “Deep learning approaches for multi-modal sensor data analysis and abnormality detection,” Measurement: Sensors, vol. 33, p. 101157, Jun. 2024, doi: 10.1016/j.measen.2024.101157.

[26]. A.Haldorai, B. Lincy R, S. M, and M. Balakrishnan, “An improved single short detection method for smart vision-based water garbage cleaning robot,” Cognitive Robotics, vol. 4, pp. 19–29, 2024, doi: 10.1016/j.cogr.2023.11.002.